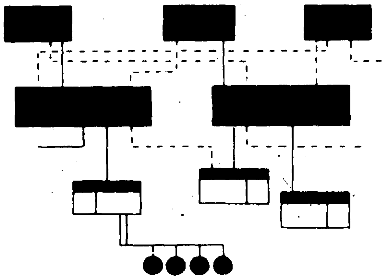
GE-625/635 Comprehensive Operating Supervisor (GECOS-III)

Information Systems

Information Systems Equipment



GENERAL ELECTRIC

CPB-1518

CPB-1518

GE-625/635 Comprehensive Operating Supervisor (GECOS-III)

REFERENCE MANUAL

May 1968

INFORMATION SYSTEMS

GENERAL ELECTRIC

PREFACE

This reference manual presents a description of the Comprehensive Operating Supervisor for the GE-625/635 computer system. It is assumed that the basic philosophy of GECOS will be used by all installations in the execution of their job programs.

Chapter 1 provides a general description of GECOS and many of its features and capabilities. Chapter 2 describes the phases through which a user's activity/job passes. Chapter 3 gives a brief summary of the system components. Chapters 4 through 11 describe the areas with which the user must be concerned when he submits his job for processing under GECOS. Chapter 12 and 13 are included for the user who requires more detail concerning the way in which GECOS allocates peripherals and the use of the permanent file catalog. Typical deck setups, error messages, and status codes are given in the Appendixes. For ease of reference, an index is included.

Changes in technical content from the description of GECOS-II (CPB-1195) are indicated by change bars.

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, Processor Equipment Department, General Electric Company, 13430 North Black Canyon Highway, Phoenix, Arizona 85029.

(5M 5-68)

CONTENTS

APPENDIXES

ILLUSTRATIONS

1. INTRODUCTION

The primary objectives of the GE-625/635 software system are:

- To integrate batch processing, remote batch, and time-sharing requirements for maximum utilization
- To provide a common file system with file protection and access control, which is accessible from all modes of processing
- To increase the total "throughput" of the computer (the amount of work that may be performed in any given time)
- To reduce user program "turn around" time in large-scale installations (elapsed time from program submission to the machine room up to return of program solutions)
- To provide logical easy-to-use programmer and operator interfaces with system software
- To maximize peripheral availability through concurrent test-and-diagnostic program execution

The attainment of these objectives is achieved by the GE-625/635 Comprehensive Operating Supervisor (GECOS), the overall manager of the hardware/software system, through efficient use of the hardware features and the supervision of a multiprogramming/multiprocessing environment (which is the normal operating mode of the GE-625/635). The significant features provided by GECOS as related to the several primary objectives above are summarized in the list following. These features are implemented by the modules, programs, and subroutines within GECOS:

Maximum system throughput via multiprogramming
Multiprocessor control
On-line permanent file system
Concurrent time-sharing
On-line peripheral test system
Multiple local and/or remote job input streams
Scheduling and coordination of jobs
Input/Output supervision on an interrupt-oriented basis
File-oriented programming (instead of device-oriented)
Fault detection with standard GECOS or optional programmer-supplied corrective actions
Maximum efficiency of core memory by dynamic program relocation and by system-controlled subprogram overlays
Modular construction to simplify maintenance

MULTIPROGRAMMING

Although each user programmer writes his job program as though he had exclusive use of the computer, he is in fact generating a program that will reside concurrently in memory with other user programs and will be executed in a time-shared manner. That is, any given program is processed until it is held up (usually because of the need for some input/output to be completed) at which time the next most urgent program is processed. Transfer between programs under multiprogram execution is performed by means of the hardware interrupt facility (in the System Controller) working with the Dispatcher routines in the Input/Output Supervisor. The ways by which a user program can be temporarily delayed in execution are:

DELAY TYPE	REASON
Roadblock	Program cannot progress until all input/output requests have terminated
Relinquish	Program relinquishes control until any one of its input/output requests has terminated
Forced Relinquish	Program was interrupted because a timer runout occurred.

Each time a program yields control to the Operating Supervisor by means of Roadblock, Relinquish, or Forced Relinquish,the Supervisor has the opportunity to give control to another program in core which can make effective use of the processor.

In giving such control, the Supervisor examines the following conditions:

- Program urgency (priority) compared to other programs that reside in memory
- Roadblock status involving completion of all input/output
- Completion of input/output that was pending when the last Relinquish was given
- Request present for use of the processor

ON-LINE MEDIA CONVERSION

Media conversions are of two basic types (1) bulk media conversion, whereby large volumes of data in a single format and for a single purpose are processed and (2) system media conversion where low-volume sets of data each with its own format and purpose are processed.

Bulk media conversion is performed by a system program, the GE-625/635 Bulk Media Conversion (BMC), which may be called into execution by use of a control card. Other control cards will direct the routine as to where to find the input and where to place the output. Use of BMC is described in GE-625/635 Bulk Media Conversion, CPB-1096.

On-line media conversions for both input and output are performed as a normal part of the multiprogramming environment of the GE-625/635. Normal job input is carried out by input media conversion, which reads job input from card readers, magnetic tape, file system, and from remote devices concurrently; scans the control cards for execution information; and records the jobs on the input queue located on system storage.

System media conversions of program output data are automatically performed by the Output Media Conversion routines executed in protected memory. The programmer specifies that a particular output file be written on the permanently assigned system output (SYSOUT) file by use of the PRINT, PUNCH,or WTREC calling sequences described in the GE-625/635 File and Record Control reference manual, CPB-1003. Once on the SYSOUT file, the output is converted to hard copy or punched cards by Output Media Conversion, concurrently with other user programs under execution in the multiprogramming environment.

CENTRALIZED INPUT/OUTPUT

In the multiprogramming environment where several programs may concurrently request input/output, a facility must be provided (1) for processing such multiple requests in terms of the efficient use of the entire peripheral complement and (2) for maintaining continuous processing of the multiple programs in core storage. The GECOS program that performs these general functions is the Input/Output Supervisor (IOS).

The main functions of IOS are to initiate an input/output activity and to respond to the termination of an input/output activity. In addition, IOS provides the following functions:

- File code to physical unit translation
- File protection of user files
- Pseudo tape processing on disc/drum
- Supervision of all input/output interrupts
- Queueing of input/output requests
- Utilization of crossbarred peripheral channels
- Maintenance of an awareness of the status of each peripheral
- Accounting of time spent by the processor and all peripherals for each program executed

When IOS receives a request to perform an input/output function, it looks at the communication cells and issues a Connect instruction. If the particular channel is busy, the request is placed in a waiting queue. If the request queue for that program is full or if the program indicated should be roadblocked until all input/output is complete, control is given to another program residing in memory.

When the input/output operation terminates, control is given to IOS to perform all necessary termination functions. At this point, the request queue is examined and, if any requests for the channel are in queue, they are then performed. If applicable, latency reduction techniques are applied to maximize mass storage effectiveness.

MASTER/SLAVE RELATIONSHIP

The processor operates in either master or slave mode. GECOS executes in both modes, as required. When executing a user program, a processor is in slave mode. The prime reason for the master mode of operation is to protect GECOS and user programs, as well, from modification by other user programs. This feature is vital in the multiprogramming environment and is closely tied in with memory protection, accounting determinations, multiprogram interrupt management, intermodule communications control, and input/output operations. Each of these functions is implemented by a processor instruction that requires the master mode. These are listed below.

All instructions available to the processor in slave mode are available in master mode. The following instructions can be executed only when the processor is in master mode.

- Load Base Address Register (LBAR)
- Load Timer Register (LDT)
- Set Memory Controller Interrupt Calls (SMIC)
- Read Memory Controller Mask Registers (RMCM)
- Set Memory Controller Mask Registers (SMCM)
- Delay Until Interrupt Signal (DIS)
- Connect Input/Output Channel (CIOC)

The last of these instructions, Connect Input/Output Channel, is the beginning of every peripheral operation. Thus, all peripheral operations are reserved for execution in master mode, and in particular by the Input/Output Supervisor within GECOS.

MASTER MODE ENTRY

Although master mode operation by the processor is a primary safeguard for executive routines and user programs in memory, the applications programmer can force the processor into this mode, but only for accessing routines that are part of GECOS. This is done by use of the Master Mode Entry (MME) instruction and one of the system-symbol operands described in Chapter 7. Any other use of MME causes an abort of the user program. Thus, through the MME instruction, the programmer can communicate with modules of GECOS to exchange any necessary information for the execution of his program.

MASS STORAGE ORIENTATION

"Compute overhead" time is reduced and multiprogramming is enhanced through the use of an external system (mass) storage unit. This system storage can be either drum or disc and enables optimized accessing of system routines.

The drum and/or disc is used primarily for the following purposes:

1. System storage area--Least used modules of GECOS and all system programs are stored on the drum. Included in this storage area are the Assembler, compilers (FORTRAN, COBOL, etc.), portions of GECOS, sort/merge, utility routines used by system routines, tables associated with storage allocation, operational statistics, hardware diagnostics, and the General Loader (GELOAD) with its debugging routines.
2. Temporary data storage--Temporary data files used during job execution can be stored on the drum or disc for fast access.
3. Permanent user files--Permanent data files can be stored on the drum or disc and accessed through the software system.

PROGRAM FILE ORIENTATION

The software system is further described as file oriented because (1) GECOS assigns peripheral devices to an activity (by file code designators) and (2) it manages all assigned peripherals during input or output operations so that the programmer never deals directly with input/output subsystems or devices. The programmer references all peripherals by use of file code designators (two alphanumeric characters), that are referenced in two ways: (1) on file control cards used by the Allocator in GECOS to specify those files needed to execute the activity and (2) in communicating to the File and Record Control (GEFRC) program or to the Input/Output Supervisor in GECOS. The file code designators and their assigned Peripheral Assignment Table (PAT) are used by the Input/Output Supervisor for peripheral identification.

RELATED GE-625/635 DOCUMENTATION

In addition to this reference manual describing the Comprehensive Operating Supervisor (GECOS), the following documentation may also be of interest:

- GE-625/635 System Manual, CPB-371
- GE-625/635 File and Record Control, CPB-1003
- GE-625/635 Programming Reference Manual, CPB-1004
- GE-625/635 FORTRAN IV Reference Manual, CPB-1006
- GE-625/635 COBOL Reference Manual, CPB-1007
- GE-625/635 General Loader, CPB-1008

- GE-625/635 Bulk Media Conversion, CPB-1096
- GE-625/635 UTILITY, CPB-1422
- GE-625/635 GERTS Programmer's Reference Manual, CPB-1417
- GE-625/635 GERTS Operator's Reference Manual, CPB-1418
- GE-625/635 GERTS Reference Manual, CPB-1416
- GE-625/635 GECOS-III Software Maintenance Documents (Implementation):

    GE-625/635 GECOS-III Introduction and System Tables, CPB-1488

    GE-625/635 GECOS-III Startup, CPB-1489

    GE-625/635 GECOS-III System Input, CPB-1490

    GE-625/635 GECOS-III Dispatcher and Peripheral Allocation, CPB-1491

    GE-625/635 GECOS-III Rollcall, Core Allocation, Operator Interface, CPB-1492

    GE-625/635 GECOS-III Fault Processing and Service MMEs, CPB-1493

    GE-625/635 GECOS-III I/O Supervision, CPB-1494

    GE-625/635 GECOS-III Exception Processing, CPB-1495

    GE-625/635 GECOS-III Termination and System Output, CPB-1496

    GE-625/635 GECOS-III File System Maintenance, CPB-1497

    GE-625/635 GECOS-III Utility Routines, CPB-1498

    GE-625/635 GECOS-III Comprehensive Index and Glossary, CPB-1499

    GE-625/635 GECOS-III Flowcharts, CPB-1500

    GE-625/635 GECOS-III Time-sharing System, CPB-1501

- GE-625/635 File System (FILSYS), CPB-1513

2. JOB FLOW

Jobs flowing through GECOS pass through five phases.

- Input Media Conversion
- Allocation
- Execution
- Termination
- Output Media Conversion

This chapter explains these phases and then traces a hypothetical job as it flows through GECOS.

Throughout this manual the terms "job" and "activity" have specialized meanings. An "activity" consists of a single program (such as FORTRAN compilation or an object program load and execution). A "job" is a set of activities which together constitute a problem or a production application. The activities constituting a job must be processed in sequence.

INPUT MEDIA CONVERSION (IMC)

When a job is presented to the system, either from an on-line card reader or any other input device (e.g., DATANET-30* communications processor, IMCV tape), it is processed by IMC. IMC performs two functions in processing a job. It separates the input into data used to control or describe the job and into data used by the job. In separating the input, IMC builds two files for each job. One file (J*) contains the control cards describing the job and 1-word pointer records pointing to various parts of the second (data) file. (This file eventually becomes the first part of the execution report.) The data file (*J) comprises various subfiles consisting of the data to be used by the job.

If remote batch input is presented to the system, Remote IMC processes input from the DATANET-30 communications processor. Data blocks from the DATANET-30 are buffered and stored in random access memory. Remote IMC also multiplexes job source input from remote terminals connected to the DATANET-30 processors. Then, input processing continues with the segregation of the job into the two files, as with local job input.

*DATANET, Reg. Trademark of the General Electric Company

ALLOCATION

During allocation, peripherals and memory are assigned to a specific activity after the control tables are examined to determine gross peripheral requirements. Through this technique of allocating peripherals in advance of execution, the operator can perform functions (such as mounting tapes) while other activities are in execution. It is during the Allocation phase that these instructions are issued to the operator. When the assigned peripherals are ready and sufficient memory is assigned to the activity, the Allocation phase is completed, and the activity passes into the Execution phase.

If the peripheral requirements of an activity can be met only partially, the job is bypassed until all of its peripheral needs can be satisfied. If allocation is still not possible, after x tries or y minutes (assigned by the installation), the threshold value of 40 may be assigned by the operator. This blocks peripheral and memory allocation for other activities. Other jobs are not considered for allocation until the requirements of this job can be met. No activity of a job is initiated until all prior activities of that job have been completed.

EXECUTION

During execution, each activity is executed under the supervision of the Dispatcher. All user communication with GECOS is provided at execution time by the use of Master Mode (MME) routines, as described in detail in Chapter 7. These routines provide requests for service, exchange of information, and control functions. Many of the MME routines are called upon automatically by the compilers and by GEFRC (see GE-625/635 File and Record Control, CPB-1003); the user need not be concerned with a majority of the MME routines if he is using any of these.

GECOS features a dynamic dispatch queue. Program entries are in the queue only if they are ready to be executed. The queue is ordered according to priority of the programs to be executed. The top priority program is at the head of the queue. Dispatching is therefore a very simple and straightforward mechanism of picking the top entry from the queue and dispatching to it. The dispatch queue is interrupt-oriented. Control is returned to the Dispatcher at the completion of interrupt processing. Since the queue may be modified during interrupt processing, the priority of program execution is truly interrupt-oriented.

Responses to any I/O exception conditions or processor faults are handled during this phase. When encountered, standard actions are performed or the programmer (or the operator) is given the option of handling the situation in his own way. This is done through Exception Processing Routines (GEPR), which are described in Chapter 11 and are listed in Appendix C.

TERMINATION

A job or activity may terminate either normally or abnormally. The specific termination procedure depends upon the type of termination. Both procedures are discussed in Chapter 9. Upon program termination, an error and accounting record (see Chapter 10) is produced on the System Output File (for eventual printing on the execution report), the System Output File is closed, the operator is told if files require dismounting, tables used during execution are deleted, and memory and peripherals are deallocated.

When successive GMAP or FORTRAN activities are encountered without intervening $ IDENT or File control cards, they are run as a single GMAP or FORTRAN activity.

Thus a job containing ten FORTRAN compilations followed by a $ EXECUTE is run as two activities, a FORTRAN compilation of ten routines and a GELOAD execution. Activity termination takes place only after the last FORTRAN compilation and after the GELOAD execution.

OUTPUT MEDIA CONVERSION

All output written on the System Output File (P*) is transcribed into printer or punch card output.

EXAMPLE OF JOB FLOW

The following hypothetical case is traced as it flows through GECOS to illustrate the specific actions taken after a job is submitted in the form of the following card deck:

```
                              Column
             1               8                      16
             $               SNUMB                  TST12,3
             $               IDENT                  JOB12,JOE SMITH
             $               OPTION                 FORTRAN
             $               FORTRAN                LSTOU,DECK,DUMP
             $               INCODE                 IBMF
Activity           •         •              •       Source Input
   1               •         •              •
             $               FORTRAN                DUMP
                   •         •   5*         •       Source Input
                   •         •              •
             $               FORTRAN                DUMP
                   •         •              •       Source Input
                   •         •              •
Activity     $               GMAP                   LSTOU,DECK,DUMP
   2         $               TAPE                   G*,X1D,123,,ASSEMBLER
                                                           5   3
             $               EXECUTE                DUMP
             $               LIMITS                 50,20000,4000,9000
             $               TAPE                   02,X1R
             $               TAPE                   03,X2R
Activity     $               TAPE                   04,X3R
   3         $               INCODE                 IBMF
                   •         •              •       Data
                   •         •              •
             $               ENDJOB
```

The deck TST12 is submitted to computer operations for processing. Operations punches and inserts a $ SNUMB (job sequence number) control card in front of the deck and places it into the card reader

Input Media Conversion

Input Media Conversion (IMC) scans the cards for a $ SNUMB control card, which will tell IMC that a new job is beginning.

ACTIVITY 1

Upon detecting a $ SNUMB, IMC initializes the control stack for the first activity by placing the Job Identifier and Urgency in their relative positions. The next card ($ IDENT, JOB12, JOE SMITH) is read into the control stack buffer in its entirety. This information will be used to produce the error and accounting report for this activity. The next control card found is the $ OPTION card. IMC opens a Loader input (R*) file and puts the $ OPTION card onto R*. The $ OPTION... FORTRAN card will be used later during loading to set the standard load options for FORTRAN. The $ FORTRAN card indicates that the first activity has been defined. A $ SOURCE card is created and placed on the R* file; detail entries are made in the control stack for allocation of the B*, C*, K*, *1, G*, P*; and S* files (see Chapter 6) and the standard limits are set for the FORTRAN Compiler. The LSTOU and DECK options listed on the $ FORTRAN card are encoded into the switch word contained in the control stack buffer. These options will override the standard options associated with the FORTRAN Compiler. Source input following the $ FORTRAN is written onto S* in the GE-635 character set (encoded by the $ INCODE card).

ACTIVITY 2

When the $ GMAP control card is encountered, the standard GMAP Assembler limits are set up. The $ TAPE card creates a file detail entry for G*. (The $ TAPE card overrides the normal file which would be opened upon encountering just the $ GMAP card without a following $ TAPE card.) The presence of the $ TAPE card signals the existence of source input on a tape named G* having a Reel #123, which the operator must mount.

ACTIVITY 3

The detection of the $ EXECUTE card signals that Activity 2 has been defined and input for execution activity (Activity 3) is beginning. The limits specified on the $ LIMITS card override the standard limits which would normally be used in an execute activity. In addition, when the $ EXECUTE card is detected, detail entries for the normal execution files I*, B*, and P* are made in the control stack, and Bit 5 of the Program Switch Word is turned on.

The next three $ TAPE cards make file detail entries for three tape files designated 02, 03, and 04 which will be assigned cyclically to a physical channel (X1, X2, and X3). Also a part of Activity 3, the $ INCODE card signals IMC to transliterate data from the IBM character set to the GE character set. The data cards following the $ INCODE card will then be placed on the I* file. The $ ENDJOB signals that this job has completed Input Media Conversion and is now ready for Allocation.

Allocation, Execution, and Termination

IMC assigns a program number (1-63) and makes an entry into the Allocation Urgency List for the TST12 job and control is returned to the Dispatcher. The Dispatcher enables the Allocator which checks the Allocation Urgency List. When TST12 is the most urgent job requiring allocation, the Allocator reads the control stack for the first activity to determine the peripheral requirements. The peripherals and scratch files are allocated, and memory is allocated for the first activity. A MME GECALL is placed in the slave program prefix (32_8) with a location specified to load FORTRAN from the Software Library, and compilation begins. Source input is read from the S* file, and the FORTRAN compilation is completed. The output generated is put onto the output file P*. Since the execution bit (Program Switch Word) was turned on, the output is also put onto B* (for a subsequent Load and Go). The B* file will be used by subsequent activities of this job, and therefore is saved and put into an Abeyance Table so that any future requests for B* will be read from this file.

After termination of the first activity is completed, control is returned to the Allocator, and allocation of the second activity begins. A MME GECALL... GMAP is placed in the slave program prefix (32_8) with a location specified to load GMAP from the Software Library, and the assembly begins. The GMAP Assembler reads source input from the G* file and puts its output onto the B* file.

Allocation for the execution activity is enabled, and allocation of peripherals begins. Memory requested in the $ LIMITS card (20,000) is added to the memory requirements for GELOAD (9000) minus any memory which can be shared with GELOAD during loading (4000). If sufficient memory is available, it is allocated and a program number is assigned to the execution activity. A MME GECALL... GELOAD is placed in the slave program prefix (32_8) and execution begins. The object decks are read and loaded and then the program is entered at its primary entry point. When execution (Activity 3) is completed, all peripherals and memory are released, and the entry made in the Allocation urgency table is deleted.

Output Media Conversion

Output written on P* is printed or punched and the accounting record is produced.

3. SYSTEM SUMMARY

GECOS comprises 13 major functional components. These components are listed below and discussed briefly in this chapter.

- Startup
- System Input
- Dispatcher
- Peripheral Allocation
- Rollcall, Core Allocation, and Operator Interface
- Fault Processing and Service MME's
- I/O Supervision
- Exception Processing
- Termination
- System Output
- File System Maintenance
- Utility Routines
- Time-Sharing

Each of these functions is accomplished by one or more modules within GECOS. These are discussed in detail in the GECOS-III Software Maintenance Documents referenced in the summaries below.

STARTUP

Startup is performed by a Startup package which is, in itself, a miniature operating system. It provides a variety of operations including loading, initialization, and utility functions required for system startup.

Implementation of the sections comprising Startup and the use of the Startup Deck are described in GE-625/635 GECOS-III Startup, CPB-1489.

SYSTEM INPUT

The system input modules accept system input, whether from an on-line card reader, remote terminal, or other device. This input is then segregated into control and data files by these modules, which are called into core only when needed. Multi-input streams may be used since copies of the input modules are read in when required. Implementation of system input including remote input is described in GE-625/635 GECOS-III System Input, CPB-1490.

DISPATCHER

The Dispatcher element functions to keep as many system components as possible in simultaneous use. The Dispatcher accomplishes this by selecting the highest urgency program (either master or slave mode) which can make effective and immediate use of a processor and/or peripheral subsystems. By doing this, the Dispatcher allows overlapping of I/O operations of the various programs. A program, whether it is a GECOS module or a slave activity, can be entered only from the Dispatcher. The Dispatcher permanently resides in core.

Implementation of the Dispatcher is described in GE-625/635 GECOS-III Dispatcher and Peripheral Allocation, CPB-1491.

PERIPHERAL ALLOCATION

Peripheral allocation in GECOS-III is done by a priviledged slave program comprising a group of modules which are called into core as required.

Peripheral allocation implementation is described in GE-625/635 GECOS-III Dispatcher and Peripheral Allocation, CPB-1491.

ROLLCALL, CORE ALLOCATION, AND OPERATOR INTERFACE

Rollcall, core allocation, and processing operator messages are done by a GECOS-III program POPM, which always resides in core.

The GE-625/635 GECOS-III Rollcall, Core Allocation, and Operator Interface, CPB-1492, describes the implementation of these functions.

FAULT PROCESSING

The method by which faults are processed depends upon whether the fault occurred in the master or slave mode.

If in the master mode, an error message is typed on the console typewriter (see Appendix B for error messages) and the registers, IC, etc., at the time of fault are dumped. If a noncontrol processor found the fault, the noncontrol processor will make a request to the control processor asking for an interrupt. The control processor then interrupts all noncontrol processors by a DIS (Delay until Interrupt Signal), and the control processor takes the dump even though it did not find the fault. If a control processor initially finds a fault, it issues a DIS to all noncontrol processors and, in this case also, takes the dump.

If in slave mode when a fault is encountered, the fault will cause interrogation of the user fault vectors, if the fault is a program fault. If the user specifies a fault processing subroutine, control will be transferred via this vector back to the slave activity. If no fault processing subroutine is specified in the user fault vector or if the fault is one in which no option exists (hardware fault) for letting the user specify a fault processing subroutine, a MME GEBORT is set in the user program and control is transferred to MME GEBORT (see Chapter 7). GECOS allows the following faults to be processed optionally by a user error subroutine.

Program Faults

1. Memory
2. Divide Check
3. Overflow
4. Command
5. Illegal Op Code
6. Fault Tag
7. Derail

The user is allowed an unlimited number of Divide Check, Overflow, and Derail faults. However, he can have only one Memory, Command, Illegal Op Code, or Fault Tag fault at which time he will be aborted.

The following faults do not have an option allowing the user to go to a fault processing routine.

Hardware Faults

1. Lockup
2. Parity
3. Op not Complete
4. Startup
5. Timer Runout
6. Shutdown
7. Execute
8. Connect

In addition to those faults listed above, a Master Mode Entry (MME) will cause a fault which is then transferred to a fault routine. Standard actions for processing faults are given in Appendix C. The GE-625/635 Programming Reference Manual, CPB-1004, describes the fault trapping procedure.

I/O SUPERVISION

I/O Supervision involves the acceptance, initiation, and termination of all I/O requests. All I/O operations on the GE-625/635 are performed by executing the instruction MME GEINOS. This instruction, which is followed by pertinent I/O parameters, as discussed in Chapter 11, gives control to the I/O Supervisor (IOS) which initiates activity on magnetic tape, drum, and disc units; remote terminals; card reader; card punch; printer; perforated tape reader/punch; and DATANET-30 processor. In addition, IOS provides the following capabilities:

- Symbolic-to-Physical Unit Translation. Users of the GE-625/635 refer to files symbolically (file codes) in the various source languages available within the software system. This enables maximum flexibility of assignment of files at execution time and permits device independence of the files upon which a program may call. The I/O Supervisor will convert the symbolic file designations to absolute physical assignments at execution time.

- File Protection. IOS ensures that one program may not refer to the files of another. This is achieved by using the program number and file code when searching the slave PAT table the first time and using file code PAT pointer in successive searches.

- Simulated Tape Processing on Disc/Drum. IOS provides a method of processing on disc and drum which simulates the serial mode of processing normally peculiar to magnetic tape. This mode of processing (linked) not only provides a degree of device independence to computer configuration but also allows the opportunity to reduce program setup time by eliminating the use of magnetic tape for some files.

- Supervision of Interrupts. IOS responds to all I/O interrupts and takes appropriate action. The programmer will normally not be required to concern himself with the interrupt system of the computer.

- Queueing of I/O Requests. IOS maintains queues of I/O commands. Many commands can be queued for each I/O device. For each linked file, only one command for that file will be queued at any time. In multidevice channel subsystems, such as magnetic tape and disc, IOS will not necessarily select queued commands for execution in the order in which they were requested. For each linked file, the command will be done in the requested order, however.

- Utilization of IOS with Crossbarred Magnetic Tape Subsystems. If the primary channel is busy at the time I/O is to be initiated, IOS will automatically select an alternate channel (if available) on a crossbarred magnetic tape subsystem.

- Pseudo Command. In the special case of the typewriter, a pseudo command is provided which will allow a programmer to type a message requiring an answer. Using this command guarantees that the answer to a question will follow in sequence behind its asking.

- Awareness. The GE-625/635 I/O equipment provides information to a central processor which allows IOS to maintain awareness of the status of each peripheral device. The IOS maintains tables which allow it to record such information as a magnetic tape is rewinding; some peripheral device is in a Power Off condition; or a magnetic tape handler is in Ready status or has been rewound and is in Standby status.

- Accounting. IOS keeps a complete record of the time spent by the central processor and each peripheral for every program executed. These statistics are later recorded on the accounting file and printed on the execution report for use in billing and analysis (see Chapter 10). Use of the I/O Supervisor is described in detail in Chapter 11. Implementation is described in GE-625/635 GECOS-III I/O Supervision, CPB-1494.

SYSTEM OUTPUT

Output media conversion is accomplished through Disc-SYSOUT. Data is stored on disc and handled by output media conversion directly with Disc-SYSOUT. As soon as the data is transferred to the output media, the disc is available for reuse as storage for additional data directed to the output destination.

SYSOUT line limits are as follows:

ALGOL	- 10,000	SYSEDIT	- 5,000	
(~~BMC~~)CONVERT	- 1,000	UTILITY	- 5,000	
COBOL	- 20,000	ILANG	- 10,000	?
EXTEDIT/FILEDIT	- 2,000	1401SIM	- 5,000	
FORTRAN	- 12,000	225SIM	- 5,000	
FILSYS	- 1,000	44SIM/94SIM	- 5,000	
GMAP	- 10,000	Execute	- 5,000	
IDS	- 10,000	CAFS		
JOVIAL	- 10,000			

The lines counted include only those supplied by the slave program, including GEBORT core dumps and GESNAP print lines. For every 300-word block of GESNAP or GEBORT lines, 2 lines of SYSOUT are counted for the panel and 1 additional line for each 300 data words. If less than 300 words are in a block, 1 line is still counted.

Implementation of the system output modules is described in GE-625/635 GECOS-III Termination and System Output, CPB-1496.

FILE SYSTEM MAINTENANCE

The heart of GECOS-III is a centralized file system of hierarchical tree-structured design. This design provides multiprocessor access to a common data base, full file protection, and maximum access control. The file system is comprised of a permanent on-line data base controlled by a master catalog which identifies each user. Within the master catalog are pointers to catalogs for every user. Thisuser's catalog may define files or may, in turn, name and point to still lower level catalogs.

The centralized file system is manipulated by a File System (FILSYS) activity. The use of this is described in GE-625/635 GECOS File System (FILSYS), CPB-1513. The implementation of the file system is described in GE-625/635 File System Maintenance, CPB-1497.

4. PROGRAM SWITCH WORD

The Program Switch Word contains the job switches, system options, and user switches, as shown in Figure 1.

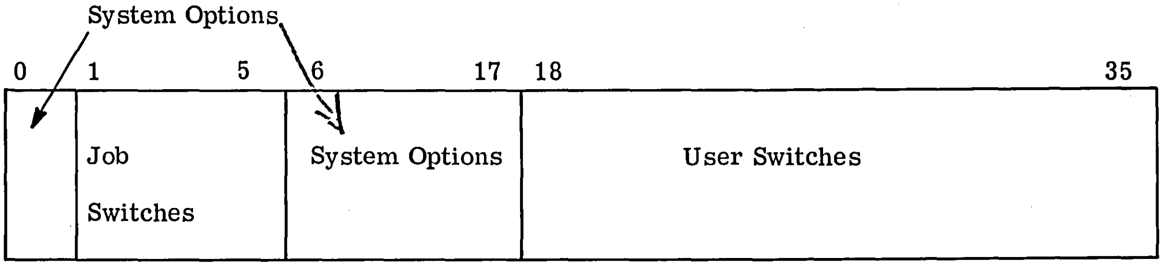


Figure 1. Program Switch Word

JOB SWITCHES

The Job Switches (Bits 1-5) are used to convey information about the entire job, and therefore the status (Bit = 1 or 0) is saved between activities.

At present, Bits 1, 2, 4, and 5 are the only ones used. Bit 1 is turned on (=1) when a GMAP card specifying the NXEC option is encountered. Bit 2 is turned on when a $ GMAP card specifying the SYMTAB option is encountered. Bit 4 is turned on (=1) when a $ REMOTE card is encountered, and Bit 5 is used to denote whether there is an execution activity within the job. If an execution activity exists, Bit 5=1; if none, Bit 5=0.

SYSTEM OPTIONS

System options are stored in Bits 0 and 6-17 of the Program Switch Word. The status of the bits is retained only while the activity is in execution. When the option is used, the bit is set to a 1. The options available with each software processor system are given in the discussion of the appropriate system control cards. (See Chapter 5.) The following options and the associated bits controlling them are given below.

Bit	Bit=1	Bit=0
0	DUMP Option	NDUMP Option
6	COMDK/0N1	NCOMDK
7	DECK/0N1	NDECK
8	LSTOU/0N3	NLSTOU
9	UPDATE/0N4	No UPDATE
10	LSTIN/0N5	NLSTIN
11	DEBUG, NGMAC, STAB/0N6	NSTAB
12	ABORT subactivity	No ABORT subactivity
13	GMAP/System Editor Interface	No GMAP Interface

Bit	Bit=1	Bit=0
14	COPY Option/FORTRAN- or COBOL-created G* File	NCOPY Option
15	GESAVE/PRODUCT	No GESAVE/No Product (see below)
16	EXTEND	No EXTEND
17	PRODUCT	No PRODUCT

These bits are set as each activity is allocated. Bits 0 and 6 through 11 are set after encountering a language processor control card ($ COBOL, $ GMAP, $ IDS, etc.) or a $ EXECUTE control card specifying options. (See Chapter 5.) Bit 12 is set when a $ ABORT card is encountered. The GMAP Interface (Bit 13) is set by the System Editor to establish a special interface between GMAP and the System Editor. (See also GE-625/635 System Editor, CPB-1138.) If either a $ IDS or $ COBOL control card specifies the COPY option, Bit 14 is set to 1. This bit is also used to indicate that a particular G* File was created by COBOL or FORTRAN. Bit 16 is set when a $ EXTEND card is encountered. (See Chapter 5.) Bit 15 is set to 1 by the user program (normally GELOAD) prior to the first access to MME GESAVE. This signals the initial access; thus, the file is rewound prior to its use. If a $ PRODUCT card is encountered with only a name specified, Bit 17 is set ON. If the $ PRODUCT card contains an FCB name in addition to a program name, Bits 15 and 17 are set ON.

USER SWITCHES

Like the Job Switches, the User Switches (Bits 18-35) are retained between activities of a job and are available for use by the programmer for whatever need he may have within his job. The user sets the switches by means of a MME GESETS or a MME GERETS. (See Chapter 7.)

Is there also a $ SET card that permits setting of the user switches?

5. CONTROL CARD FORMATS

The various control cards used by GECOS in the execution of activities of a job are described under the following categories:

- Basic Input Deck
- $ File Cards
- Assembler and Compiler Cards
- Loader Cards
- Utility Cards
- Call Cards
- Conditional Activity Cards

Typical deck setups are shown in Appendix A.

Although the many control cards are used for different functions, as indicated above, certain basic rules apply:

1. All control cards are identified by a $ in column 1.
2. The control card name (i.e., SNUMB, IDENT, DISC, etc.) begins in column 8. Exception: Conditional Activity Cards.
3. Variables begin in column 16.
4. Except where noted, the variables must not exceed column 72.
5. No more than six variables are allowed in the operand field (column 16-72), except where noted.
6. Each variable must be separated by a comma. Exception: $ IF card.
7. No imbedded blanks are permitted; a blank terminates the field definition. (A dash may be used for a spacer.) Exception: $ COMMENT card.
8. Variables which exceed 12 characters will be truncated.

In addition to the restrictions given above, each control card description is accompanied by restrictions concerned with the particular card.

BASIC INPUT DECK

The following control cards are used by Input Media Conversion (IMC) and comprise the Basic Input Deck:

- $ SNUMB
- $ IDENT
- $ COMMENT

- $ OBJECT
- $ DKEND
- $ EXECUTE
- $ LIMITS
- $ ENDJOB

The ***EOF card, although not an IMC control card, is also described in this section.

An example of a Basic Input Deck, utilizing the control cards mentioned above, is shown below.

In addition, a description of each card is given in the following pages.

```
1      8       16

$      SNUMB   00345
$      IDENT   00123,JOE-SMITH,DEBUG,TEST
$      COMMENT JOB IS FOR GECOS
$      OBJECT
         .  }
         .  }  Object Deck
         .  }
$      DKEND
$      EXECUTE
$      LIMITS
$      ENDJOB
***EOF
```

$ SNUMB

```
 1       8       16
|$      |SNUMB  |Job Identifier, Urgency
```

Function:

The $ SNUMB control card is used by IMC to identify the job internally and to assign an urgency to the job for use in allocating the job.

Operand Field:

The operand field contains two variables, the Job Identifier and the Urgency Level.

The Job Identifier is from one to five characters in length and must be present on the card.

The Urgency Level is a number from 1 to 63 (at present 40 is "threshold" level) and represents the relative importance of the job. If the urgency is omitted, a value, based on the system resources needed by the job, is calculated. Zero should not be used unless purposely intended as the job will be put "IN LIMBO" AND not considered for Allocation until the urgency is changed by the operator to a non-zero value.

Example:

```
 1       8       16
|$      |SNUMB  |00345,6
                 └─┬─┘ └┐
 Job Number 345 ───┘    │
 Urgency Level  6 ──────┘
```

Rules:

1. The $ SNUMB card must be the first card of every job.
2. The Job Identifier can be alphabetic, numeric, or alphanumeric.
3. The Job Identifier must not be blank, must not be zero (00000), or must not contain a comma in column 16. Detection of any of these will cause the job to be ignored.
4. If jobs are resubmitted to GECOS (previous abort, new test, etc.), the $ SNUMB card should carry a new Job Identifier.

$ IDENT

```
1      8       16
$      IDENT   Account No., Identification
```

Function:

The $ IDENT control card is used to identify the user of a job or activity and to supply accounting information. Each activity may be preceded by a $ IDENT card or this card may be used for a series of activities. If used for a series of activities, all accounting information for these activities will be combined under the same account number. The complete card image is printed as the first line of the execution report for each activity.

Operand Field:

The format of the Account No. is an option of the individual installation. However, it is recommended that the first field contain any alphanumeric account number not exceeding 12 characters. GECOS scans the field looking for a comma and then takes the next nine columns to create a banner for the execution report. If no comma is found before column 52, columns 52 through 60 are used for the banner.

The Identification subfield may contain up to 12 alphanumeric characters. When permanent files are accessed by a $ PERM, $ SELECT, or $ PRMFL card, this field is used to identify the user to the file system. If files are created, this becomes the identifier of the creator. When files are subsequently requested, the user identification must match.

Example:

```
1      8       16
$      IDENT   23233,JOE-SMITH
Account 23233
Requestor is Joe Smith
```

Rules:

1. At least one $ IDENT card must immediately follow the $ SNUMB card.

2. Account number must not exceed 12 characters and can be either alphabetic or numeric.

3. Columns 16-72 are available in Words 66_8-77_8 of the Slave Program Prefix during activity execution.

$ COMMENT

```
1                 8           16
|$                |COMMENT    |Any Comments
```

Function:

The $ COMMENT control card is used by the programmer to communicate to the operator through the on-line typewriter.

Operand Field:

The contents of the operand field are typed on the typewriter; trailing blanks are not typed. The typing takes place immediately before execution of the activity in which the $ COMMENT card occurs.

Example:

```
1                 8           16
|$                |COMMENT    |JOB IS FOR GECOS
                              \_______  _______/
                                      \/
 COMMENT------------------------------
```

Rules:

1. The $ COMMENT cards that appear in a deck before an activity is defined (i.e., preceding $ FORTRAN, $ GMAP, $ EXECUTE, etc.) are considered with that activity.

   The $ COMMENT cards that appear after an activity has been defined and before a control card defining a new activity will be considered part of the activity defined.

   Thus, $ COMMENT cards pertaining to an execution activity which follows a number of other activities should be located following the $ EXECUTE control card.

2. Use of hyphens as spacers is not required in the $ COMMENT card; the operand field is typed exactly as it appears on the card.

3. Each activity may have only one $ COMMENT card. If more than one $ COMMENT card is present only the last card encountered is typed.

4. The comment must not exceed column 72.

$ OBJECT

1	8	16 - 57	60	67	73
$	OBJECT	Remark	Previous Change Date	Current Change Date	Program Ident. No.

Function:

The $ OBJECT control card is used to identify the beginning of an object program or subprogram in the input sequence of a job and the date on which the program was assembled. The card is produced by the Assembler or Compiler as the first card of an object program or subprogram deck.

Operand Field:

The Remark is an optional comment produced as a product of the second subfield of the LBL pseudo-operation not exceeding 42 alphanumeric characters. You want it? / You put it in!

The Previous Change Date is the date the program was last assembled. It is optional and available for user convenience. you want it; you put it in.

The Current Change Date is the date of this current assembly. This date is supplied by the Assembler or Compiler.

Program Identification is an alphanumeric identification number designating the object program or subprogram and is a product of the first subfield of the LBL pseudo-operation which, if not specified, is produced at starting at 00000000.

1	8	16	60	67	73
$	OBJECT	LK	091665,	101265	. LP30000

Previous Change Date 9/16/65
Current Change Date 10/12/65
Object Program. LP30000

Rules:

1. The $ OBJECT card must be the first card of an object deck.
2. The dates are merely user conveniences which appear on the execution report and are not required.

$ DKEND

1	8	16	73
$	DKEND	Normally, not used (See below for exceptions)	Prog. Ident. No.

Function:

The $ DKEND control card is used to identify the end of an object program or subprogram in the input sequence of a job. This card is produced by the Assembler as the last card of all assembled or compiled programs or subprograms.

Operand Field:

Except for the Program Identification No., the operand field is normally not used. The two cases when it is used are given in the examples below.

Program Identification No. is the natural sequence number of the card and is generated by GMAP.

Examples:

1	8	16
$	DKEND	CONTINUE

When encountering CONTINUE, GELOAD does not switch to loading from the R* file but continues loading from B*.

In the second case, certain one-pass compilers generate a card with the following format:

1	8	16
$	DKEND	. ALODR

This card instructs GELOAD to perform certain maintenance functions during loading. When . ALODR is encountered, . ALGOL is searched for in the load table. Its definition point points to the length of the program, which determines the load address for the next program. Also, this location defines the starting point for a two-word entry postpreface table. The end of the table is signaled by having the first word of an entry be zero. After recording this information, GELOAD removes . ALGOL from the load table so that following decks of the same format may be loaded. The two-word entry postpreface table is then examined serially as follows:

1. If word 1 = 0, return to normal loading.

2. If word 1 $\neq$ 0, word 2 is treated as two elements:

   Bits 0 - 17 = a
   Bits 18 - 35 = k

a. For k = 0 or 1 (primary or secondary SYMDEF), the load table is searched for the name in word 1. If it is already in, it is ignored (or it is defined if it is undefined). If not in, the name is entered in the load table and a is treated as a relative address in the program just loaded.

b. For k = 2, error message is printed.

c. For k = 3, it is treated identical to k = 0 or 1; a, however, is regarded as absolute instead of relative.

d. For k = 4, error message is printed.

e. For k = 5 (SYMREF), the load table is searched for the name in word 1. If the name is in the table and is defined, the chain is defined in the program just loaded. If the name is in the table and not defined, the chain of the SYMREF is continued. If the name is not in the table, it is entered and the chain is started as a normal SYMREF.

f. For k = 6 (LABELED COMMON), error message is printed.

g. For 7 ≤ k < 2577, error message is printed.

h. For k = 2577 (forward reference), bits 0 - 17 of word 1 are treated as the beginning of a chain and the chain is filled in with the value of a.

i. For k > 2577, error message is printed.

Rules:

1. The $ DKEND card must follow an object program or subprogram.

2. The two exceptions discussed above may appear on the same card, separated by a comma, in situations which warrant this action (i.e. batch compiles generated by one-pass compilers requiring the .ALODR option).

$ EXECUTE

```
 1      8         16
|$      |EXECUTE  |Sense Switches, Blanks, and/or Dump Option
```

Function:

The $ EXECUTE is used to request the loading and subsequent execution of the object program.

Operand Field:

The operand field provides for the definition of from one to six Sense Switches which may be set to initialize the ON condition. These switches may subsequently be tested or modified within the object program via the MME GESETS or GERETS. (See Chapter 7.)

ON1 - Sense Switch 1 will be turned ON   Switchbit #6 is set

ON2 - Sense Switch 2 will be turned ON   Switch bit #7 " "

ON3 - Sense Switch 3 will be turned ON   " " #8 " "

ON4 - Sense Switch 4 will be turned ON   " " #9 " "

ON5 - Sense Switch 5 will be turned ON   " " #10 " "

ON6 - Sense Switch 6 will be turned ON   " " #11 " "

See Program Switch Word page 19

These Sense Switches are the same ones that are turned on when Assembler/Compiler options are specified. If omitted, the switch is assumed to be off.

The Dump Options (standard option underlined) are as follows:

DUMP - Slave core dump will be given if activity is terminated abnormally

NDUMP - Only program registers will be dumped if activity is terminated abnormally

Example:

```
 1      8         16
|$      |EXECUTE  |ON3,ON4,DUMP
```

Rules:

1. The $ EXECUTE control card must appear after all subprograms to be executed but before their data.
2. The options can be listed in any order in the operand field.
3. If the operand field is left blank, the switches are assumed to be off and NDUMP is implied.

$ LIMITS

```
 1      8        16
|$     |LIMITS  |Time, Storage 1, Storage 2, Print Lines
```

Function:

The $ LIMITS control card is used to extend standard activity limits. If omitted, allocation will be made as indicated under 'Default Allocation" in Chapter 12.

Operand Field:

Four entries are contained in the operand field.

Time, the first entry, specifies the maximum processor run time for the activity, expressed in hundredths of an hour. Beyond this limit, the job is aborted with an I8 code. If an activity calls upon a system program, (i.e., an object program calling SORT), this processing time must be included all the time estimate for the activity. The maximum value that can be punched in this field is 999.

Storage 1 denotes the maximum core storage requested for running the program. The units of storage are decimal digits representing the number of words desired. (Actual memory allocation is made in multiples of 1024 words). The smallest request that will be allocated is 3072 words. If the value is followed by the character "K," this value is multiplied by 1024 before its use.

Storage 2 indicates the amount of memory, requested by a slave program,that may be shared with GELOAD when the program is loaded. An explanation of the types of memory usage which may be shared with GELOAD is given below.

> An execution activity may be broken into two phases; loading and execution. During loading, GELOAD places information into preassigned areas of the user's memory. There may be other areas, such as LABELED COMMON regions, buffers, and BLANK COMMON into which no information is being loaded at load time but will be used by the program during execution. By proper use of control cards ($ USE, $ EQUATE, $ OPTION), all of these regions may be assigned to the low end (addresswise) of the user's memory such that the area which they represent may be shared with GELOAD during loading. (When execution begins, GELOAD (.SETU. routine) will have cleared this area for use by the program.)

A minus sign may precede the value specified in Storage 2. If indicated thus, the value specified is added to the size of GELOAD to allow extra space for load tables.

Print Lines specify the maximum number of lines to be written on SYSOUT during program execution for later printing.

Example:

```
1       8           16
|$      |LIMITS     |10,10000,0,1000
```

Maximum Run Time 0.10 hr.

Maximum Storage 10,000 words

No memory can be shared

Maximum of 1,000 lines written on SYSOUT

Rules:

1. In a "Compile and Go" or "Assemble and Go" operation, the $ LIMITS card following the $ EXECUTE card defines the limits for the anticipated execution of the user's program and not the compilation or assembly process.

2. The $ LIMITS card is not required for system programs such as FORTRAN or COBOL, since standard limits are predefined. If these limits are to be modified, however, the $ LIMITS card should follow immediately behind the system call card ($ FORTRAN, $ GMAP, $ COBOL, etc.).

3. The $ LIMITS card, if used, should follow immediately behind the $ EXECUTE card, or other system call card (see 2).

4. When the $ LIMITS card is omitted, the limits are as follows:

This picture duplicated on page 17

Activity	Hr.	(+SSA) Memory	Lines
ALGOL	0.08	28k	10,000
(BMC) CONVERT	0.08	6k	1,000
COBOL	0.15	32k	20,000
EXTEDIT/FILEDIT	0.04	32k	2,000
FORTRAN	0.05	24k	12,000
FILSYS	0.03	8k	1,000
GMAP	0.04	24k	10,000
JOVIAL > IDS	0.08	28k > 32k	10,000
SYSEDIT	0.08 > 0.05	32k	5,000 < 10,000
UTILITY	0.03	8k	5,000
? ILANG	0.16	45k	10,000
1401SIM	0.08	12k	5,000
225SIM	0.08	24k	5,000
44SIM/94SIM	0.08	48k	5,000
Execute caps	0.05	16k	5,000

5. Storage 2 may contain any number. However, if this number is less than .CRSZL (size of the Loader), the Allocator allocates the extra core needed to load the program. If the number is greater than .CRSZL, the Allocator reduces this number to .CRSZL. If a minus sign precedes the overlay value, the value is added to .CRSZL to allow extra space for load tables.

PROGRAM	0.50	20K	30,000
PRODUCT	0.50	20K	30,000

6. Fields can be explicitly nulled:

$ LIMITS 5,,,5000

which will result in:

$ LIMITS 5,16,0,5000

7. When the lines to be outputted by SYSOUT are limited due to the above specified circumstances the lines counted include only those supplied by the slave program. Banner lines, GEPR error messages, and the accounting report, are not counted. GEBORT core dumps and GESNAP print lines are counted as described in Chapter 3, "System Output."

$ ENDJOB

1	8	16
$	ENDJOB	Not Used

Function:

The $ ENDJOB control card is used to indicate that the job being processed is a candidate for allocation and execution, provided errors were not detected. If errors are detected, they are noted on the console typewriter as they occur; GEIN completes the processing; and the entire job is deleted without being allocated.

Operand Field:

Not used.

Example:

1	8
$	ENDJOB

Rules:

1. The $ ENDJOB must be the last card of every job.

***EOF

```
1
***EOF
```

Function:

The ***EOF card serves two purposes. During system initialization, the ***EOF card is placed at the end of the Startup Deck. When this card is detected, Startup examines the Module Name Table to see if all required modules have been loaded. If so, loading of GECOS is complete. If not, Startup loads the remaining required modules from the GECOS System Library on the System Storage Device. If any required modules cannot be found, their module names are added to the Memory Load Map with asterisks (******) in the address field, indicating that all modules could not be found. The second purpose of the ***EOF card is for use with the input/output (I/O) system package. The occurrence of a hopper empty status initiates a search by the I/O package to determine whether the last card read was an ***EOF card. If the last card before hopper empty was an ***EOF card, an EOF status is set by the I/O package, the card reader is released, and GEIN is released from memory. If the last card before hopper empty was not an ***EOF card, the operator is alerted that the hopper should be loaded.

Operand Field:

Not used.

Example:

```
1
***EOF
```

Rules:

1. The ***EOF card is an operation control card.
2. The ***EOF can only be used with card reader input.
3. The ***EOF card must be the last card before a hopper empty status is encountered.

FILE CARDS

File control cards are used to define file information pertaining to a program to be processed. The following control cards are classified as $ File control cards and are described in the following pages.

- $ DATA
- $ DISC (or $ DRUM)
- $ ENDCOPY
- $ FILE
- $ NTAPE
- $ PERM
- $ PPTP
- $ PPTR
- $ PRINT
- $ PRMFL
- $ PUNCH
- $ READ
- $ REMOTE
- $ REPORT
- $ SELECT
- $ SYSOUT
- $ TAPE (or $ TAPE9)
- $ TYPE

Card Placement in Deck

Within a control sequence, $ File cards must follow the control card that defines the activity and precede any data cards associated with the execution. All $ File cards associated with the execution of an object program must appear behind the $ EXECUTE control card, described in the preceding section entitled Basic Input Deck.

$ DATA

```
1       8        16
$       DATA     File Code, Options
```

Function:

The $ DATA control card is used to write files onto temporary linked disc or drum files. After termination of the activity for which this file was created, the file and all references to it will be removed from the computer system.

Operand Field:

The operand field contains a two-character, alphanumeric file code which is assigned by the programmer and used in referring to the file. This is the same file code which is given as subfield 2 in the file control block macro instruction. (See GE-625/635 File and Record Control, CPB-1003.)

The following Options are available. If not specified, those underlined options will be assumed.

CKSUM	-Binary checksums will be checked. If not used, this field must be nulled (see example below).
NCKSUM	-No checksums will be checked.
COPY	-All $ control cards (except $ SNUMB and $ DATA cards) will be copied on the file specified by the file code without checking by GEIN.
NCOPY	-No $ control cards will be copied.
TAKEc	-Card decks punched by non-GE card punches will be read. If a non-GE character (invalid) punch is encountered, the character following the TAKE option (c) will be substituted for the invalid character. The (c) character may be any legitimate GE-625/635 character.

Example:

```
1       8                16
$       DATA             CD,,COPY,TAKE+
```

The cards or card images following the $ DATA card and preceding the next $ control card will be put onto temporary, linked drum/disc file CD

CKSUM field is nulled

A + is substituted for any invalid character encountered

Rules:

1. The variable field must contain a file code. This code can be any two numbers or letters. An asterisk should not be used in naming the files since system files are designated symbolically by means of an asterisk in one of the two characters of the system file name.

   The $ DATA card must precede those data cards to be put onto the temporary file referred in the operand field. If the programmer wants GEIN to create a file for him, he must place his data cards after all other cards of the activity, except in cases where a $ DATA card is also in the job. In this case, the file created will be assigned to I* automatically, and the temporary file called with the $ DATA card will be assigned to the file specified in the operand field as shown in the example below.

```
$ SNUMB
$ IDENT
$ OBJECT
    ...  }
    ...  }    Object deck
    ...  }
$ DKEND
$ EXECUTE
$ LIMITS
$ TAPE
    ...  }    Data.  These will be put on the I*
    ...  }
$ DATA AB
    ...  }
    ...  }    These will be put on temporary
    ...  }    linked file AB
$ ENDJOB
```

2. Input files created by Input Media Conversion (namely, I* and any other file designated by a $ DATA control card) are recorded in standard system format. Upon encountering a new activity or a $ control card, an end-of-file is automatically appended to these files.

3. A separating comma must be used if the second field (CKSUM option) is omitted and the third field (COPY option) is present.

4. The last data card of the file (specified on the $ DATA card) to be copied must be immediately followed by a $ ENDCOPY card or a $ SNUMB card.

5. The TAKE option, if used, must be in the fourth subfield.

$ DISC, $ DRUM, OR $ FILE

1	8	16
$	DISC	File Code, Logical Unit Designator, Access
$	DRUM	File Code, Logical Unit Designator, Access
$	FILE	File Code, Logical Unit Designator, Access

Function:

The $ DISC (or $ DRUM) control card is used to set up allocation of a file on a disc or drum. The $ FILE control card is used to set up allocation of a mass storage file on the fastest device available.

In addition, these cards may be used to assign a reserved ("named") disc/drum unit, as described in Appendix E.

Operand Field:

The operand field contains three fields.

The File Code is a unique 2-character alphanumeric designation assigned by the programmer to identify a given file within his activity. This is the same file code which is given as subfield 2 in the file control block macro instruction. See GE-625/635 File and Record Control, CPB-1003.)

The Logical Unit Designator contains a 3-, 4-, or 5- character designation of the file (software) and file unit (hardware). The first character is the symbolic file unit designator. This designation can be any letter. Allocation is rotated among available devices. The second portion of the Logical Unit Designator is numeric (one or two digits, not exceeding 62_{10}). The numerals have no specific meaning except that, for any one file unit, the quantity of different numerals is equal to the number of files requested on that file unit. The third portion is the disposition code. The following are the codes used:

R - Release

S - Save for a subsequent activity

Two disposition codes may be used. If used, the second code indicates desired disposition if the activity is aborted. Thus, codes SR instruct the system to save the file if the activity terminates normally but release the file if the activity aborts.

If saved (S) for a subsequent activity, the disc or drum is assigned to files with the same file unit designator in subsequent activities of this job. If a D (Dismount) is inadvertently included, this will be interpreted as Release (R) since dismounting does not apply to disc or drum files but only applies to tape files. If a C (Continue) is inadvertently included, it is changed to an S (Save).

The Logical Unit Designator is required only when a S (Save) disposition is used with the specified file. A null field implies R (Release) disposition.

Access contains a numerical and letter designation. The numerals designate the number of 3840-word links allocated to the file. The numerals are followed by either an L, meaning a linked file, or an R, meaning a random file. A discussion of these files is contained in Chapter ~~12~~ 11. Omission of the Access field implies one 3840-word linked link. If the letter is omitted, an L (linked file) is assumed.

If the programmer wished to access a file on a disc or drum that has been allocated and saved by a previous activity of the job, he is not permitted to alter the type (linked or random) or the size of the allocated file from that which was specified on the saved file card in the previous activity. Altering the file type will cause a new file to be allocated.

Example:

```
 1      8        16
|$      |DISC    |AB,X11S,5L
```

Designation identifying file AB

Unit, file 11, which will be "saved" for a subsequent activity

Five 3840-word links allocated to file AB; these are "linked" files

Rules:

1. A disposition code should be indicated; if not, an R is assumed.

2. When a file is designated in the Logical Unit Designator as "saved" for a following activity, the programmer must insert a file card with the same Logical Unit Designator in the subsequent activity before the activity is used. The file type and size must not be altered.

3. After a specific file has been declared a "saved" file, the designator used for that file may not be used to designate another file within that job.

4. Multiple files, using the same Logical Unit Designator, on disc or drum are not permitted. The following is an example of this illegal combination:

```
 1      8        16
|$      |DISC    |AB,X1R,5L
|$      |DISC    |AC,X1R,3L
```

Both files are requested in the same activity. Job will be aborted.

This combination is now treated as a single file with two file codes. (5L is used; 3L is ignored)

5. If the number of links is blank or zero, the job will be deleted by Input Media Conversion.

6. When a file is stored on a disc or drum as a random file, all references to elements of the file by the activity 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 ensure that a specified relative address is within the allocated bounds.

7. When a file is stored on a disc or drum as a linked file, appropriate chaining is provided from one 320-word record of the file to the next record by GECOS.

8. Normally, there are twelve 320-word records in a link.

$ ENDCOPY

1	8	16
$	ENDCOPY	Not used

Function:

The $ ENDCOPY control card is used to turn off the switch which allows control cards to be written to a specified file without checking by GEIN (see also $ DATA).

Operand Field:

Not used

Example:

The following is a deck setup using the $ ENDCOPY control card:

```
$ SNUMB
$ IDENT
$ OBJECT
    . . .  } Object Deck
    . . .
$ DKEND
$ EXECUTE
$ LIMITS
$ DATA AB, , COPY
    . . .
$ ENDCOPY
$ ENDJOB
```

Rules:

1. The $ ENDCOPY must be preceded by a $ DATA card specifying the COPY option.

2. The $ ENDCOPY must follow the last data card of the file specified in the previous $ DATA card.

$ NTAPE

1	8	16
$	NTAPE	File Code, Channel Designator, No. of Tapes

Function:

The $ NTAPE control card is used by GEIN to request a number of utility tapes to be assigned to the tape channel specified in the Operand field.

Operand Field:

The operand field contains three fields.

The File Code is a two-character, alphabetic or alphanumeric designation assigned to the first tape by the programmer. Each subsequent tape will be assigned by GECOS and will be equal to the file code of the previous tape plus one (binary addition).

The Channel Designator is the designation indicating the symbolic tape channel on which the tapes are to be allocated. Allocation is rotated among available handlers on all tape PUB's.

No. of Tapes specifies the number of utility tapes to be allocated.

Example:

1	8	16
$	NTAPE	U1,A,3

File U1 is the designation assigned
by the programmer to the first tape.
The next tape will be labeled U2; next, U3.

Tape channel A

Three tapes

Rules:

1. All three variables in the operand field must be present.
2. The tapes assigned can not be input tapes and can not be passed on to following activities.
3. A channel designator of "X" may be used with $ NTAPE.
4. Used only in assigning 7-track tapes.

$ RPERM for removable disc pac files (DSU-160)

$ PERM

1	8	16
$	PERM	File Code, File Name, Subfile, Access

Function:

The $ PERM control card is used to access a permanent data file previously reserved by FILSYS activity or by Startup. Similar to $ DISC/DRUM in that it refers to data files, the $ PERM card refers to the data files which will reside permanently within the system, whereas the data files accessed by the $ DISC/DRUM cards are released at job termination.

Operand Field:

The File Code specifies the file code for the portion of the file on the first disc/drum unit allocated. If the file in the catalog resides on more than one disc/drum, each subsequent device will be assigned a file code equal to one (binary addition) plus the file code of the previous disc.

The File Name must be present and must agree with the name given to the file in the permanent file catalog.

The Subfile is a one- to six-character name denoting some portion of the entire file. Subfiles are defined during a FILSYS activity and are used to partition an ID·S file. If this argument is null, the subfile is considered to be the entire file. If less than 6 characters are used, the subfile name will be filled with trailing blanks.

Access describes the type of access the user wants for the IDS subfile. The valid arguments are:

UPDATE - The user requests exclusive access to this subfile for purposes of updating records in this subfile.

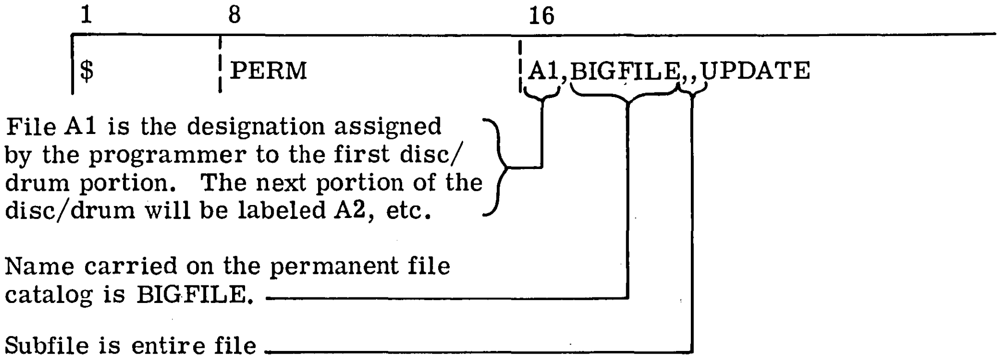
RETRVE - The user requests non-exclusive access to this subfile for purposes of retrieving records in this subfile.

RECVRY - The user requests access to an aborted subfile for purposes of recovery.

A null field for the access will allow the user to have non-exclusive access (implied RETRVE) to that portion of the file that he has requested.

The presence of an invalid argument for access should produce a job delete.

Example:



```
1      8                  16
$      PERM               A1,BIGFILE,,UPDATE
```

File A1 is the designation assigned by the programmer to the first disc/drum portion. The next portion of the disc/drum will be labeled A2, etc.

Name carried on the permanent file catalog is BIGFILE.

Subfile is entire file

Rules:

1. The file code and file name must be present in the operand field.

2. The File Name must agree with the entry in the permanent file catalog for the file.

3. The File Name must not exceed 12 characters.

4. A permanent file may be split over several physical disc or drum units.

$ PRMFL

```
1      8       16
|$     |PRMFL  |File Code, Permit, Mode, File String
```

Function:

The $ PRMFL control card is used to access a permanent data file previously reserved by a FILSYS activity (described in GE-625/635 File System (FILSYS), CPB-1513).

Similar to the $ DISC/DRUM cards in that it refers to data files, the $ PRMFL control card refers to data files which reside permanently in the system.

Operand Field:

The File Code specifies the 2-digit file code for the portion of the file on the disc/drum unit allocated.

The Permit indicates permit desired for the operation to be performed:

R - Read
W - Write
A - Append
E or X - Execute

If multiple permissions are desired, they are separated by slashes (see example below).

The Mode specifies whether the file is linked (sequential) or random. The letter S indicates linked (sequential); the letter R indicates random.

The File String is as defined in the file system descriptor and includes the catalog, the password, and the file name that describes the specific file. Each successive name is separated from its file name by a slash. If there is a password associated with the catalog, subcatalog, or file, it follows the catalog, subcatalog, etc., and is separated from it by a $ sign:

. . . . Catalogl$Password/Catalog2$Password/FILENAME$ PASSWORd

Example:

```
1      8       16
|$     |PRMFL  |H*,R/W,R,RANKIN/XDELE
```

$ PPTP, $ PPTR, $ PRINT, $ PUNCH, $ READ, $ TYPE

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
$	READ	File Code, Logical Unit Designator
$	TYPE	File Code, Logical Unit Designator

Function:

The $ control cards with PPTP, PPTR, PRINT, PUNCH, READ, or TYPE in the control field (starting in column 8) are used to allocate the following peripherals:

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

GECOS always allocates the typewriter file codes T*, *T, T/, and /T; the user need not use the $ TYPE card for allocation of these files.

These cards may also be used to allocate a reserved, or "named device," as described in Appendix E.

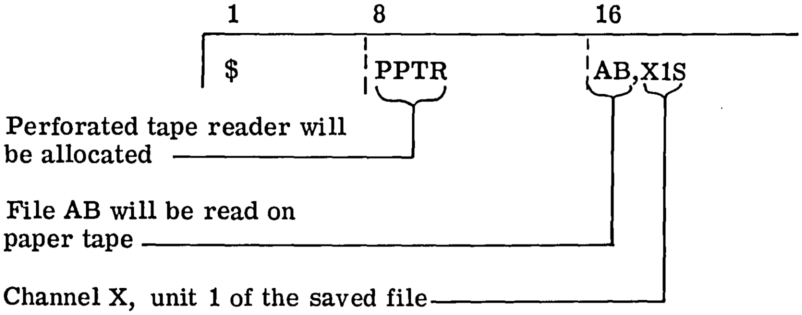
Operand Field:

The File Code contains the two-character, alphabetic or alphanumeric designation assigned to the file which will be printed, read, typed, etc. This is the same file code which is given as subfield 2 in the file control block macro instruction. (See GE-625/635 File and Record Control, CPB-1003.)

The Logical Unit Designator contains a three- or four-character designation of the initial unit and channel assigned to the file to be printed, read, typed, etc. (See $ DISC Card.) The same Disposition code specified on the $ DISC or $ TAPE card, is also included. If the Disposition Code is omitted or a D (Dismount) is given, an R (Release) Disposition Code will be assumed. If a C (Continue) is given, it will be changed to S.

The second field may also contain a Name and Disposition code, instead of the normally used Logical Unit Designator, if the card is being used to allocate a named device (see Appendix E).

Example:



Perforated tape reader will be allocated

File AB will be read on paper tape

Channel X, unit 1 of the saved file

$ REMOTE

```
1      8       16
|$     |REMOTE |File Code, Destination
```

Function:

The $ REMOTE control card is used to assign an output file to a remote station for use by the GE-625/635 Remote Terminal Supervisor (GERTS).

Operand Field:

The File Code entry is the standard two-character, alphanumeric designation assigned by the programmer to uniquely identify a file. System output files (such as P*) may also be used.

The Destination is a standard GERTS two-character, alphanumeric station number uniquely identifying the station to which the file should be routed. A blank destination field specifies the originating station.

Example:

```
  1      8       16
  |$     |REMOTE |P*, 26
P* System File will be routed
to GERTS station 26
```

Rules:

1. If the destination field is blank, the station identification code that appears in the $ SNUMB card is used. However, if a job is submitted remotely, all SYSOUT is automatically returned to the originating station. Thus, a blank Destination field is redundant.

$ REPORT

```
 1      8       16
|$      |REPORT |Report Code, Device Code
```

Function:

The $ REPORT control card is used to (1) change the standard paper on the printer to a special form or (2) redirect output from a normal device to a different device.

Operand Field:

The Report Code defines the type of report and may be as follows:

Report Code	Report Type	Media	Code
06	FORTRAN Logical Unit 6	Printline image	03
74 ~~(%)~~	Execution Report Assembler/Compiler Report Load Map	Printline image	03
75	Compressed Deck	Binary/Hollerith Card Image	01
76	Object Deck	Binary/Hollerith Card Image	01/02
77	ALTER Input List	Printline image	01

The Device Code is the code of the device to which output is directed. At present, only one code may be used:

Device Code	Device
PR	Printer

If device is PR, a Paper Form number and a Ply Number may also be used. The paper form may be from 1 to 12 characters. The ply number may be a 1- or 2-digit ply number.

Example:

```
 1      8       16
|$      |REPORT |6, PR, FORMABC, 2
```

Rules:

1. This card is specified for one report for one activity.

2. At present, only the PR device code is available.

$ SELECT

```
 1      8         16
|$     |SELECT   |File Name
```

or:

```
 1      8         16
|$     |SELECT   |Catalog Name 1 $ Password 1/File Name n $ Password n
```

Function:

The $ SELECT control card instructs IMC to call the File System, which searches the permanent file catalog and locates the required catalog entry. IMC will continue processing its input from the permanent file just as it had from the normal input device. If the catalog entry does not exist for the file card, the job will be deleted after the remainder of the deck has been processed in search of errors.

Operand Field:

The operand field contains the File Name (maximum of 12 characters) of the file which GEIN will begin reading and interpreting as soon as the $ SELECT card is detected. The selected file will be used in lieu of the normal input device until its end-of-file condition is encountered. The input will then revert back to the normal input and continue processing the input stream (see example below). The File Name is the identification of the disc or drum file when the catalog entry was made for this file.

Either of the two examples shown below may be used. If the first is used, the master catalog entry is equated to PERM/File Name. However any GECOS-II permanent files must be put into GECOS-III format. This is done by a GECOS-II catalog maintenance SAVE. This saved file must then be loaded under GECOS-III Startup.

Example:

The following is an example of the use of the $ SELECT card.

Input Stack through GEIN

```
$ SNUMB    001B
$ IDENT    00141, JOE-SMITH, RUN-B
$ OBJECT
  . . .
  . . .      Object Deck
  . . .
$ DKEND
$ EXECUTE
$ LIMITS    5,40000,,100
$ TAPE      AB,A1S,,123,1,FILE-B
  . . .
  . . .
```

```
Input Stack through GEIN                         Subprogram Called from the Permanent File

. . .
. . . Data Cards
. . .
$ SELECT RUNTB
        |
        +----------------------------------------+
                                                 |
                                                $ OBJECT
                                                  . . .
                                                  . . .        Object Deck
                                                  . . .
                                                $ DKEND
                                                $ EXECUTE
                                                $ LIMITS   5, 10000,,400
                                                $ TAPE     AB, X1D,,99999
                                                $ TAPE     BB, X2D,,99999
                                                  (end-of-file)
                             +-----------------------------+
$ TAPE       AB, 1D,,123,1,FILE-B }            These file control cards will
$ TAPE       BB, 2D,234,1,END-B   }  --------  replace those on the drum for
$ENDJOB                                        files AB and BB. Note: File code
                                               must be identical; logical unit
                                               designator should be identical.
```

The information called from the drum/disc may consist of any legitimate program input, including control cards. However, the programmer may change any control card by including a replacement, referenced by file code on a $ file card in the input deck after the $ SELECT card (see use example above). The $ control cards called from the drum or disc are replaced with those $ control cards in the input deck that have identical file codes. However, control cards can only be replaced within the same activity. A $ DATA card cannot replace a $ DISC card.

Rules:

1. Control card "nesting" (use of a $ SELECT card within a Select activity) is permitted to a depth of 10.

2. The information on the permanent file must be in standard system format.

3. A $ DATA card in the input stack cannot override a $ DISC card on a permanent file.

4. A $ ETC card may be used to continue the operand field.

5. In the file name string, the first entry in the string must specify the catalog name.

$ SYSOUT

```
 1            8          16
|$           | SYSOUT   | File Code
```

Function:

The $ SYSOUT control card is used by GEIN to assign output files to system Output Media Conversion. Unless otherwise defined, the P* file is automatically assigned to SYSOUT. This control card permits additional files to be so assigned. An example of its use follows. Activity A produces a large printed report which exceeds the limits of SYSOUT, so it is recorded on tape with file code PR for subsequent printing via Bulk Media Conversion. During checkout, the size of this report is small and does not warrant an extra conversion activity. Therefore the $ SYSOUT PR card is used to process the report by SYSOUT.

Operand Field:

The File Code is the designation used by the programmer to identify the file to be assigned to SYSOUT.

Example:

```
 1            8          16
|$           |SYSOUT    |AB
```

File AB will be put on SYSOUT

Rules:

1. This control card can only be used for files recorded in standard system format with the appropriate media codes and report codes included in the logical records.
2. It may only be used for files written through GEFRC.
3. There is no restriction on the number of files which may be assigned to SYSOUT.
4. The file code cannot be 00; if 00, the job will be deleted and a J4 error code will be issued.

$ TAPE OR $ TAPE9

1	8	16
$	TAPE	File Code, Logical Unit Designator,
	or	Secondary Logical Unit Designator, File
$	TAPE9	Serial No., Reel Sequence No., File Name

Function:

The $ TAPE and $ TAPE9 control cards are used to assign 7-track and ASA 9-track tape units, respectively.

In addition, these cards may be used to assign a reserved ("named") tape unit, as described in Appendix E.

Operand Field:

The File Code is a unique two-character, alphanumeric designation assigned by the programmer to 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. See GE-625/635 File and Record Control, CPB-1003.)

The Logical Unit Designator is necessary if files are to be saved or dismounted; it contains a 3-, 4-, or 5-character designation of the initial unit and channel assigned to the file. The first character is the symbolic channel designator. This designation can be any symbolic letter. Device requests will be allocated cyclically, one device to a channel, until requests are satisfied. The second portion of the Logical Unit Designator is numeric (one or two digits, not exceeding 62_{10}). The numerals have no special meaning except that, for any one channel, the quantity of different numerals is equal to the number of files requested on that channel. The third portion is the Disposition Code. The following are the codes used:

At activity termination, one or all of the following occurs.

S - Save
(1) No specific de-allocation ~~termination~~ occurs;
(2) unit is assigned to files with the same LOGICAL ~~channe~~l-unit designator in subsequent activities of the 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 Logical ~~channe~~l-unit designator in subsequent activities of the job.

D - Dismount
(1) Tape is rewound using "rewind and standby;"
(2) operator is issued dismount instructions;
(3) tape unit is released back to GECOS for re-assignment.

R - Release
(1) Tape is rewound using "rewind" instruction;
(2) dismounting instructions are not to be issued;
(3) tape is released back to GECOS for re-assignment.

Two disposition codes may be used. If used, the second code indicates desired disposition if the activity is aborted. Thus, codes SR instruct the system to save the file if the activity terminates normally, but release the file if the activity aborts.

The second field may also contain a Name and Disposition Code, instead of the normally used Logical Unit Designator, if the card is being used to allocate a named device (see Appendix E).

The Secondary Logical Unit Designator is any 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 multireel file is not used, this variable is omitted. However, the comma denoting end-of-field must be included if other fields follow.

File Serial No. is the tape serial number of the first reel of the file and is used as the primary input file label checking parameter. When this value cannot be specified in advance for the 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 an operator if this field is present.

Reel Sequence No. 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. A null value will be assumed to be equivalent to a 1.

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.

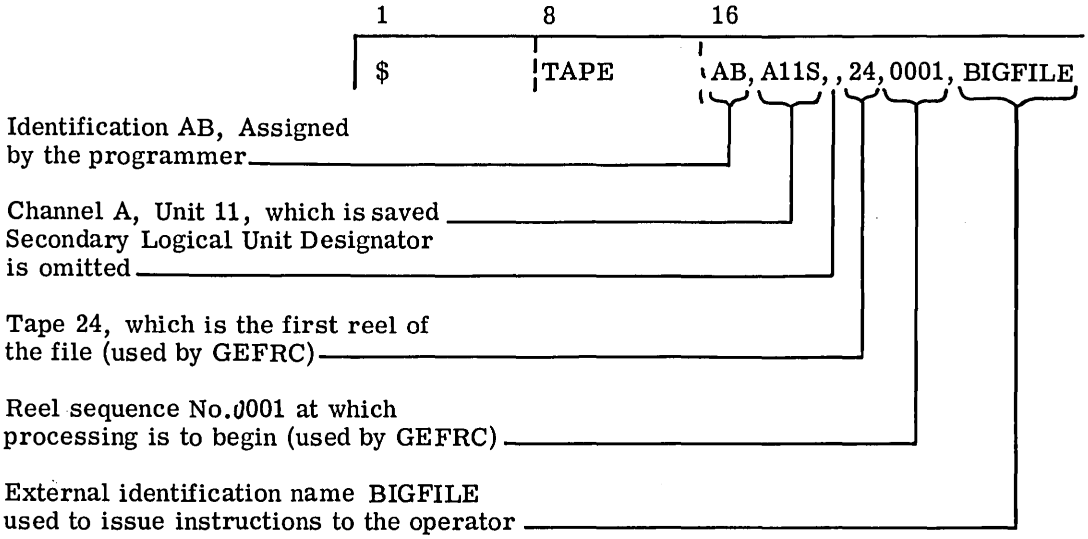
NOTE: The presence of the File Serial No. will cause a standby tape to be allocated and mounting instructions issued by GECOS.

The following is a summary of the tape handler status and the console messages associated with the occurrence of a File Serial No. and/or File Name.

		Allocator scans,	Console	Operator
File Serial No.	File Name	in order, for	Message	Action Required
None†	None	Ready	None	None
		Standby	*RDY	Ready Tape
None†	Yes	Ready	None	None
		Standby	*RDY	Ready Tape
Yes	Not examined	Standby	*MNT	Mount and Ready Tape
		Ready (Rewind Standby Issued)	*MNT	

† Write permit ring must be installed otherwise a *RNG message is issued

Example:



```
1      8       16
$      TAPE    AB,A11S,,24,0001,BIGFILE
```

Identification AB, Assigned by the programmer

Channel A, Unit 11, which is saved

Secondary Logical Unit Designator is omitted

Tape 24, which is the first reel of the file (used by GEFRC)

Reel sequence No.0001 at which processing is to begin (used by GEFRC)

External identification name BIGFILE used to issue instructions to the operator

Rules:

1. Where specified (see Operand Field) that the variables can be omitted, commas must be supplied to denote the missing fields.

2. There must be a $ TAPE control for each tape file to be used within the activity.

3. If the Disposition Code is omitted, a "D" (Dismount) is assumed.

4. The user should be sure that the last activity using a given tape specifies the proper Disposition Code ("R" or "D") for that tape. An "S" or "C" Disposition Code in the last activity results in a *DMT message only when the reel is write inhibited.

5. When a File Serial No. is not present, a write permit ring must be installed before a unit is considered ready. Failure to install the write permit ring will result in a *RNG message.

LOADER CARDS

The control cards used to instruct GECOS and the General Loader (GELOAD) include the following which are described in the following pages

- $ DUMP
- $ ENTRY
- $ EQUATE
- $ FFILE
- $ LIBRARY
- $ LINK
- $ LOWLOAD
- $ NOLIB
- $ OPTION
- $ RELCOM
- $ SOURCE
- $ USE

$ DUMP

1	8	16
$	DUMP	Name

Function:

The $ DUMP control card is used to signal GELOAD that the following cards should be processed for use with the DEBUG subroutine.

Operand Field:

The Name is the first primary SYMDEF of the subprogram to be debugged.

Example:

1	8	16
$	DUMP	START

Subprogram START will be debugged

Rules:

1. The $ DUMP control card must precede all other cards of the program or overlay except the $ LOWLOAD card.
2. At least one DEBUG card (see GE-625/635 General Loader, CPB-1008) must follow the $ DUMP control card.

$ ENTRY

1	8	16
$	ENTRY	Name

Function:

The $ ENTRY control card is used to denote the SYMDEF (Primary or Secondary) in the program at which entry is made to execute the program or subprogram. In the overlay mode, this card refers to the current link (see following discussion of the $ LINK card).

Operand Field:

The operand field contains the SYMDEF Name where entry is to be made.

Example:

1	8	16
$	ENTRY	LOCI

Entry will be made at LOCI
in the program or subprogram

Rules:

1. If no $ ENTRY control card is present, the entry is made at symbolic location. . . . . (the entry point for FORTRAN main routine) if. . . . . has been defined in loading as a Primary or Secondary SYMDEF or the entry is made at the first defined primary SYMDEF of the first subprogram loaded.

$ EQUATE

```
1      8       16
$      EQUATE  Equates
```

Function:

The $ EQUATE control card permits defining new SYMDEF's by SYMDEF's previously defined, defining new SYMDEF's relative to previously defined SYMDEF's, and equating Labeled Common regions relative to Blank Common.

Operand Field:

The operand field contains Equates in the form of SYMDEF names. The new SYMDEF's which are equated are enclosed in obliques, as shown in Example 1 below. If a new SYMDEF is defined relative to a previously defined SYMDEF, the relative position is enclosed in parentheses, as shown in Example 2 below.

Examples:

```
1.  1      8       16
    $      EQUATE  NAME1/NAME2, NAME3/, NAME4/NAME5/
```

Example 1 defines SYMDEF's NAME2 and NAME3 as SYMDEF's with the equivalent location of SYMDEF NAME1. The location of SYMDEF NAME5 is equivalent to the location of SYMDEF NAME4.

```
2.  1      8       16
    $      EQUATE  NAME (10)/NAMED/
```

Example 2 defines a new SYMDEF NAMED relative to SYMDEF NAME. The relative increment 10 is enclosed in parentheses. Note: A negative value is not permitted. The location of NAMED is set equal to the location of NAME + 10.

```
3.  1      8       16
    $      EQUATE  .CMN./LC1/, .CMN.(100)/LC2/
```

Example 3 defines the location of Labeled Common region relative to Blank Common. The GELOAD symbol .CMN. is a standard symbol synonymous with the beginning of Blank Common. The Labeled Common region LC1 is assigned an address equal to the beginning of Blank Common. Labeled Common region LC2 is assigned an address equal to the beginning of Blank Common plus 100. The length of Blank Common (which is printed with the memory map) is adjusted to represent the new length if the Labeled Common regions have extended the total.

Rules:

1. Commas are required between sets of equates, as shown by the arrow below.

```
1       8           16
$       EQUATE      NAME/NAME1,NAME2/,NAME3/NAME4/
                                     ↑
```

2. Subscripts are permitted as part of the definition only:

```
1       8           16
$       EQUATE      NAME(5)/NAME1/
```

3. The SYMDEF NAME used to define the new SYMDEF NAMES must have been previously defined by appearing as a SYMDEF or Labeled Common in a preface card. If any of the new NAMES (NAME2 and NAME3 in Example 1 above) have been previously defined, they are redefined and a nonfatal error message is printed on the execution report.

$ FFILE

1	8	16
$	FFILE	File Code, Parameters

Function:

The $ FFILE control card is used by GELOAD to describe nonstandard file control blocks (FCB's). One FCB is created for each $ FFILE card in the deck.

Operand Field:

The operand field contains a File Code and various Parameters used in describing the FCB's. The File Code is any two-character alphanumeric designation used by the programmer to uniquely identify the file. The Parameters given in the following can be used.

Field	Word and Bit Positions Altered in File Control Block (FCB)	Function
STDLBL	Word (-5) Bit (24)	Create or verify standard labels. Unless NSTDLB is specified, the STDLBL will be assumed
NSTDLB	Word (-5) Bit (24)	Do not create or verify standard labels
NBUFFS/n	Word (-5) Bits (26-27)	Assign "n" (0, 1, or 2)
BUFSIZ/nnn	Word (+4) Bits (0-17)	Set buffer size=nnn where nnn is a decimal number
LGU/(II,JJ,..)	Not FCB	Logical files II,JJ,.. (as many as desired) are on the same device as logical file FC, where II,JJ, etc. are 2-digit numerics ≤ 43
RETPER/nnn	Word (-8) Bits (0-17)	Set retention period = nnn where nnn is a decimal number ≤ 999
MLTFIL	Word (-5) Bit (25)	This is a multifile device
MODBCD	Word (0) Bits (20-21)	Set recording mode to BCD
PTMODS	Word (0) Bits (20-21)	Set recording mode for single-character paper tape
PTMODD	Word (0) Bits (20-21)	Set recording mode for double-character paper tape
PTMODE	Word (0) Bits (20-21)	Set recording mode for paper tape edit
MODMIX	Word (0) Bits (20-21)	Set recording mode for mixed (binary and BCD card files)
ASA9	Word (0) Bits (20-21)	Set recording mode for 9-track tape commands

FIXLNG/nn	Word (0) Bits (18-19) Word (+1) Bits (0-17)	The records are a fixed length of nn, where nn is a decimal number
NOSRLS	Word (-5) Bit (23)	Do not use block serial numbers
LODENS	Word (0) Bit (22)	Set device to low density
IGNORE	Not FCB	The user wishes to specify his own FCB. GELOAD will not create an FCB for this file code
PREHED/SYMDEF	Word (-14) Bits (0-17) Word (-8) Bit (18)	SYMDEF is the entry to a preheader label checking routine supplied by user
POSHED/SYMDEF	Word (-13) Bits (0-17) Word (-8) Bit (18)	SYMDEF is the entry to a postheader label checking routine supplied by user
PRETRL/SYMDEF	Word (-12) Bits (0-17) Word (-8) Bit (18)	SYMDEF is the entry to a pretrailer label checking routine supplied by user
POSTRL/SYMDEF	Word (-11) Bits (0-17) Word (-8) Bit (18)	SYMDEF is the entry to a posttrailer label checking routine supplied by user
ERRXIT/SYMDEF	Word (-5) Bits (0-17)	SYMDEF is the entry to an error analysis
MIXLNG/SYMDEF	Word (0) Bits (18-19) Word (+5) Bits (0-17)	SYMDEF is the entry to a user supplied routine to determine length of records being read. Bits in word (0) are set to indicate mixed length records.
DSTCOD/(PRNTR,MTAPE,...	Not FCB	Final destination codes for each of the logical units specified in LGU/(....) field. These are used only with programs generated by the ALGOL Compiler

Fields recognized with DSTCOD:

Device	Mnemonic
Printer	PRNTR
Card reader (binary)	BCRDR
Card reader (BCD)	DCRDR
Card reader (Mixed Mode)	MODMIX
Card punch (Binary)	BCPNCH
Card punch (BCD)	DCPNCH
Magnetic tape	MTAPE
Disc	DISC
Drum	DRUM
Paper tape mode - single character	PTMODS
Paper tape mode double character	PTMODD
Paper tape mode - edit	PTMODE

PHYREC

Informs UTILITY to handle physical records rather than logical records. UTILITY determines the mode (except ASA9, which must be specified), density, and labeling of the input magnetic tape file. Physical tape records are processed by DCW's generated within UTILITY.

NDATE

Informs UTILITY (when using the COPY function for labeled magnetic tape to labeled magnetic tape) to use the current (new) date as the creation date of the output file. If this option is not specified, the creation date will be the same as the creation date of the input file.

RANDOM/n

Informs UTILITY this input file is a random disc/drum file which can be dumped (DUMP or DDUMP) or skipped (SKIP). Records are a fixed length of n words (n ≤ 1280). If n = 0, record size is assumed to be 40 for disc; 64 for drum.

Example:

The following example (dumping an H* file) shows the use of the $ FFILE card.

```
1      8        16
-----------------------------------------------
$      ABORT
$      FFILE    H*,NSTDLB,NOSRLS,PHYREC
$      FUTIL    H*,,RWD/H*/,DUMP/1F/
```

Rules:

1. The $ FFILE card must be located after the $ EXECUTE control card when used with a GELOAD activity ($ OPTION must be included) or after the $ UTILITY card when used with a UTILITY activity. When used with an ABORT subactivity, the $ FFILE card must follow the $ ABORT control card. The UTILITY activity and ABORT subactivity control cards are described in the section entitled, "Utility Cards."

2. The Parameters need not be given in any specified order.

3. The $ OPTION card is used to specify FCB and is required when a $ FFILE card is used with a GELOAD activity.

4. If the PHYREC option is not specified, UTILITY will handle logical records.

5. The PHYREC option cannot be used for disc and drum files.

6. The PHYREC and NDATE options may be used only with a UTILITY activity; the DSTCOD/...option may be used only with ALGOL. If these are used in any activities other than those specified, the user will be aborted.

7. Processing of a given random file (via RANDOM option) must be completed on one $ FUTIL card since all open files are closed at the completion of a $ FUTIL card. It must also be completed before another file is started.

$ LIBRARY

```
 1        8          16
| $       |LIBRARY   |File Code list
```

Function:

The $ LIBRARY control card is used to indicate to GELOAD that user libraries are present and will be searched prior to the system subroutine library search.

Operand Field:

The operand field contains the file code (s) of the user library (s) in the order in which they are to be searched. These file codes must be defined by means of $ File Cards appearing after the $ EXECUTE card. (The File Control Cards are discussed in the preceding section, "File Cards".)

Example:

```
 1        8          16
| $       |LIBRARY   | A1, A2
                        |   |
The user library A1 will|   |
be searched ------------+   |
                            |
Next, the user library A2   |
will be searched -----------+
```

Both of these searches will be made prior to the system subroutine library search.

Rules:

1. The File Code must not exceed two alphanumeric characters. Each File Code is separated by a comma.

2. The Codes must be specified in the order that the user library files are to be searched.

3. A $ TAPE, $ TAPE9, or $ PERM control card referencing the same file code must be included with the program and the file must be labeled.

4. The GELOAD will release the tape back to GECOS upon encountering an end-of-file.

5. The number of file codes on a $ LIBRARY control card is limited to 10. These files must be labeled.

$ LINK

1	8	16
$	LINK	Name, Origin, Option

Function:

The $ LINK control card is used to specify the beginning of an overlay or link and the origin of the overlay.

Operand Field:

The Name which appears in the first variable field identifies the overlay. It must not exceed six alphanumeric characters.

The Origin of the overlay is specified in the second field and also must not exceed six alphanumeric characters. This identfying SYMDEF must have been defined in the first field of a previous $ LINK control card.

The only Option available with the $ LINK card is NOPAC. The NOPAC option is used only when references to SYMDEF overlays (field 2) are not to be purged from the Load Table.

Examples:

Example 1

1	8	16
$	LINK	LINKA

The subprograms following this card and preceding another $ LINK or $ EXECUTE card will be loaded and contained in an area defined as LINKA. The loading addresses for these subroutines would be the next available sequential addresses as in the loading of any set of subroutines.

Example 2

1	8	16
$	LINK	LINKB, LINKA

The subprogram following this card and preceding another $ LINK or a $ EXECUTE card will be loaded and contained in an area defined as LINKB. The loading addresses for these subroutines would begin relative to the address assigned to LINKA at the time of its loading. LINKA must have been defined on a previous $ LINK card. The subprograms contained in LINKA are written in System Loadable Format onto an H* file. All SYMDEF's within these subprograms are purged from the GELOAD Load Table; no future references to these symbols will be allowed.

Example 3

1	8	16
$	LINK	LINKB, LINKA,NOPAC

The subprograms following this $ LINK card are handled exactly as in Example 2 except that the SYMDEF's used within LINKA are not purged from the GELOAD Load Table.

Rules:

1. The Name specified in the first field must be a unique alphanumeric identifier not exceeding six characters and must not be a SYMDEF.

2. The Origin specified in the second field must have been previously identified as an overlay (field) on a previous $ LINK card:

```
$ SNUMB
$ IDENT    \
  . . .     }__________Main Program
  . . .    /
$ ENTRY                LOCI
$ LINK     \           LINKA
  . . .     }----------Link A
  . . .    /
$ LINK     /           LINKB,LINKA
  . . .     }----------Link B
  . . .    \
$ ENTRY                LOC 2
$ LINK     \           LINKC
  . . .     }----------Link C
  . . .    /
$ LINK                 LINKD
  . . .     }----------Link D
  . . .
$ LINK                 LINKE,LINKC
  . . .     }----------Link E
  . . .
$ ENTRY                LOC 3
$ EXECUTE
$ LIMITS   \
  . . .     }----------Data - I* file
  . . .    /
$ ENDJOB
***EOF
```

3. Unless specified by the option NOPAC in field 3, references to all SYMDEF's of overlay origin are purged from the Load Table.

4. In the example with rule 2 above, the user should be aware that, when LINKE is origined at LINKC, both LINKC and LINKD are written to the H* file and their symbols purged from the GELOAD Load Table.

$ LOWLOAD

1	8	16
$	LOWLOAD	Integer (or Blank)

Function:

The $ LOWLOAD control card will initiate loading just above the slave program prefix (64 words) and the control words (2 words) used if a link job is being loaded. (Since, at this point in the loading procedure, GELOAD does not know if a link job is being loaded, allowance must be made for the two control words.)

Operand Field:

The operand field can be either Blank or can contain an Integer. If Blank, loading will start above the fault cells and the control words. If the operand field contains an Integer, the loading address will be incremented by that amount specified by the Integer. Blank Common will start at 100_8, and the two control words will follow the locations reserved for Blank Common.

Example:

1	8	16
$	LOWLOAD	1000

Initial loading address
will be incremented by 1000

Rules:

1. The $ LOWLOAD card must precede all subprograms and control cards used by GELOAD. For example, it must appear before any $ DUMP card and its associated DEBUG statements, subprograms, or $ USE control cards.

$ NOLIB

1	8	16
$	NOLIB	File Code or Blank

Function:

The $ NOLIB control card is used to prevent making a library search for the current program or link being loaded (see also accompanying $ LINK control card description). If operand field is blank, no search is made. If operand field contains a file code(s), only those libraries (files) will not be searched.

Operand Field:

Not used.

Example:

1	8	16
$	NOLIB	

Rules:

1. A separate $ NOLIB card must be used in each link overlay where no library search is desired.

2. With the exception of L*, all file codes specified on the $ NOLIB card must have been previously specified on a $ LIBRARY card.

$ OPTION

1	8	16
$	OPTION	Options

Function:

The $ OPTION control card is used to alter the standard loader options during loading.

Operand Field:

The operand field may contain one or more of the following Options. In the discussion below, the standard options are underlined; if no option is specified, those underlined options will be used.

MAP - A memory map will be produced.

NOMAP - No memory map will be produced.

CONGO - The job is executed regardless of any nonfatal errors detected during loading.

GO - The job is executed only if no errors, fatal and nonfatal,are detected during loading.

NOGO - Loading will proceed to completion, but no execution of the loaded program will take place. Execution pointers and indicators are set for wrapup. No dump is taken.

SET/n/ - Allocated memory is set to the octal pattern specified in n (memory is normally set to zero). The number n can be any octal pattern up to 12 octal characters (n will be right-justified with leading zeros).

ERCNT/n/ - The number of fatal and nonfatal error messages, which are printed, is limited to a total number (n) before loading is aborted. The limit is normally set at 150.

SYMREF - SYMREF's used in each subprogram loaded are listed in the memory map. This option may be set or reset at any time during loading.

NOSREF - No SYMREF's are printed.

LOCOMN - All Labeled Common is assigned below BLANK COMMON. This option must be set before any Blank Common is assigned. The origin of Blank Common is readjusted upward. All Labeled Common is then assumed part of Blank Common and subject to rules governing Blank Common.

NOSETU - The Setup Subroutine (. SETU.) is bypassed and entry is made directly into the user's program. Normal entry to the user program is through . SETU., which does certain initialization functions. . SETU. sets from 40_8 to lowest relocatable address of allocated memory to the octal pattern specified by SET/n/, normally zero.

FCB - Switches are set to call Phase 3 of GELOAD (GE-625/635 General Loader, CPB-1008), which is used to generate File Control Blocks for the job as determined by file control cards. (See preceding section).

NOFCB - Phase 3 of GELOAD is not called.

ALGOL - This option sets all options required for the loading of an ALGOL program (i.e., FCB, LOWLOAD), and .SETU. is replaced by .ASETU, the ALGOL setup routine). This is required when loading programs generated by the ALGOL Compiler.

COBOL - This option sets all options required for loading a COBOL program.

FORTRAN - This option sets the FCB option along with the standard (underlined) options. This is required when loading programs generated by the FORTRAN Compiler.

JOVIAL - This option sets the FCB option along with the standard (underlined) options.

SAVE/Name - This option saves a nonlink program on an H* File with a unique 6-character (maximum) Name such that it may be called and executed at a later date. The program may be retrieved by using the Name in the MME GERSTR calling sequence. If the FCB is implied at the time the program is saved, a unique name is given to the FCB record on the H* File by shifting the Name one character to the right and inserting an asterisk as the first character followed by the specified Name. The FCB is retrieved by using, in the MME GERSTR calling sequence, the Name of the saved program preceded by the asterisk.

NOMSUB - This option eliminates printout of the message: "NONFATAL ERROR MISSING ROUTINE xxxxxx." In large link jobs, this message might appear many times with each link, thus raising the error count and increasing the size of the load map. The message "NONFATAL ERROR MISSING ROUTINE MME GEBORT INSERTED AT REFERENCES" is still being printed as under normal conditions.

Examples:

```
1      8       16
$      OPTION  FORTRAN, SYMREF, SET/400000000000/
```

```
1      8       16
$      OPTION  SAVE/SAVTES, FORTRAN
```

The example above would save two text records on an H* File. The first would be retrieved by the name SAVTES and the FCB's would be retrieved by *SAVTE if the FCB option was in effect when the program was saved.

Rules:

1. The options must be separated with commas.

2. The ALGOL, COBOL, JOVIAL and FORTRAN options must be specified for loading and executing ALGOL, COBOL, JOVIAL and FORTRAN object decks, respectively.

3. The $ OPTION card must precede (in the deck) the object deck for which the option is to take place.

$ RELCOM

1	8	16
$	RELCOM	Integer

Function:

The $ RELCOM card is used to increment the Blank Common loading counter by the amount specified in the variable field. The Blank Common loading counter and its reference symbol .CMN. are normally set just above the fault cells (cell 100_8).

Operand Field:

The operand field contains a decimal Integer equal to the amount which the Blank Common loading counter is to be incremented.

Example:

1	8	16
$	RELCOM	1000

Rules:

1. The $ RELCOM card must precede any object decks and compiler/assembler activities (subprograms).

$ SOURCE

```
 1      8       16
|$      |SOURCE  |
```

Function:

The $ SOURCE control card is created by the system on the R* file when a $ System Call Card is encountered. It is used by the system to instruct GELOAD that a subprogram has been generated on the B* file and that GELOAD must load this subprogram before continuing loading from the R* file.

Operand Field:

Not used.

Example:

```
 1      8       16
|$      |SOURCE  |
```

Rule:

1. Placement of the $ SOURCE card need not concern the user; GECOS creates this card.

$ USE

```
1      8         16
$      USE       Names and/or Size
```

Function:

The $ USE control card instructs GELOAD to enter the Names specified in the operand field into its Load Table as SYMREF's or as Labeled Common regions.

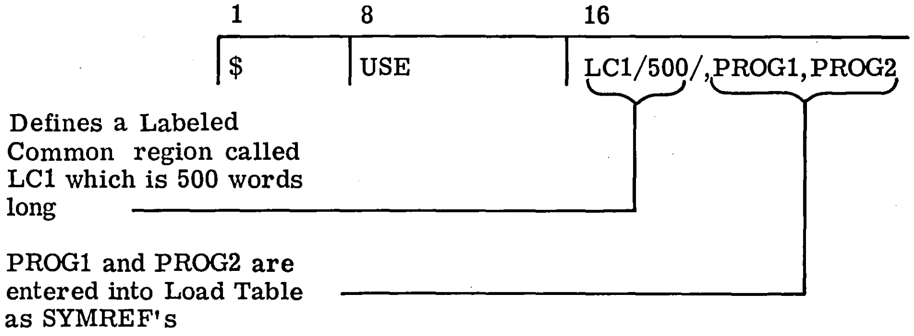
Operand Field:

The operand field can contain either a single name or a name followed by a numeric size.

A single Name is used to place a SYMREF in the Load Table.

A Name followed by a Numeric Size is used to define a Labeled Common region. By placing the character L (see Example 2 below) at the end of the Size field, the Labeled Common region being defined will be handled as if under the LOCOMN option on the $ OPTION card (see accompanying description of the $ OPTION card). All other Labeled Common regions are handled normally.

Example 1

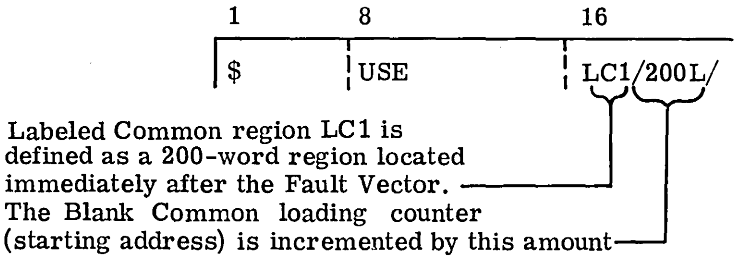


```
1      8         16
$      USE       LC1/500/,PROG1,PROG2
```

Defines a Labeled Common region called LC1 which is 500 words long

PROG1 and PROG2 are entered into Load Table as SYMREF's

Example 2



```
1      8         16
$      USE       LC1/200L/
```

Labeled Common region LC1 is defined as a 200-word region located immediately after the Fault Vector.

The Blank Common loading counter (starting address) is incremented by this amount

Rules:

1. All fields are separated by commas.
2. Names must conform to all rules governing symbols (see GMAP Implementation SSI, CPB-1078).

UTILITY CARDS

The following control cards, used for utility functions, are described in the following pages:

- $ ABORT
- $ ETC
- $ EXTEND
- $ FUTIL
- $ INCODE
- $ UTILITY

$ ABORT

```
 1        8         16
|$        | ABORT   | Not Used
|         |         |
```

Function:

The $ ABORT control card allows the user to perform file operations (such as file dumps, file copy, etc.) as a subactivity by means of the UTILITY routine (see CPB-1422). The $ ABORT control card defines the subactivity initiated by an abnormal termination. Upon encountering the $ ABORT card, GEIN sets the Abort bit (Bit 12) in the Program Switch Word to 1 (On), as described in Chapter 4.

Operand Field:

Not used.

Example:

The following is an example of a deck setup utilizing the $ ABORT card:

```
$ SNUMB
$ IDENT
$ OBJECT
  . . .
  . . .          Object Deck
  . . .
$ DKEND
$ EXECUTE
$ TAPE
$ LIMITS
$ ABORT  \       Abort subactivity creating a U* file
$ FFILE   >      with the UTILITY routine
$ FUTIL  /
$ ENDJOB
```

In the example above, the U* File (generated by the $ ABORT card) is used as input to the UTILITY routine.

Rules:

1. The $ FFILE, $ FUTIL or $ ETC cards must follow the $ ABORT card so that the utility processing is defined.

2. With the exception of the $ ABORT subactivity, deck setup should correspond to any UTILITY routine deck setup. (See accompanying description of $ UTILITY card and GE-625/635 UTILITY, CPB-1422.)

3. The $ FUTIL card which follows the $ ABORT card must reference files already in the aborting activity's slave PAT table.

$ EXTEND

```
 1      8       16
|$      |EXTEND |Not Used
```

Function:

The $ EXTEND control card is used to enable checkout of octal patching of system programs without reediting the software library program. Upon encountering the $ EXTEND card, IMC sets the Extend bit (Bit 16) in the Program Switch Word to 1 (On), as described in Chapter 4.

Operand Field:

Not used

Example:

```
 1      8        16              25
|$      | IDENT  |BOBCAT-1       |        )
|$      | EXTEDIT|               |        |
|$      | TAPE   | E*,X1S        |        |
|$      | SYSNAME|FORTRAN        |RELOC   |   Creating the E* file.  The E* file
|$      | RELADD |102            |        |   and the control cards used are
|27     | OCTAL  |R32710000      |        }   described in GE-625/635 System
|32     | OCTAL  |R1057002306R   |        |   Editor reference manual,
|12764  | OCTAL  |0,0,0          |        |   CPB-1138
|$      | SYSNAME|.FORT1         |RELOC   |
|$      | RELADD |1024           |        |
|10024  | OCTAL  |26651710000    |        )
|26651  | OCTAL  |15567235000,100756000000,10025710000
|       |        |
|       |        |          )
|$      | FORTRAN|          |
|$      | EXTEND |          }   Using the created E* file to
|$      | TAPE   |E*,X1R    |   extend FORTRAN
|       | . . .  |          |
|       | . . .  |          |
|$      | ENDJOB |          )
```

Rules:

1. The contents of the E* file must have been created in a previous activity (see example above) and assigned to E* by means of a $ TAPE control card.

2. The $ EXTEND card must follow immediately behind the system call card of the software system to be extended or patched.

$ ETC

1	8	16
$	ETC	Continuation of preceding card

Function:

The $ ETC control card is used to continue the variables in the operand field of the preceding card.

Operand Field:

The operand field contains the continuation of the preceding card.

Example:

1	8	16
$	FFILE	FC, P1, P2,

1	8	16
$	ETC	PN

Rules:

1. The $ ETC card must be the next card after the card it is continuing.
2. It is the responsibility of the program processing the $ ETC card to process the variable fields of both cards.
3. The last variable of the card preceding the $ ETC card must be followed by a comma.
4. The $ ETC control card can be used with $ FFILE and $ FUTIL control cards.

$ FUTIL

1	8	16
$	FUTIL	File Code 1, File Code 2, Options

Function:

The $ FUTIL control card (or cards) is used as a directive in the UTILITY activity (see also accompanying $ ABORT and $ UTILITY card descriptions). It is used to direct certain operations so that UTILITY will produce a combination octal/BCD dump, will copy certain files, or will compare certain files. The UTILITY routine is described in GE-625/635 UTILITY, CPB-1422.

Operand Field:

The operand field contains the file code of each of the two files (maximum per $ FUTIL card) being serviced. The File Codes are the two-character alphanumeric codes assigned to the files. This is the same file code assigned by the programmer to uniquely identify a particular file. The UTILITY routine creates a file control block and reserves a buffer for each file code.

The following Options are available with the $ FUTIL card:

COPY/M/	-	Copy M from first file (File Code 1) to second file (File Code 2)
COMP/M/	-	Compare M from first file (File Code 1) to second file (File Code 2)
SKIP/M,N/	-	Skip M on first file (File Code 1) and N of second file (File Code 2)
DUMP/M,N/	-	Dump M on first file (File Code 1) and N of second file (File Code 2)
RWD (or REW)/ AB/	-	Rewind file AB
HOLD/AB/	-	Do not close file AB

Where:

M is any combination of Number of Files (#F) and/or Number of Records (#R) from the first file (specified in File Code 1).

N is any combination of Number of Files (#F) and/or Number of Records (#R) from the second file (specified in File Code 2).

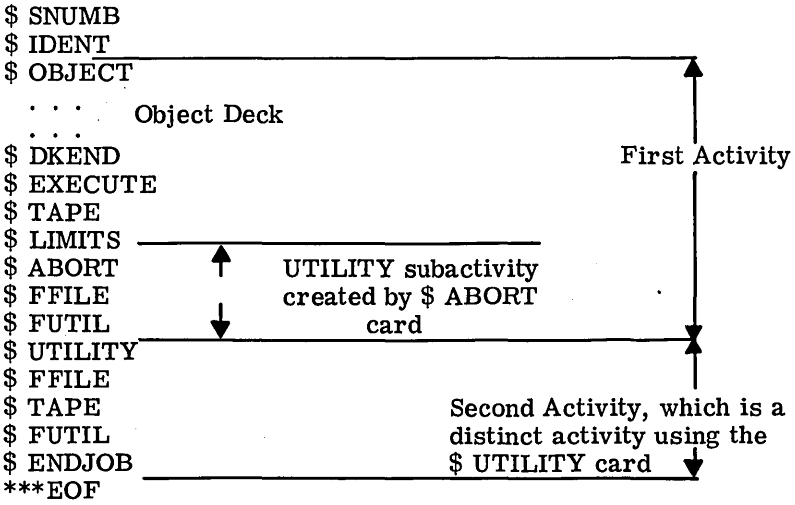
For example, these parameters may be specified as 3F4R, 4F, or 5R.

Examples:

1	8	16
$	FUTIL	AB,CD,RWD/AB/,DUMP/1F/,RWD/CD/,COPY/1F/,RWD/CD/,
$	ETC	DUMP/,1F/

File AB will be rewound and the first File of AB will be dumped. File CD will be rewound and the second (first was dumped) file of AB will be copied as the first file of CD. File CD will then be rewound, and one file will be dumped.

The following is a deck setup utilizing the $ FUTIL directive cards in a UTILITY activity used both as a subactivity (called by $ ABORT card) and as a separate activity (called by the $ UTILITY card).



```
$ SNUMB
$ IDENT
$ OBJECT
  . . .  Object Deck
  . . .
$ DKEND
$ EXECUTE
$ TAPE
$ LIMITS
$ ABORT
$ FFILE
$ FUTIL
$ UTILITY
$ FFILE
$ TAPE
$ FUTIL
$ ENDJOB
***EOF
```

First Activity

UTILITY subactivity created by $ ABORT card

Second Activity, which is a distinct activity using the $ UTILITY card

Rules:

1. The $ FUTIL control card must be located following the $ ABORT and/or $ UTILITY. Multiple $ FUTIL cards are permitted within an activity.

2. One File Code can be a null field if only one file will be serviced (i.e., for rewinding one tape file).

```
1         8         16
$         FUTIL     ,AB,RWD/AB/,DUMP/,1F/
$         FUTIL     AB,,RWD/AB/,DUMP/1F/
```

3. The File Codes consist of any 2-character alphanumeric designation with the exception of an oblique (/) and a comma (,), which can not be used since these symbols are used as delimiters.

4. Variable field can not exceed column 72; the program ignores columns 73-80

5. The $ ETC card may be used to continue the variable field.

6. If an end-of-file is encountered during a record countdown, the record count is considered exhausted.

7. All file marks except octal 75 and 76 are considered file delimiters by UTILITY.

$ INCODE

```
 1        8        16
|$       | INCODE |BCD Card Code
```

Function:

The $ INCODE control card is used to indicate that the following BCD input cards must undergo character transliteration from the character set specified on the $ INCODE card to the internal binary representation of the GE-635 Standard Character Set. All BCD cards following the $ INCODE card will undergo transliteration up to the next $ control card.

Operand Field:

The operand field contains a mnemonic identifying the character set used in punching the input cards. The following are the mnemonics and the character sets they identify:

IBMF	signifies	IBM FORTRAN set
IBMC	signifies	IBM COBOL set
GE225	signifies	GE-225 set
IBMEL	signifies	IBM Extended Language code (029 keypunch codes)

These character sets are defined in Appendix F.

Example:

Deck Setup:

```
$ COBOL
$ INCODE        IBMC
  . . . \
  . . .  }      Data
  . . . /
$ ENDJOB
```

```
 1        8        16
|$       | INCODE |IBMC
```

The cards following the $ INCODE card were punched in the IBM COBOL character set

Rules:

1. The $ INCODE must follow the $ control card denoting the compiler language (that is, $ FORTRAN) and precede the data to be transliterated.

2. To transliterate card images which are on a PERM file and called by a $ SELECT card, the $ INCODE should precede the card images on the PERM file.

$ UTILITY

1	8	16
$	UTILITY	Dump option

Function:

The $ UTILITY control card allows the user to perform file operations (such as file dumps, file compares, etc.) as a separate activity, by means of the UTILITY routine. This card differs from the $ ABORT card in that it calls the UTILITY routine as a separate activity. When GEIN encounters this card, it opens a U* file and all following cards, up to the next $ control card indicating a new activity, are placed on that file. The U* file then becomes the input file for UTILITY. The UTILITY routine is described in GE-625/635 UTILITY, CPB-1422.

Operand Field:

The operand field may contain one of the following Dump options. If left blank, the standard (underlined) option is assumed.

DUMP - Slave core dump will be given if activity terminates abnormally.

NDUMP - Only program registers will be dumped if activity terminates abnormally.

Example:

The following is an example of a deck setup with the separate UTILITY activity.

```
$ SNUMB
$ IDENT
  . . . )
  . . . }  OBJECT DECK
  . . . )
$ EXECUTE
$ DATA                        AC
$ INCODE                      IBMF
  . . . )
  . . . }  DATA CARDS
  . . . )
$ TAPE                        BD, X1S
$ UTILITY                     DUMP
$ TAPE                        BD, X1R
$ FUTIL                       BD,,RWD/BD/,DUMP/1F/
$ ENDJOB
```

Rule:

1. The $ FFILE and/or $ FUTIL cards must not precede the $ UTILITY card. The UTILITY routine will allow only 40 $ FFILE cards per activity; there is no limit on the number of $ FUTIL cards.

SYSTEM CALL CARDS

The System Call control cards are used by Input Media Conversion to call the various language processors and to specify the various options available when using those processors.

- $ ALGOL
- $ COBOL
- $ FILSYS
- $ FORTRAN
- $ GMAP
- $ IDS
- $ JOVIAL
- $ PRODUCT
- $ PROGRAM
- $ 44SIM
- $ 94SIM
- $ 1401SIM
- $ 225SIM

In addition, the following control cards, which are used with the various language processors, are described.

- $ UPDATE
- $ ALTER

Certain general rules apply to the use of System Call Cards:

1. Options can be listed in any order in the operand field.
2. If no options are specified, GECOS will assume the standard options (underlined).
3. A null field will terminate the card scan.

$ ALGOL

1	8	16
$	ALGOL	Options

Function:

The $ ALGOL control card is used to call the ALGOL compiler.

Operand Field:

The operand field specifies the system options. The following options are available with ALGOL (standard options are underlined):

LSTIN - A listing of source input will be prepared by the ALGOL Compiler

NLSTIN - No listing of source input will be prepared

LSTOU - A listing of assembled object program output will be prepared

NLSTOU - No listing of assembled object program output will be prepared

DECK - A binary object program deck will be prepared as output by ALGOL

NDECK - No binary object program deck will be prepared as output

COMDK - A compressed deck of the source program will be prepared by ALGOL

NCOMDK - No compressed deck of the source program will be prepared

DUMP - Slave core dump will be given if activity terminates abnormally

NDUMP - Only program registers will be dumped if activity terminates abnormally

DEBUG - A listing of the intermediate files of the ALGOL Compiler will be produced

NDEBUG - No listing of the intermediate files of the ALGOL Compiler will be produced

Example:

1	8	16
$	ALGOL	LSTIN,DECK,COMDK

Rules:

1. The $ ALGOL control card must precede the source cards of each program or subprogram to be processed and must precede any other control card associated with that activity.

2. The options can be listed in any order in the operand field.

3. If no options are specified in the operand field, GECOS will use the standard options (underlined).

$ COBOL

1	8	16
$	COBOL	Options

Functions:

The $ COBOL control card is used to call the COBOL compiler.

Operand Field:

The operand field specifies the system options. The following options are available with COBOL (standard options are underlined):

LSTIN - A listing of source input will be prepared by the COBOL compiler

NLSTIN - No listing of source input will be prepared

LSTOU - A listing of the assembled object program output will be prepared by GMAP

NLSTOU - No listing of the assembled object program output will be prepared

NDECK - No binary object program deck will be prepared

DECK - A binary object program deck will be prepared as output by GMAP

COMDK - A compressed deck of the source program will be prepared

NCOMDK - No compressed deck of the source program will be prepared

DUMP - Slave core dump will be given if activity terminates abnormally

NDUMP - Only program registers will be dumped if activity terminates abnormally

COPY - A copy prepass is required (See Rule 4.)

NCOPY - No copy prepass is required

ON6 - COBOL will generate a REF ON so that GMAP will build a Symbol Reference Table

SYMTAB - GMAP will prepare a listing of the Symbol Reference Table (if one has been built) even though NLSTOU is also specified.

Example:

```
1      8       16
$      COBOL   DUMP
```

Rules:

1. The $ COBOL Control card must precede the source cards of each program or subprogram to be processed and must precede any other control card associated with that activity.

2. The options can be listed in any order in the operand field.

3. If no options are specified in the operand field, GECOS will use the standard options (underlined).

4. All source decks which use the COPY clause or the RENAMING file option (see GE-625/635 COBOL, CPB-1007) must use the COPY option on the $ COBOL card.

$ FILSYS

1	8	16
$	FILSYS	Dump option

Function:

The $ FILSYS control card is used to call the GECOS-III FILSYS activity, described in GE-625/635 GECOS File System (FILSYS), CPB-1513.

Operand Field:

The operand field specifies the DUMP option if that option is desired. If the field is blank, NDUMP is implied.

DUMP -	Slave core dump will be produced if the activity terminates abnormally
NDUMP -	Only a panel dump will be produced if the activity terminates abnormally

Example:

1	8	16
$	FILSYS	DUMP

Rules:

1. The $ FILSYS control card must precede the source cards of each program or subprogram to be processed and must precede any other control card associated with that activity.
2. If no option is specified in the operand field, GECOS will use the standard options (underlined).

$ FORTRAN

1	8	16
$	FORTRAN	Options

Functions:

The $ FORTRAN control card is used to call the FORTRAN compiler.

Operand Field:

The operand field specifies the system options. The following options are available with FORTRAN (standard options are underlined):

LSTIN - A listing of source input will be prepared by the FORTRAN compiler

NLSTIN - Inoperable

LSTOU - A listing of the assembled object program output will be prepared by GMAP

NLSTOU - No listing of the assembled object program output will be prepared

DECK - A binary object program deck will be prepared as output

NDECK - No binary object program deck will be prepared

COMDK - A compressed deck of the source program will be prepared

NCOMDK - No compressed deck of the source program will be prepared

STAB - A table of all symbols used in the source program (statement numbers and variable names will be generated with appropriate control information for use by GELOAD in decoding DEBUG statements)

NSTAB - No symbol table will be produced

DUMP - Slave core dump will be given if activity terminates abnormally

NDUMP - Only program registers will be dumped if activity terminates abnormally

SYMTAB - GMAP will prepare a listing of the Symbol Reference Table (if one has been built) even though NLSTOU is also specified.

Example:

```
1         8           16
$         FORTRAN     LSTIN,NLSTOU,STAB
```

Rules:

1. The $ FORTRAN control card must precede the source cards of each program or subprogram to be processed and must precede any other control card associated with that activity.
2. The options can be listed in any order in the operand field.
3. If no options are specified in the operand field, GECOS will use the standard options (underlined).

$ GMAP

1	8	16
$	GMAP	Options

Function:

The $ GMAP control card is used to call the Macro Assembler.

Operand Field:

The operand field specifies the system options. The following options are available with GMAP (standard options are underlined):

LSTOU - A listing of the assembled object program output will be prepared

NLSTOU - No listing of the assembled object program output will be prepared

DECK - A binary object program deck will be prepared as output

NDECK - No binary object program deck will be prepared

COMDK - A compressed deck of the source program will be prepared

NCOMDK - No compressed source deck will be prepared

GMAC - The GMAP System Macros are required for assembly

NGMAC - Used only when assembling GECOS modules to specify no GMAP macros (See GE-625/635 Programming Reference Manual, CPB-1004.)

ON5 - Print all source images regardless of any pseudo-ops that might otherwise result in their not being printed.

DUMP - Slave core dump will be produced if activity terminates abnormally

NDUMP - Only a panel dump of program registers will be produced if activity terminates abnormally

NXEC - Execution activity should be deleted if fatal errors are encountered during assembly

SYMTAB - A listing of Symbol Reference Table (if one has been built) will be prepared even though NLSTOU is also specified

Example:

```
1      8       16
$      GMAP    NLSTOU,DECK
```

Rules:

1. The $ GMAP control card must precede the source cards of each program or subprogram to be processed and must precede any other control card associated with that activity.

2. The options can be listed in any order in the operand field.

3. If no options are specified in the operand field, GECOS will use the standard options (underlined).

$ IDS

```
1      8       16
$      IDS     Options
```

Function:

The $ IDS control card is used to call the IDS Translator.

Operand Field:

The operand field specifies the system options. The following options are available with IDS/COBOL (standard options are underlined):

LSTIN - An IDS input listing and COBOL input listing will be prepared

NLSTIN - No IDS listing of input will be prepared. Option is reset to LSTIN prior to calling COBOL.

LSTOU - A listing of the assembled object program output will be prepared by GMAP

NLSTOU - No listing of output will be prepared

NDECK - No binary object program deck will be prepared

DECK - A binary object program deck will be prepared as output by GMAP

COMDK - A compressed deck of the source program will be prepared during translation

NCOMDK - No compressed deck of the source program will be prepared

ON6 - COBOL will generate a REF ON so that GMAP will build a Symbol Reference Table.

DUMP - Slave core dump will be produced if activity terminates abnormally

NDUMP - Only a panel dump of program registers will be produced if activity terminates abnormally

COPY - A copy prepass is required (See rule 4.)

NCOPY - No copy prepass is required

SYMTAB - GMAP will prepare a listing of the Symbol Reference Table (if one has been built) even though NLSTOU is also specified.

Example:

```
1      8      16
$      IDS    LSTIN,COMDK
```

Rules:

1. The $ IDS control card must precede the source cards of each program or subprogram to be processed and must precede any other control card associated with that activity.

2. The options can be listed in any order in the operand field.

3. If no options are specified in the operand field, GECOS will use the standard options (underlined).

4. All source decks which use the COPY clause or the RENAMING file option (see GE-625/635 COBOL, CPB-1007) must use the COPY option.

$ JOVIAL

1	8	16
$	JOVIAL	Options

Function:

The $ JOVIAL is used to call the JOVIAL compiler.

Operand Field:

The operand field specifies the system options. The following options are available with JOVIAL (standard options are underlined).

LSTIN - A listing of source input will be prepared by the JOVIAL compiler

NLSTIN - No listing of source input will be prepared

LSTOU - A listing of the assembled object program output will be prepared

NLSTOU - No listing of the assembled object program output will be prepared

DECK - A binary object program deck will be prepared by the JOVIAL compiler

NDECK - No binary object program deck will be prepared

COMDK - A compressed deck of the source program will be prepared

NCOMDK - No compressed deck of the source program will be prepared

NDEBUG - No object code debug will be produced

DEBUG - An object code debug will be produced

DUMP - Slave core dump will be produced if activity terminates abnormally

NDUMP - Only a panel dump of program registers will be produced if activity terminates abnormally

Example:

1	8	16
$	JOVIAL	LSTIN, COMDK, DUMP

Rules:

1. The $ JOVIAL control card must precede the source cards of each program or subprogram to be processed and must precede any other control card associated with that activity.
2. The options can be listed in any order.
3. If no options are specified, GECOS will use the standard options (underlined).

$ PRODUCT

1	8	16
$	PRODUCT	Name, FCB Option, Assembler/Compiler Options

Function:

The $ PRODUCT control card is used to retrieve and initiate execution of programs from a Production Library. (See Appendix D for a description of the generation and use of a Production Library.)

Operand Field:

The operand field contains the Name of the program being called from the Production Library and also may contain a FCB Option and/or Assembler/Compiler Options.

The Name is a 6-character (or less) Name of the program being called.

The FCB Option must be specified in the second subfield if the program designated on the $ PRODUCT card has a GELOAD-generated file control block and is a linked job.

The Assembler/Compiler Options (LSTIN, LSTOU, DECK, DUMP, etc.) are available for use with the $ PRODUCT card.

Examples:

Control card without FCB Option

1	8	16
$	PRODUCT	PAY1

Control card with FCB Option

1	8	16
$	PRODUCT	SAV1,FCB

Rules:

1. The program named in the first variable field has been edited into the Production Library. (See Appendix D.) The name must not exceed 6 characters and must agree with the name specified in the $ SYSLD card (see GE-625/635 System Editor, CPB-1138) when the library was generated.
2. No imbedded blanks are allowed.
3. No more than four Assembler/Compiler options can be used if the FCB Option is used. Otherwise five options can be used.

$ PROGRAM

```
1      8          16
$      PROGRAM    Name, Options
```

Function:

The $ PROGRAM control card is used to call a program, previously edited onto the Software Library (such as GMAP, FORTRAN, and GELOAD) and for which no call capability exists. Upon encountering a $ PROGRAM card, GECOS enters the Program Name in Word 3 of the GECOS control stack, and an activity is defined. Standard GECOS limits (24k) are assumed. File control cards following the $ PROGRAM control card will be put on R*. Output files P* and T* are also included in the slave PAT Table.

Operand Field:

The operand field contains the Name of the program being called and may also specify all Options currently available on EXECUTE, GMAP, and Compiler Call cards (see preceding sections).

The Name of the Program being called must not exceed 6 characters.

The Options can be any options currently available with EXECUTE, FORTRAN, COBOL, ALGOL, and GMAP. No more than five options will be recognized. These options set respective bits of the GECOS Switch Word (see Chapter 4).

Example:

```
1      8          16
$      PROGRAM    TEST,ON1
```

Rules:

1. The program named in the first variable field must have been edited into the Software Library. The Name must not exceed 6 alphanumeric characters.
2. A maximum of five options, following the Name, will be recognized.
3. Any data files following the $ PROGRAM card must be preceded by a $ DATA card specifying a file code (see preceding section entitled "$ File Cards"). No data files are assumed.

$ 44SIM

1	8	16
$	44SIM	Dump Option

Function:

The $ 44SIM control card is used to call the GE-625/635 IBM 7044 Simulator and causes the 44SIM program to be loaded into memory from the system file.

Operand Field:

The following options may be used. Unless otherwise specified NDUMP is assumed.

DUMP - Slave core dump will be given if activity terminates abnormally

NDUMP - Only program registers will be dumped if activity terminates abnormally

System options are defined by the $ 44OPTN card (see GE-625/635 IBM 7040/7044 Simulator, XCPB-1188.

Example:

1	8	16
$	44SIM	DUMP

Rules:

1. The $ 44SIM control card must precede the source cards of each program or subprogram to be processed and must precede any other control card associated with that activity.

2. If loading the simulation activity from an input device rather than one system file, the $ 9SA card (see GE-625/635 IBM 7040/7044 Simulator, CPB-1188) must be used following the $ EXECUTE card.

$ 94SIM

1	8	16
$	94SIM	Dump Option

Function:

The $ 94SIM control card is used to call the GE-625/635 IBM 7094 Simulator.

Operand Field:

System options are defined by the $ 94OPTN card (see GE-625/635 IBM 7094 Simulator, CPB-1081).

The following Dump Options are available. Unless otherwise specified, NDUMP is assumed.

DUMP - Slave core dump will be given if activity terminates abnormally

NDUMP - Only program registers will be dumped if activity terminates abnormally

Example:

1	8	16
$	94SIM	DUMP

Rules:

1. The $ 94SIM control card must precede the source cards of each program or subprogram to be processed and must precede any other control card associated with that activity.

2. If loading the simulation activity from an input device rather than the system file, the $ 9SA control card (see GE-625/635 IBM 7094 Simulator, CPB-1081) must be used following the $ EXECUTE card.

$ 1401SIM

1	8	16
$	1401SIM	Dump Option

Function:

The $ 1401SIM control card is used to call the IBM 1401 Simulator.

Operand Field:

The following Dump Options are available. Unless otherwise specified, NDUMP is assumed.

DUMP - Slave core dump will be given if activity terminates abnormally

NDUMP - Only program registers will be dumped if activity terminates abnormally

Example:

1	8	16
$	1401SIM	DUMP

Rule:

The $ 1401SIM control card must precede the source cards of each program or subprogram to be processed and must precede any other control card associated with that activity.

$ 225SIM

1	8	16
$	225SIM	Dump Option

Function:

The $ 225SIM control card is used to call the GE-225 Simulator (described in GE-625/635 GE-225 Simulator, CPB-1186).

Operand Field:

The following Dump Options are available. Unless otherwise specified, NDUMP is assumed.

DUMP	- Slave core dump will be given if activity terminates abnormally
NDUMP	- Only program registers will be dumped if activity terminates abnormally.

Example:

1	8	16
$	225SIM	DUMP

Rule:

1. The $ 225SIM control card must precede the source cards of each program or subprogram to be processed and must precede any other control card associated with that activity.

$ UPDATE

1	8	16	73	80 ~~80~~
$	UPDATE	Operand	Identification (Optional)	

Function:

The $ UPDATE control card is used when supplying alter input to a compiler or assembler.

Operand Field:

The operand field is only used when a listing of the $ ALTER cards is desired. In this case, the option LIST is entered beginning in column 16. Any $ ALTER cards out of order (see $ ALTER description above) will be listed as ALTER file errors.

The Identification field is optional and can be used to place an identifer in columns 73-80 of each source card contained in the Alter file (A*).

Example:

1	8	16
$	UPDATE	LIST

During GMAP input processing, GEFRC will write alter input data from the A* file on SYSOUT, giving the user a listing of his alter input data.

Rules:

1. The $ UPDATE card always precedes all $ ALTER cards for the activity being altered.
2. Input to an alter file is terminated when another control card is encountered ($ INCODE is an exception).
3. Any previous information written in columns 73-80 on the source cards will be overwritten if the same columns of the $ UPDATE card contain an entry (any nonzero or nonblank information).
4. The $ UPDATE control card must follow the source or compressed deck to be altered.

$ ALTER

1	8	16
$	ALTER	M,N

Function:

The $ ALTER control card is used to make corrections to the source card or COMDK input to the various language processors. Source cards can be inserted or deleted, or can replace other source cards from the input file to the assembler or compilers by specifying their respective alter numbers in the operand field.

Operand Field:

The operand field contains alter numbers M, N taken from a previous assembly listing of the job to which changes are to be made. The alter numbers are consecutive card numbers starting with 00001 and increasing by one for each source input card.

When it is desired to insert cards into a deck, the N subfield is not used. In this case, the source cards following this $ ALTER up to but not including the next $ ALTER card or $ control card will be inserted just prior to the card corresponding to alter number M.

When it is desired to delete and/or replace one or more cards from a deck, the N subfield is given as shown above. When M and N are equal, card M will be deleted. When N identifies a card following M, all cards, M through N, will be deleted. In addition, any cards following the $ ALTER card up to but not including the next $ ALTER card (or other control card) will be inserted in place of the deleted cards.

Example:

1	8	16
$	ALTER	100,102

Signifies deletion of source cards 100, 101, and 102.

Rules:

1. All cards to be put on the alter file (A*) must be preceded by the $ UPDATE control card.
2. ALTER numbers specified on a $ ALTER card must be in ascending order (i.e., M ≤ N).
3. The ALTER file must be prepared in ascending ALTER number order. The $ ALTER cards not following this convention will be printed on the execution report as ALTER file errors.

CONDITIONAL CARDS

The following control cards are used to conditionally execute activities or skip certain activities depending on the results of previous activities of that job:

- $ GOTO
- $ IF

Another conditional card, the

- $ WHEN

is used to permit conditional controls within an activity.

The above cards are used in conjunction with

- $ Label

card which defines the Label given in the operand field of the $ GOTO, $ IF, or $ WHEN cards.

$ GOTO

```
1              8               16
$              GOTO            Label
```

Function:

The $ GOTO card causes an unconditional transfer of control to the portion of the program specified by the label name or to end of job.

Operand Field:

The operand field contains a Label which may not exceed six characters

Example:

```
1              8               16
$              GOTO            THERE
```

Control will skip to that portion labeled THERE. If the label THERE is not found, an end of job will occur.

```
1              8               16
$              GOTO            ENDJOB
```

An immediate end of job will occur.

Rules:

1. A GOTO-referenced control label must be located such that R* File carryover will not cause GEFRC block serial number sequence errors while R* is being read.

2. The named control label (see $ Label in this section) should not precede the $ GOTO card referencing it.

Can the label be BREAK (for the $ BREAK card) or the label ENDJOB is used (for the $ ENDJOB card)?

$ IF Conditional execution/omission of activities

```
1      8      16
$      IF     Operator, Label
```

Function:

The $ IF card permits conditional branching to a following control label or to end of job, based on the abort of the immediately preceding activity or on the setting of bits in the Program Switch Word. (See page 19

tests bit 12 for $ABORT
or the ABORT flag in
the SSA .STATE
in Slave Prefix

Operand Field:

The operand field contains an Operator field which may be either ABORT or a Boolean expression indicating bit status (+ meaning OR, * meaning AND, / meaning NOT).

The Label may not exceed six characters.

Examples:

```
1      8      16
$      IF     18, THERE
```

If bit 18 is On,
transfer to THERE.

```
1      8      16
$      IF     ABORT, ENDJOB
```

If last activity aborted,
transfer to end of job.

```
1      8      16
$      IF     19+31, HERE
```

If bits 19 or 31 are On,
transfer to HERE.

```
1      8      16
$      IF     /21+25*/26, IN
```

If not bit 21 or bit 25
and not bit 26,
transfer to IN.

hierarchy
NOT
AND
OR

NOT 21
NOT 26
.NOT.21.OR.25.AND..NOT.26

Rules:

1. Evaluation is from left-to-right and if any OR condition is true, the branch is taken.

2. The $ IF card may not be continued by a $ ETC card. However, successive $ IF cards may be used.

3. The $ IF card acts as an activity delimiter. It must follow all control cards, except other $ IF cards, relating to an activity.

4. A IF-referenced control label must be located such that R* File carryover will not cause GEFRC block serial number sequence errors while R* is being read.

5. The named control label (see $ Label in this section) should not precede the $ GOTO card referencing it.

Conditional inclusion/exclusion of control cards within an activity.

$ WHEN

```
1      8        16
$      |WHEN    |Operator, Label
```

Function:

The $ WHEN card permits conditional branching to a following control label or to end of job. It may be used to permit conditional $ File cards or other controls within an activity.

Operand Field:

The operand field contains an Operator field which may be either ABORT or a Boolean expression indicating bit status (+ meaning OR, * meaning AND, / meaning NOT).

The Label may not exceed six characters.

Example:

```
1 3        8         16
$ |        |SNUMB    |
$ |        |IDENT    |
$ |        |GMAP     |
$ |        |WHEN     |18,NOPR
$ |        |TAPE     |P*, A1S
$ |NOPR:   |         |OMIT TAPE P* IF NO LISTING
$ |        |GMAP     |
$ |        |WHEN     |18,NOPS
$ |        |TAPE     |P*, A1D
$ |NOPS:   |         |OMIT TAPE P* IF NO LISTING
$ |        |ENDJOB   |
```

The comments appearing on the $ Label cards ($ NOPR and $ NOPS) will appear on the execution report. The first $ WHEN card allows, if bit 18 is On, the skipping of the following $ TAPE card.

$ Label

```
 1  3
|$ | Label:
```

Function:

The $ Label control card defines the entry at which control is transferred from a preceding $ GOTO, $ IF, or $ WHEN control card.

Operand Field:

The Label is a name not exceeding six characters. It may begin in col. 3 - 7 and must terminate with a colon and at least one blank. Comments may follow the blank, if desired. These will be printed on the execution report.

Examples:

```
 1  3
|$ |HERE:bGOTO HERE IF BIT 18 IS ON

 1   4
|$  |WHERE:bGOTO THERE

 1       7
|$      |THERE:
```

Rules:

1. The Label may begin in col. 3 - 7; it must not exceed six characters.
2. At least one blank must follow the colon.

6. SYSTEM FILES

System files are created by GECOS for its own use and/or for use by software processor systems (GMAP, FORTRAN, GELOAD, etc.). System files used as input to other software processors working under GECOS are created by GEIN. System files are sometimes created by GEIN and passed between software processors, as in a FORTRAN compilation. In this case, GEIN creates S*, which is used as input to FORTRAN. FORTRAN creates a G* file, which is then passed to GMAP. The interrelationship of the system files and the various software processors is shown in Figure 2. The system files used with each processor system are listed in Figure 3. System files are designated symbolically by two characters, as are user files. The system files are distinguished from user files, however, in that one of the two characters is an asterisk. For example, A* and *1 refer to system files, whereas AB and C1 can be used to refer to user files. Like user files, the system files are arranged in the standard system format, shown in Figure 4. A detailed description of the standard system format is given in GE-625/635 File and Record Control, CPB-1003.

Included in this section are descriptions of the following system files:

- Alter file (A*)
- Object program file (B*)
- Binary deck file (C*)
- GMAP source file (G*)
- Program link file (H*)
- Data Storage File (I*)
- Compressed deck file (K*)
- System library file (L*)
- System output file (P*)
- Loader input file (R*)
- Compiler source file (S*)
- Utility file (U*)
- IDS source file (*S)
- Intermediate file, FORTRAN and GMAP (*1)
- Intermediate file, ALGOL (*2)
- Intermediate file, IDS and COBOL (*3)

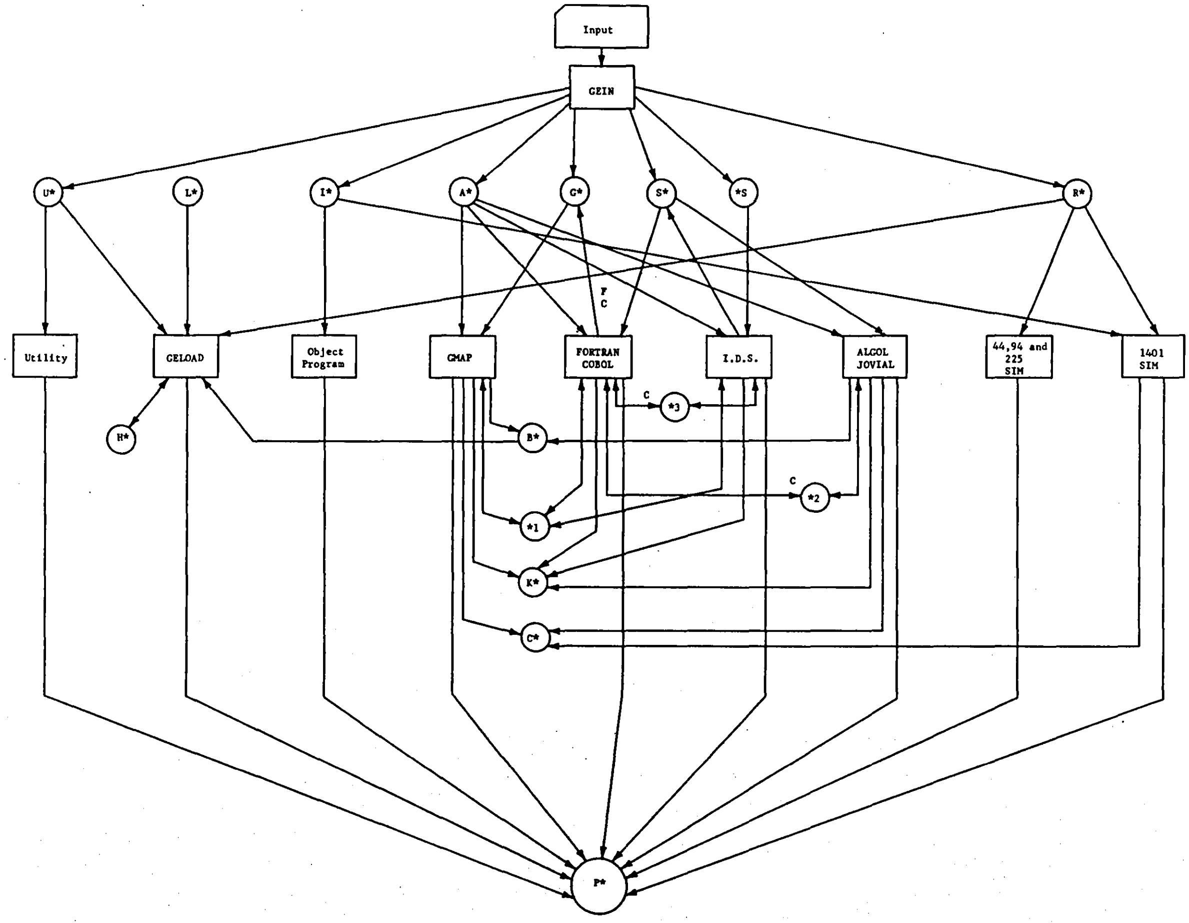


Figure 2. Interrelationship of System Files to Various Software Processors

Software Processor System	Files Used	Normally Used Size[a]	 File Media[b]
ALGOL	A*	~~Variable-~~1L[c]	SS
	B*	2 L[d]	DISC
	C*	-	SYSOUT
	K*	-	SYSOUT
	P*	-	SYSOUT
	S*	6L Variable-L[c]	SS
	*2	12 L[d]	DISC
COBOL	A*	~~Variable-~~1L[c]	SS
	G*	~~2L~~ 10L	SS
	K*	-	SYSOUT
	P*	-	SYSOUT
	S*	6L ~~Variable-L~~[c]	SS
	*1	~~5L[d]~~ variable	SS
	*2	1L	DISC
	*3	~~15R~~ 30R	DRUM
FORTRAN	A*	1L - c	SS
	G*	5L[c] ~~2L[d]~~	SS
	K*	-	SYSOUT
	P*	-	SYSOUT
	S*	6L ~~Variable~~-L[c]	SS
	*1	~~5L[d]~~ variable	SS
GELOAD	B*	~~4L~~ 2L	DISC
	H*	5R	DISC TAPE
	I*	Variable-L[c]	SS
	L*	-	TAPE
	P*	-	SYSOUT
	R*	Variable-L[c]	SS
GMAP	A*	~~Variable-L~~[c] 1L	SS
	B*	2L[d]	DISC
	C*	-	SYSOUT
	G*	Variable-L[c]	SS
	K*	-	SYSOUT
	P*	-	SYSOUT
	*1	~~5L[d]~~ variable	SS
IDS	A*	1L ~~Variable-L~~[c]	SS
	K*	-	SYSOUT
	P*	-	SYSOUT
	S*	~~Variable~~-L[d]	SS
	*S	Variable-L[c]	SS
	*1	Variable-L[d]	SS
	*3	~~15R~~ 30R	DRUM
JOVIAL	A*	1L ~~Variable-L~~[c]	SS
	B*	2L[d]	DISC
	C*	-	SYSOUT
	K*	-	SYSOUT
	P*	-	SYSOUT
	S*	Variable-L[c]	SS
	*1	~~5L[d]~~ variable	SS
	*2	1L[d]	DISC
	*3	~~1R 8K~~ 1K	DISC
Utility	P*	-	SYSOUT
	U*	1L-variable c	TAPE DISC DRUM
44 Simulator (44SIM)	R*	Variable-L[c]	SS
	P*	-	SYSOUT
	S*	Variable-L[c]	SS
94 Simulator (94SIM)	R*	Variable-L[c]	SS
	P*	-	SYSOUT
	S*	Variable-L[c]	SS
225 Simulator (225SIM)	R*	Variable-L[c]	SS
	P*	-	SYSOUT
	I*	Variable-L[c]	SS
1401 Simulator (14SIM)	I*	-	SS
	R*	Variable-L[c]	SS
	P*	-	SYSOUT
	C*	-	SYSOUT

[a] L denotes linked links; R denotes random links. A definition of links is contained in Chapter 12, "Allocation".

[b] SS denotes system storage; drum with Drum-GECOS; disc with Disc-GECOS.

[c] These GEIN-created files will use as many links as are necessary to contain the input data involved. However, the total number of links which may be in use at any point in time is limited by a constant contained in the Allocator (currently 50 links).

[d] These files, which are used as intermediate data files, are initially allocated a minimum number of links, as shown. That number may be increased dynamically via GEMORE during their use by the various language processors.

Figure 3. Files Used by the Various Software System Processors

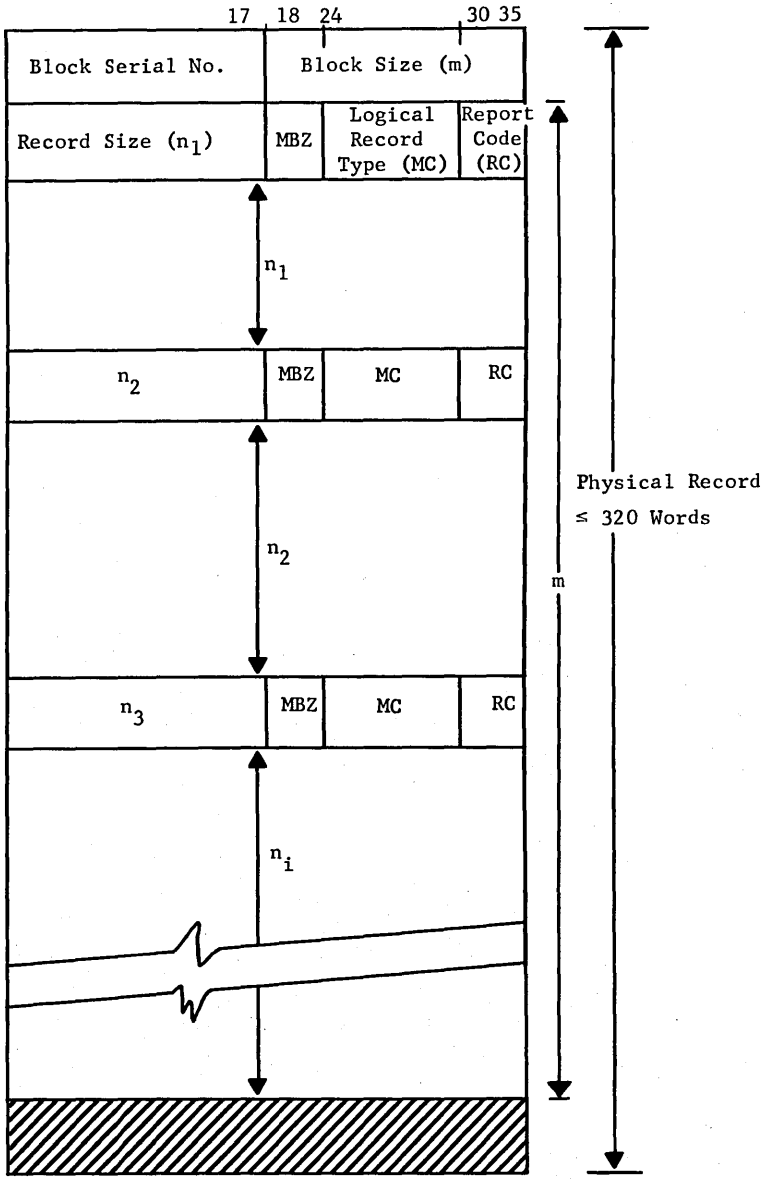


Figure 4. Standard System Format of GECOS System Files

ALTER FILE (A*)

The Alter File (A*) is created by GEIN and used by GMAP, ALGOL, COBOL, JOVIAL and FORTRAN. Upon reading a $ UPDATE card, GEIN puts the $ ALTER cards following that $ UPDATE card into the A* file. The file is then used to insert the appropriate changes into the program being assembled or compiled. The A* file consists of $ ALTER control cards and source or symbolic language cards which are to be merged with the cards of the primary input file. The $ UPDATE and $ ALTER control cards are described in Chapter 5. A detailed description of the alter procedure may be found in GE-625/635 File and Record Control, CPB-1003.

The storage medium for A* is system storage (disc, if using Disc-GECOS; drum, if using Drum-GECOS).

OBJECT PROGRAM FILE (B*)

The Object Program File (B*) is created by GMAP, JOVIAL, or ALGOL and used by GELOAD whenever a $ EXECUTE control card is included in the source deck and no fatal error has occurred during compilation. When GELOAD requires the object deck of the subprogram or program compiled, it will load the object program from B* rather than from R*, which the loader normally reads for its input.

The storage medium for B* is normally disc, but B* can be assigned to tape or drum. During normal allocation of B* by GEIN, the size is initially allocated 2 linked links. These initial allocations may be increased dynamically via GEMORE during their use by the various language processors.

BINARY DECK FILE (C*)

The Binary Deck File (C*) contains the binary object deck created by the GMAP Assembler and/or ALGOL, etc. The file is written for punching in the following format:

$$\left.\begin{array}{l}\text{OBJECT}\\ \cdot\;\cdot\;\cdot\\ \cdot\;\cdot\;\cdot\\ \cdot\;\cdot\;\cdot\\ \$\ \text{DKEND}\end{array}\right\}\text{Object Deck}$$

The C* File is created when the deck option is specified in the appropriate control card ($ GMAP...DECK, $ FORTRAN...DECK, etc.).

The file medium is SYSOUT. The file is limited by SYSOUT to 1000 cards including K*. The report code (bits 30 to 35) of word 1 of the standard system format (see Figure 4) should be 76_8.

GMAP SOURCE FILE (G*)

The GMAP Source File (G*) is created by GEIN, COBOL, or FORTRAN and used by GMAP. GEIN creates the G* File when encountering a $ GMAP control card. The COBOL and FORTRAN compilers create a G* File as their output file. The G* File is then used as input to the GMAP Assembler.

The G* File is stored on system storage (on disc, in Disc-GECOS; on drum, in Drum-GECOS). Initially, G* is allocated 2 linked links; however, the size may be increased dynamically via GEMORE by the various language processor. When used as input to GMAP, the size of G* varies and is dependent upon the number of links necessary to contain the input data involved.

PROGRAM LINK FILE (H*)

The Program Link File (H*) is created by GELOAD and used by the user program for temporary storage of programs that are called into and out of memory. Procedures used to create an H* File are described in GE-625/635 General Loader, CPB-1008. Loading the subsequent links contained on H* is accomplished by using the LLINK or LINK subroutine, as described in the GE-625/635 FORTRAN IV Input/Output Library, CPB-1137. Normally on disc, the H* File may also reside on magnetic tape. This would be done if H* were intended for reuse. Restarting a link job from a saved H* tape is described in CPB-1137.

The loader writes programs onto H* in the order that they are overlayed and not in the order that they are read into memory. The first two words of a linked program contain (1) the link entry point and the location of the link's DEBUG table, if used and (2) the upper and lower address of memory. See also description of the $ LINK control card in Chapter 5.

Unless intended for reuse, H* is usually written on disc and has a size of 5 random links, if a $ LINK card is present.

DATA STORAGE FILE (I*)

The Data Storage File (I*) is created by GEIN and used by the various user programs and the 1401 Simulator (1401SIM). The I* file resides on system storage.

COMPRESSED DECK FILE (K*)

The Compressed Deck File (K*) is created by the GMAP Assembler or any of the system language processors. The file created is an exact duplicate of the source program processed by the language processor, including the merged Alter File input. The K* File is a compressed version of the source input and is created when the COMDK option is specified in the appropriate control card (that is, $ GMAP.... COMDK). This compressed program with changes or corrections can then be used later by the appropriate assembler or compiler as its input. A detailed description of the COMDK format is provided in GE-625/635 File and Record Control, CPB-1003. The K* File is normally assigned to SYSOUT and limited to 1000 cards, including C*. However, it can be assigned to tape, disc, or punched cards by using the appropriate control card (that is, $ DISC.... K* or $ PUNCH.... K*). A detailed description of the control cards available for use is given in Chapter 5.

SYSTEM SUBROUTINE LIBRARY FILE (L*)

The System Subroutine Library File (L*) is created by the System Library Editor and is used by GELOAD.

The L* File contains the system subroutine library. These subroutines are in relocatable binary card image records and are loaded by GELOAD. The subroutines include the FORTRAN I/O, math routines, GEFRC, etc.

The L* File is normally assigned to magnetic tape.

SYSTEM OUTPUT FILE (P*)

The System Output File (P*) is created by the various software systems (GMAP, GELOAD, ALGOL) to collect all systems output. The P* File is normally assigned to SYSOUT. However, if the 10,000-line execution limit (no limit for compiles and assemblies, see Chapter 5) of Disc- or Tape-SYSOUT may be exceeded, P* can be assigned to tape or disc. The output can then be printed by means of Bulk Media Conversion, which has no such limits.

The P* File resides on disc or magnetic tape depending upon which version of SYSOUT is used.

LOADER INPUT FILE (R*)

The Loader Input File (R*) is created by GEIN and used by GELOAD, 1401 SIM, 225 SIM, and 94 SIM. GEIN places all Loader control cards and their associated object programs onto the R* File. When a language processor activity is specified by a control card ($ GMAP, $ ALGOL, etc.), a $ SOURCE card (see Chapter 5) is generated and placed on the R* File. This is in anticipation of loading the newly assembled program from the object program file (B*) in a compile-and-go type of job.

The Loader reads the pertinent control cards and loads the object program from either the R* or the B* file (the B* file is used if the object program was reassembled).

The R* file resides on system storage.

COMPILER SOURCE FILE (S*)

The Compiler Source File (S*) is created by GEIN and used by COBOL, ALGOL, JOVIAL, and FORTRAN. Upon reading a $ COBOL, $ ALGOL, etc., GEIN creates S*, which contains all the source cards following the $ call card. When the compilation is about to be run, the S* File is read and used as source input to the appropriate compiler.

The S* File is contained on system storage normally; however, it may be assigned to drum or magnetic tape.

UTILITY FILE (U*)

The Utility File (U*) is created by GEIN for use with the UTILITY Routine CD600B2.001, (see also $ ABORT, $ UTILITY, and $ FUTIL, described in Chapter 5). The U* File contains the control cards and data for UTILITY on drum; however, it may be assigned to magnetic tape by means of a $ TAPE control card.

IDS SOURCE FILE (*S)

The IDS Source File (*S) is created by GEIN and used to contain IDS-COBOL data statements. The file is used as input to the IDS translator, which puts the statements into a form acceptable for the standard COBOL compiler.

The *S File resides on system storage as a linked file or it may be assigned to magnetic tape.

INTERMEDIATE FILE (*1)

The Intermediate File (*1) is created by FORTRAN, IDS, JOVIAL, COBOL, or GMAP and is used by FORTRAN, IDS, or GMAP for working storage. If the number of links initially allocated to this file is exceeded during the write phase, additional links will be allocated as necessary as long as links are available on the system storage device (see MME GEMORE, Chapter 7).

The *1 File normally resides on system storage. It is allocated 2 or 4 linked links or 2 tape, depending on the size of the source file. However, the size may be increased dynamically via GEMORE during use by the various language processors.

INTERMEDIATE FILE (*2)

The Intermediate File (*2) is created by ALGOL, JOVIAL, or COBOL and used by ALGOL, JOVIAL, or COBOL for working storage. If the number of links initially allocated to this file is exceeded during the write phase, additional links will be allocated as necessary as long as links are available on the system storage device (see MME GEMORE, Chapter 7).

The *2 File normally resides on disc; however, it may be reassigned. It is initially allocated 1 linked link for COBOL programs or 12 linked links for ALGOL programs. However, the size may be increased dynamically via GEMORE during use by ALGOL.

INTERMEDIATE FILE (*3)

The Intermediate File (*3) is created by the COBOL compiler and used by IDS and COBOL for working storage. The allocated space of 15 random links can accommodate approximately 10,000 card images per program. If this limit is exceeded, the user program will be aborted.

7. MASTER MODE ENTRY ROUTINES

In the multiprocessing, multiprogramming environment of the GE-625/635, programs in execution are, of necessity, restricted from accessing one another. However, user programs do have the ability to call upon the pertinent master mode routines for the various services which they require, as mentioned in Chapter 2. This ability is provided through the Master Mode Entry (MME) machine instruction.

When a MME instruction is encountered, the processor (either control or noncontrol) places itself in master mode and control is given to the Fault Supervisor. The Fault Supervisor stores all program registers, stores the instruction counter and the indicators (to allow reference and return to the requesting program), and then examines the address portion of the MME instruction which contains a numerical value corresponding to the symbol used by the programmer. The processor then transfers control to the pertinent MME routine. A nonvalid address in the MME instruction will cause the program to be aborted.

Figure 5 indicates the system mnemonic symbol, its meaning, and the associated value substituted in the MME address field by the assembler. Some MME instructions are not allowed during courtesy call. These are indicated in the figure.

Symbol	Meaning	Value	
		Octal	Decimal
GEINOS	Input/output request	1	1
*GEROAD	Roadblock	2	2
*GEFADD	Physical file address request	3	3
*GERELS	Component release	4	4
*GESNAP	Snapshot dump	5	5
GELAPS	(Elapsed) time request	6	6
*GEFINI	Terminal transfer to monitor	7	7
*GEBORT	Request to abort program	10	8
*GEMORE	Request for additional peripherals or memory	11	9
*GEFCON	File control block request	12	10
*GEFILS	File switching request	13	11
*GESETS	Set switch request	14	12
*GERETS	Reset switch request	15	13
GEENDC	End courtesy call	16	14
*GERELC	Relinquish control	17	15
GESPEC	Special interrupt request	20	16
GETIME	Date and time-of-day request	21	17
*GECALL	System loader	22	18
*GESAVE	Write file in system format	23	19
*GERSTR	Read file in system format	24	20
*GEMREL	Release memory	25	21
GESYOT	Write on SYSOUT	26	22
*GECHEK	Checkpoint	27	23
*GEROUT	Remote output record	30	24
*GEROLL	Reinitiate or rollback program	31	25
*GEUSER	User-supplied MME	32	26
GELOOP	Loop protection	33	27
*GEWAKE	Call-Me-Later	34	28
*GEIDSE	IDS Journalization/Page Range	35	29
*.EMM'	Enter master mode	36	30
*GELBAR	Load base address register	37	31
*GEFRCE	GEFRC entry	40	32
*GEFSYE	File system	41	33
*GEPRIO	I/O priority	42	34

*Not allowed during courtesy calls; see Chapter 11.

Figure 5. Master Mode Symbols and Their Values

Before a return is made to the user, the program registers are restored to their original values, except in the case of those MME's which return information to the user in registers. The various service functions of the MME's are described n the following.

GEINOS -- INPUT/OUTPUT REQUEST

The MME GEINOS instruction is used for all I/O requests (see Chapter 11) on the GE-625/635. The master mode routine that processes the MME GEINOS takes one of the following actions:

1. It aborts the program if any of its select sequence parameters refer to core locations outside the program limits.
2. It aborts the program if the file identification (file code) is illegal for the requesting program.
3. It postpones acceptance of the I/O request if the pertinent request queue is already in use.
4. It queues the request, initiating I/O action immediately, if the channel is not busy.

In general, a request for I/O is made by means of the instruction MME GEINOS followed by a select sequence of 2L+1 words, where L is the number of peripheral commands in the sequence. If the value of L is 1, the select sequence contains three words. If the value of L is 2, the select sequence is five words long.

Entry

Entry to this routine is made from the fault interrupt vector as the result of the MME GEINOS (see also Chapter 11 for definition of the calling parameters).

Calling Sequence

Where L is 1:	MME GEINOS
	Operation word 1
	Identification word 1
	Return word
	Return
Where L is 2:	MME GEINOS
	Operation word 1
	Identification word 1
	Operation word 2
	Identification word 2
	Return word
	Return

Restrictions

Certain I/O requests involve two related operations which must be performed sequentially; for example, SEEK/READ (or WRITE) on the disc or drum and WRITE/READ on the typewriter. Additional restrictions concerning specific parameters are given in Chapter 11.

GEROAD -- ROADBLOCK

The MME GEROAD causes the requesting program to become roadblocked until all of its outstanding I/O requests have been serviced. If there are no outstanding requests, direct return is made to the program.

Entry

Entry to this routine is made from the fault interrupt vector as the result of a MME GEROAD.

Calling Sequence

```
Entry L     MME GEROAD

      L+1   Return
```

Restrictions

The MME GEROAD may not be used during a courtesy call.

GEFADD -- PHYSICAL FILE ADDRESS REQUEST

The MME GEFADD is used to obtain physical addresses for use in operator messages. This routine searches the Slave Peripheral Assignment Table (PAT) for the requesting program, looking for an entry containing the specified file code.

If a match is found for the file code supplied in the Q-register, the A-register will be set to a nonzero value. Return is made to the requesting program with the physical device address in bits 6-17 of the Q-register and with the relative PAT pointer in bits 12-17 of the A-register.

If no match was found in the PAT, both the A-register and Q-registers contain zero. The use of the A-register distinguishes between an unsuccessful search and a successful one in which the IOC channel and device addresses are legitimately all zero.

Entry

Entry to this routine is made from the fault interrupt vector (discussed in Chapter 2) as the result of MME GEFADD. At the time of executing the MME, the Q-register must contain the pertinent file code in bits 24-35.

Calling Sequence

Entry   L   MME GEFADD
         L+1 Return

Initially: C(Q) required

0 23	24 35
Must be zero	File Code (2 characters)

Results: C(Q)

0 5	6 11	12 13	14 17	18 35
Zero	Device Address	IOC	Channel Address	Zero

C(A)

0 11	12 17	18 35
Zero	PAT Pointer	Zero

Restrictions

Slave programs should not use physical addresses in I/O commands. Any physical device addresses specified by a slave program in an I/O command are ignored by IOS, which inserts the proper physical address corresponding to the file code.

No file code match will be found if the specified file code pertains to a file which has been designated on a control card as a SYSOUT file.

GERELS -- COMPONENT RELEASE

The MME GERELS entry is used to deallocate peripherals from a program. For each file code listed in the calling sequence, the corresponding peripheral file is deallocated from the requesting programs. In addition, the MME GERELS may be used to alter either the normal disposition code (first code in LUD) or the abort disposition code (second code in the LUD).

Entry

Entry to this routine is made from the fault vector module as the result of the MME GERELS.

Calling Sequence to Deallocate Peripherals

```
Entry     L        MME       GERELS

          L+1      Zero      No. of files (N) in upper half (No limit for N)
                             Alter flag in lower half:
                             Flag = 0, deallocate peripherals

          L+2      BCI       1,0000F1

          L+3      BCI       1,0000F2

          .

          .                  F1, F2,...., FN-file codes

          .

          L+N+1 BCI          1,0000FN
          L+N+2 Return
```

If this file is on magnetic tape and has a normal disposition code of C, a file protect message is issued for this file. If file is on magnetic tape and has a disposition code of D, dismounting instructions are issued. If file is magnetic tape and has a disposition code of R, the file is released and not rewound unless the file is protected (either by a file protect command or the absence of a write inhibit ring); the file is considered as having a D disposition code. It is rewound and dismounting instructions are issued.

For all other devices, files with disposition codes of D or R are treated as R. If C is specified for nontape devices, it is treated as S. No action is taken if the file has an S disposition code; the file will not be rewound nor released, and no indication will be given to the user that the request was ignored. If more than one file is on a device, the device is not deallocated when a slave program releases the file. The C(L+1) identifies the number of file codes appearing in the calling sequence.

Calling Sequence to Alter Disposition Code

```
Entry     L        MME       GERELS

          L+1      Zero      No. of files (N) in upper half
                             Alter flag in lower half:
                             Flag ≠ 0, alter the disposition code

          L+2      BCI       1,I00SF1 where I is an indicator:
                             I = 0  alter normal disposition code
                             I ≠ 0, alter abort disposition code
                             and S is the substituted disposition code.
                             Either of the following may be used:
                                  R - Release
                                  S - Save
                                  C - Continue
                                  D - Dismount

          L+3      BCI       1,I00SF2
```

.

. FN-file codes

.

L+N+2 Return

Either the normal or abort disposition code for the file FN can be altered at any convenient time in the program including during wrapup (for changing abort disposition codes).

Restrictions

The MME GERELS may not be used during courtesy call.

GESNAP -- SNAPSHOT DUMPS

The MME GESNAP is used to obtain partial memory printouts during the debugging of a program. The snapshot dump produced will be written on the system output file P*, discussed in Chapter 6, for printing on the execution report. The number of print lines produced by GESNAP is not considered as part of SYSOUT print lines.

Entry

Entry to this routine is made from the fault vector module as the result of the MME GESNAP.

Calling Sequence

```
Entry     L      MME      GESNAP
          L+1    VFD      18/A,1/P,2/S,15/N
          L+2    Return
```

where:

A = starting location of the snapshot. A may be symbolic. If A = modulo 8, N+7 is truncated to modulo 8. If A ≠ modulo 8, N+14 and A are truncated to modulo 8.

S = Slew indicator

0 = no slew
>0 = slew to top of page before snapshot dump

N = number of words to be snapped

P = An indicator which, if P ≠ 0, A = 0, and N = 0, allows only the registers to be dumped. Otherwise, registers and allocated memory will be dumped.

N > 0, registers will always be dumped.

N = 0, allocated memory beginning at A will be dumped.

If A is outside the program limits, only the panel will be dumped.

To dump all of allocated memory, when allocated memory is greater than 32k:

```
L      MME      GESNAP
L+1    VFD      18/A,1/P,2/S,15/N
L+2    Return
```

Restrictions

The MME GESNAP may not be used during courtesy call.

GELAPS -- (ELAPSED) TIME REQUEST

The MME GELAPS entry provides the requesting program with the total amount of processor time it has expended up to the time of the request.

Entry

Entry to this routine is made from the fault vector as the result of a MME GELAPS. Return is made to the requesting program with the elapsed processor time right-justified in the Q-register. Time is expressed in number of timer pulses (1 pulse = 1/64 milliseconds).

Calling Sequence

Entry	L	MME GELAPS
	L+1	Return
Results:	C(Q)	

0	35
Elapsed Processor Time	

Restrictions

The amount of processor time charged to a program is greatly affected by the following factors:

1. Competition by an I/O Controller and/or another processor for available memory cycles.
2. Housekeeping inaccuracies unavoidably introduced when a processor responds to a peripheral interrupt or (timer runout) fault. Consequently, the use of MME GELAPS is not recommended for timing studies.

GEFINI -- TERMINAL TRANSFER TO MONITOR

The MME GEFINI entry is used as the standard method of bringing an activity to a successful completion, thus allowing the next activity within the job to be allocated. Core storage allocated to this activity is released, and all peripherals allocated to this activity, except those which have been designated as "save" are de-allocated. Dismounting instructions are issued if necessary.

Entry

Entry to this routine is made from the fault vector module as the result of the MME GEFINI.

Calling Sequence

Entry	MME	GEFINI

Restrictions

All I/O activity should be terminated prior to executing this master mode entry or it will be lost. The MME GEFINI may not be used during courtesy call.

GEBORT -- REQUEST TO ABORT PROGRAM

The MME GEBORT entry is used to bring a job to a sudden stop. Core storage allocated to the activity is released, all peripherals allocated to the activity and deallocated, and dismounting instructions are issued where necessary. The number of print lines produced by GEBORT are not considered part of SYSOUT print lines.

A postmortem memory dump will be written on the execution report if the DUMP option was specified on the System Call Card. If the address of Word 23 (Bits 0-17) is within the slave program limits, return is made totheuser for wrapup (see Chapter 9).

Entry

Entry to this routine is made from the fault vector module as the result of the MME GEBORT.

Calling Sequence

Entry	MME GEBORT
Initially:	C(Q) required

0 23	24 35
Ignored	Reason Code (2 characters)

The choice of reason code is at the discretion of the user.

Restrictions

All I/O activity should be terminated prior to executing this master mode entry or it will be lost.

The MME GEBORT may not be used during courtesy call.

GEMORE -- REQUEST FOR ADDITIONAL PERIPHERALS OR MEMORY

The MME GEMORE entry is used to obtain an additional tape handler or additional links (3840-word blocks) on existing temporary linked disc/drum files allocated to mass storage devices. It may also be used to create a new file. The user must be prepared to accept a denial return if the request cannot be satisfied.

Entry

Entry to this routine is made from the fault vector as a result of a MME GEMORE. At the time of the execution of this MME, the Q-register must contain the pertinent file code in bits 24-35.

Calling Sequence

```
Entry     L      MME       GEMORE

          L+1    Zero      A, B

          L+2    Denial return

          L+3    Successful return
```

Initially: C(Q) if request for peripheral file

0 17	23	24 35
Zero (or Address, see below)		File Code

To request an additional magnetic tape handler:

$(QR)_{24-35}$ contains file code (FC) to be used

A = 1 for 7-track tape; 3 for 9-track tape

B = Scratch/Input designator:

0 if a scratch tape is wanted

1 if a tape is to be mounted. A mounting instruction will be typed:

*MNT TAPE S#sssss i-cc-dd REEL#GEMORE if C(Q) $_{0-23}$ is zero

*MNT TAPE S#sssss i-cc-dd REEL#nnnnn if $C(Q)_{0-17}$ contains an address of a reel no. (nnnnn) or a Ready (*RDY) message will be typed

To request additional links for an existing temporary linked file:

$(QR)_{24-35}$ contains file code of existing file

A = 2

B = No. of links (3840 words each) to be added to file

To request creation of a new file:

$(QR)_{24-35}$ contains a new, unique file code

A = 2

B = No. of links (3840 words each) for the new file

To request additional memory:

A = 0

B = No. of 1024-word blocks wanted

If the request for additional memory is fulfilled, the additional memory will be added to the end (high) of the previously allocated memory.

Restrictions

The MME GEMORE may not be used during courtesy call. The user must be prepared for denial. It is advised that the maximum memory requirements for a given activity be requested initially (prior to allocation).

GEFCON -- FILE CONTROL BLOCK REQUEST

This MME GEFCON entry is provided for the exclusive use of GEFRC, described in GE-625/635 File and Record Control, CPB-1003. It is used to place the following information, identified by the Q-register, in the file control block (FCB), as described in Chapter 6.

- First and last link numbers of the allocated random disc or drum file are stored in Word -7 (LOCSYM -7) of FCB
- Link/random bit (provided for disc or drum)
- SYSOUT indicator
- Type of device
- Physical device address
- File serial number (provided only when the file is defined as magnetic tape)
- Reel sequence number (provided only when the file is defined as magnetic tape)
- File present indicator
- C or D disposition indicator (provided for magnetic tape)

When additional file control blocks are linked to the original file control block (via LOCSYM -1 of the FCB), the above information is supplied for all of the files in the chain. The chain is ended by LOCSYM -1 being zero or pointing to the beginning of the chain.

An alternate function of MME GEFCON is to obtain a magnetic tape reel number from an opened FCB (at LOCSYM -7) and insert it in the PAT (word 3) for this file so that it may be output for accounting purposes

Entry

Entry to this routine is made from the fault interrupt vector module as the result of the MME GEFCON.

Calling Sequence

Entry	L	MME	GEFCON
	L+1	Return	
Initially:		C(Q) required	

0 17	18 35
Location of first FCB	

In its alternate function (see above), bit 0 is turned on (=1); with the normal use of MME GEFCON by GEFRC, bit 0 is off.

Restrictions

The MME GEFCON may not be used during courtesy call.

GEFILS -- FILE SWITCHING REQUEST

The MME GEFILS is used for switching primary and secondary logical units (physical devices). The secondary logical unit for this file code becomes the primary logical unit, and the primary logical unit, in turn, becomes the secondary logical unit. If no secondary unit has been assigned, the peripheral device assigned to the primary logical unit is assumed. The physical device address of the resultant primary logical unit is returned in the Q-register.

Entry

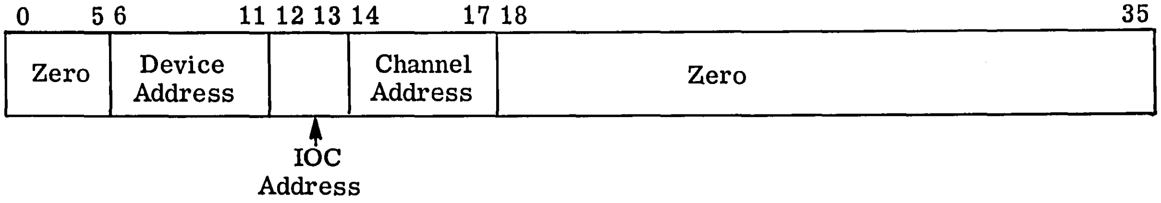
Entry to this routine is made from the fault interrupt vector module as the result of a MME GEFILS. The standard MME call input parameters pertain.

Calling Sequence

Entry	L	MME	GEFILS
	L+1	Return	
Initially:	C(Q) required		

0 23	24 35
Must be zero	File Code

Result: C(Q)



Restrictions

This entry is for use with magnetic tape files only. When files must be switched, it is the user's responsibility to issue any mounting or dismounting instructions to the operator. When using GEFRC, the mounting and dismounting instructions are automatically issued by GEFRC.

GESETS -- SET SWITCH REQUEST

The MME GESETS entry is used to set bits in the program switch word according to corresponding bits in the Q-register. For each bit position of the Q-register that contains a 1, the corresponding bit of the switch word is set to 1. For each bit position of the Q-register that contains a 0, the corresponding bit of the switch word is not changed. The resultant setting of the switch word is returned in the Q-register.

Entry

Entry to this routine is made from the fault interrupt vector module as the result of a MME GESETS.

Calling Sequence

Entry	L	MME GESETS
	L+1	Return
Initially:		C(Q) required

0		35
	36-bit mask	

Results: C(Q)

0		35
	C(switch word)	

Restrictions

The MME GESETS may not be used during courtesy call.

GERETS -- RESET SWITCH REQUEST

The MME GERETS is used to reset bits in the program switch word according to corresponding bits in the Q-register. For each bit position of the Q-register that contains a 1, the corresponding bit of the switch word is set to 0. For each bit position of the Q-register that contains a 0, the corresponding bit of the switch word is not changed. The resultant setting of the switch word is returned in the Q-register.

Entry

Entry to this routine is made from the fault interrupt vector module as the result of a MME GERETS. The standard MME call input parameters pertain.

Calling Sequence

Entry	L	MME GERETS
	L+1	Return

Initially: C(Q) required

0		35
	36-bit mask	

Results: C(Q)

0		35
	C(switch word)	

Restrictions

The MME GERETS may not be used during courtesy call.

GEENDC -- END COURTESY CALL

The MME GEENDC entry charges the elapsed time to the previous program and terminates the courtesy call that was in progress. (For a description of Courtesy Calls, see Chapter 11.)

Entry

Entry to this routine is made from the fault interrupt vector module as the result of a MME GEENDC.

Calling Sequence

Entry	L	MME GEENDC

Restrictions

Time which may be spent in a courtesy call is limited to 200 microseconds on the GE-635 and 400 microseconds on the GE-625. If the courtesy call is not completed in the specified interval, the program will be aborted.

The program will be aborted if MME GEENDC is executed when not within a courtesy call.

GERELC -- RELINQUISH CONTROL

The MME GERELC entry causes one of two actions to be taken, depending on whether or not the program has any outstanding I/O request. If there are any such outstanding requests, the program will become roadblocked until any one of its requests has been serviced. If there are no outstanding I/O requests, the Dispatcher will bypass the program in its first scan of programs requiring execution.

Entry

Entry to this routine is made from the fault interrupt vector module as the result of a MME GERELC.

Calling Sequence

Entry	L	MME GERELC
	L+1	Return

Restrictions

The MME GERELC may not be used during courtesy call.

GESPEC -- SPECIAL INTERRUPT REQUEST

The MME GESPEC entry presets conditions so that a subsequent I/O request will be accepted, but the I/O will not be started until a special interrupt occurs on the device in question. Alternatively, this routine can nullify the effects of a previous MME GESPEC request and delete any unstarted I/O request from the queue.

Entry

Entry to this routine is made from the fault interrupt vector module as the result of a MME GESPEC. The standard MME call input parameters pertain. When the MME GESPEC is executed, the Q-register must have the pertinent file code in bits 24-35. In addition, if the effects of a previous MME GESPEC are to be nullified, bit 0 of the Q-register must be 1.

Return is made to the resulting program with the following code right-justified in the A-register:

0 = Requested action taken

1 = No action taken (file code not in PAT)

2 = No action taken (MME GESPEC illegal for device)

3 = No action taken (channel was busy)

Calling Sequence

Entry	L	MME GESPEC
	L+1	Return

Initially: C(Q) required

0 23	24 35
Must be zero	File Code

Restrictions

The MME GESPEC is allowed for the card reader, card punch, printer, and remote terminals.

GETIME -- DATE AND TIME OF DAY REQUEST

The MME GETIME provides the requesting program with the date and the time of day.

Entry

Entry to this routine is made from the fault interrupt vector module as the result of a MME GETIME. Return is made to the requesting program with the date (in binary coded decimal) in the A-register and the time of day right-justified in the Q-register. Time of day is expressed in number of timer pulses past the previous midnight (1 pulse=1/64 millisecond).

Calling Sequence

Entry	L	MME GETIME
	L+1	Return

C(A):

0 11	12 23	24 35
Month	Day	Year

C(Q):

0 35
Time of day

Restrictions

The time of day returned to the program is that indicated by the simulated clock for the control processor. For a user program in a uniprocessor environment, the value will be correct to within 20 milliseconds. In a multiprocessor environment, the value may be off by as much as several seconds. Consequently, the use of MME GETIME is not recommended for timing studies.

GECALL -- SYSTEM LOADER

The MME GECALL entry makes an entry in the input list of the appropriate service program, enables the execution of that program, and prevents return to the requesting program until the user's request has been successfully carried out.

Entry

Entry to this routine is made from the fault interrupt vector module as the result of a MME GECALL.

Calling Sequence

```
Entry          L          MME   GECALL

               L+1        BCI   1,XXXXXX

               L+2        Zero  L(first word)

               L+3        Zero  Transfer address
```

The system program identified by XXXXXX is loaded into memory beginning at the location specified by C(L+2). If the C(L+3) equals 0, processor control will be transferred to the entry point of the system program. when loading terminates.

If the C(L+3) is not equal to zero, control will be transferred to the location specified by the C(L+3). Upon return, the A- and Q-registers will contain the following:

	0 17	18 35
A	Loading Origin	No. Words
Q	Transfer Address	

Restrictions

The MME GECALL may not be used during courtesy call.

GESAVE -- WRITE FILE IN SYSTEM FORMAT

The MME GESAVE entry makes an entry in the input list of the appropriate service program, enables execution of that program, and prevents return to the requesting program until the same request has been successfully carried out.

Entry

Entry to this routine is made from the fault interrupt vector as the result of a MME GESAVE in a user program.

Calling Sequence

```
Entry          L          MME        GESAVE

               L+1        BCI        1,XXXXXX
```

	L+2	Zero	L(first word), no. of words
	L+3	Zero	Transfer address, Loading increment
	L+4	Return	
Initially:		C(Q) required	

0 23	24 35
	File Code

The C(L+2) specifies the location of the first word and total number of words which are to be saved on the file specified by C(Q). This file can be tape, random disc, or random drum. The C(L+1) is used as an identifier to uniquely identify this data. The C(L+3) specifies an address to which control will be transferred upon restoration of the saved file. This will be used only if the calling sequence for MME GERSTR contains zero in the corresponding transfer address word. The loading increment, if given as the $C(L+3)_L$, is a value to be subtracted from the location of the first word and the transfer address for use in reloading the data via a MME GERSTR. Bit 15 in the Program Switch word must be set to 1 prior to the first MME GESAVE.

Restrictions

Master mode programs (e.g., ALOC, GERTS, GESNAP) programs must not use MME GESAVE. The MME GESAVE must not be used during courtesy call.

GERSTR -- READ FILE IN SYSTEM FORMAT

The MME GERSTR makes a pertinent entry in the input list of the appropriate service program, enables the execution of that program, and prevents return to the requesting program until the user request has been successfully carried out.

Entry

Entry to this routine is made from the fault interrupt vector as the result of a MME GERSTR.

Calling Sequence

Entry	L	MME	GERSTR
	L+1	BCI	1,XXXXXX
	L+2	Zero	L(first word)
	L+3	Zero	Transfer address

Initially: C(Q) required

0 — 23	24 — 35
	File Code

The MME GERSTR restores data saved by GESAVE. The data is identified from the file specified by C(Q) and identified by the C(L+1), and is read into memory starting at the location specified by C(L+2). If C(L+2)=0, the data will be placed in memory according to the data supplied when saved. The C(L+3) specifies the address to which control is to be returned. If C(L+3) is 0, control will be returned to the address previously specified in C(L+3) of GESAVE calling sequence. Upon return, the A- and Q-registers will contain the following:

	0 — 17	18 — 35
A	Loading Origin	No. Words
Q	Transfer Address	

Restrictions

The MME GERSTR may not be used during courtesy call.

GEMREL -- RELEASE MEMORY

This master mode entry is used to de-allocate a specified amount of memory from a requesting program's total assigned memory. Actual memory de-allocation is in multiples of 1024-word blocks.

Entry

Entry to this routine is made from the fault interrupt vector as the result of a MME GEMREL. The A-register and Q-register must contain parameters as described below.

Calling Sequence

```
Entry          L          MME GEMREL
```

The return is made to the address specified in the A-register. When lower memory is released, there will be a change in the base address of the requesting program; in this case the return address is assumed to be relative to the base address after memory de-allocation. When upper memory is released, the base address of the requesting program is not changed.

The number of words released is truncated to modulo 1024. The contents of Word 37_8 of the Slave Program Prefix is not changed by this MME.

Initially: C(A) required

0 17	18 35
Return address	Ignored

C(Q) required

0 17	18 35
Number of words of lower memory	Number of words of upper memory

Restrictions

Release of lower memory should only be attempted by load-type routines because of base-address relocation. The MME GEMREL may not be used during courtesy call.

GESYOT -- WRITE ON SYSOUT

The MME GESYOT is used to transmit output records to the SYSOUT collector media for subsequent printing or punching. SYSOUT collects these records, which may constitute several different reports for each concurrent activity.

The output files (P*, GMAP listing), (C*, binary cards from GMAP), and (K*, COMDK cards) are also assigned to SYSOUT, unless otherwise assigned by control cards (see Chapter 5). Additional files may be assigned to SYSOUT by the $ SYSOUT control card. Files assigned to SYSOUT may not be processed with a MME GEINOS; the MME GESYOT provides the control necessary for such a shared device.

SYSOUT line limits are defined in Chapter 3, "System Summary."

Entry

Entry to this routine is made via the fault interrupt vector as a result of the MME GESYOT.

Calling Sequence:

Normally, SYSOUT will be referenced only by GEFRC where the user reference files via a file control block rather than as a specific device. Therefore, the calling sequence is as follows:

```
L      MME     GESYOT
L+1    Zero    FCB,0
L+2    Return
```

where FCB is the location (word 0) of the file control block for the file containing the records to be transmitted. SYSOUT uses various fields of the file control block to determine the location and quantity of the data to be written.

Restrictions

Only three remote terminals per job may use MME GESYOT. Only 9 report codes per activity are allowed. The total record count for all reports cannot exceed the default allocation limit (see Chapter 12).

GECHEK -- CHECKPOINT

The MME GECHEK entry initiates a checkpoint dump and sets up bookkeeping to enable the requesting program to be read from the last checkpoint made.

Entry

Entry is made with a MME GECHEK in bits 0-17 of the Q-register containing the location of the checkpoint file control block. Bits 18-35 must be zero. This file must have been opened previously via GEFRC (GE-625/635 File and Record Control, CPB-1003). When the peripheral is magnetic tape, any number of checkpoint dumps may be written on the file. However, when the file is identified as disc or drum, only the last checkpoint will be available (each new checkpoint overlays the preceding dumps) and use of the file must be restricted to the sole purpose of checkpoint dumps. No other data may appear. Each checkpoint dump will contain sufficient information for restarting the activity at a later time from the point at which the checkpoint occurred.

Calling Sequence

```
Entry     L      MME      GECHEK

          L+1   TRA       1,QU

          L+2    Next instruction

Initially:  C(Q) required
```

0 — 17	18 — 35
L (checkpoint file control block)	Must be zero

Restrictions

Both the restart procedure and the rollback procedure provide file positioning capabilities only for those files which were open at the time of the checkpoint and have GEFRC file control blocks associated with them. Any file which has been closed between the checkpoint request and the roll back request will not be repositioned. If a file is closed and reopened, a new checkpoint dump should be taken to maintain a complete rollback capability. After once taking a checkpoint, any usage of MME GEMREL or MME GERELS must be followed by a new checkpoint. If this is not done, any subsequent rollback will be aborted with an N3 abort code.

GEFRC provides an automatic checkpoint dump at the beginning of a reel after switching tapes on a multireel file. A connection exists, in this instance, with the instruction TRA 1, QU. Checkpoint and rollback will provide the address +1 of the last MME GECHEK in the upper half of the Q-register. Upon return to the slave program after rollback, a correct entry point is ensured even if an automatic checkpoint was taken by GEFRC.

The user also has the option of providing an additional routine in the event special slave processing is required as a result of a rollback. This routine will automatically be entered if, prior to the first MME GECHEK, the entry location is placed in the upper half of word 14 in the slave memory area. This location may be referenced as follows:

```
1      8      16
WRD14  BOOL   16
```

where the tag WRD14 may be used as an operand.

When using this option, the user then has the responsibility of reentering the main body of his program as he sees fit. The upper half of the Q-register will still contain the location +1 of the last MME GECHEK taken.

A MME GECHEK may not be used during a courtesy call.

Return Codes

After entry to the checkpoint program, certain logical checks are made. The result of these checks are passed back to the slave program in the lower half of the Q-register.

The codes are as follows:

Contents of lower half of Q = 0 Successful checkpoint taken.

Contents of lower half of Q = 1 File specified for checkpoint not open. No checkpoint taken.

Contents of lower half of Q = 2 Checkpoint attempted on illegal peripheral type or on SYSOUT file. No checkpoint taken.

Contents of lower half of Q = 5 Bad status return on attempted write. An incomplete checkpoint dump may exist and a new checkpoint should be taken.

All registers will be set to the state they had immediately prior to the MME GECHEK. After roll back, the right half of the Q-register will have the indicators as they were at the time of the MME GECHEK.

GEROUT -- REMOTE OUTPUT RECORD

The MME GEROUT is used to transmit output records to a remote terminal for subsequent printing or punching. The functions which are performed for direct access mode in GEROUT by the following MME's.

Entry

Entry to all routines is made via the fault interrupt vector.

Calling Sequence for Terminal Direct Access Output

Terminal direct-access output is initiated by:

```
L      MME      GEROUT
L+1    VFD      18/Record Pointer,06/03, H12/Station Identification
L+2    Zero     Status Word Pointer
L+3    Normal   Return
```

Record Pointer points to word 0 of the record which contains, in the left 18 bits, the number of words in the record which are for transmission.

If transmission is to a batch terminal, words 1 to n are formatted as one print line, using 6-bit characters, with an octal 77 indicating end of print line and the following character indicating the slew character. If transmission is to a keyboard terminal, word 1 contains the count of characters to be transmitted to the terminal (a maximum of 320 characters). Word 2 to n contains the actual characters to be transmitted, right-justified within a 9-bit field.

The Status Word Pointer points to a word which contains the status of the operation in process. The format of the status word is as follows:

Bits 0 - 31 = Zero

Bit 32 = 1 if the I/O request cannot be acknowledged because the terminal is busy with a previous I/O.

Bit 33 = 0 if the sending of this output is initiated but not completed. Upon completion, the bit is set to 1.

Bit 34 = 1 if the terminal is disconnected.

Bit 35 = 1 if the terminal operator sent a Break message.

Return is to L+3 immediately after the transmission is initiated. If the station is not available, the slave returns to L+3, and the appropriate bit in the status word is set on.

Calling Sequence for Terminal Direct Access Output Then Input

Terminal direct access output then input is initiated by:

```
L      MME      GEROUT
L+1    VFD      18/Record Pointer, 06/04, H12/Station Identification
L+2    Zero     Status Word Pointer
L+3    Normal   Return
```

The Record Pointer points to word 0 of the record which contains, in the left 18 bits, the number of words in the record which are for transmission and, in the right 18 bits, the address of the input buffer. The output is formatted exactly as under Terminal Direct Access Output. The input buffer is filled as follows:

For teletypewriter terminals:

Word 1 = Count of characters received from the terminal (excluding characters deleted by editing).

Word 2 - 30= Characters received without translation and right-justified in 9-bit fields (teletypewriter terminals) and maximum of 80 characters. The data from batch terminals are in 6-bit fields.

This linkage can be used with a batch terminal. When this is the case, a read card (mixed) command is issued after the write has terminated. This command is repeated until either the remote operator places his response into the reader or the number of repeats have exceeded the limit. The response is then passed on to the slave in standard system format.

The Status Word Pointer is identical to the status word described in Terminal Direct Access Output.

Calling Sequence for Remote Inquiry

This is used by a slave program to establish communication with any terminal that wishes to use the program. In order to accomplish this, the slave program must identify itself by an identification name. This identification is passed on by:

```
L      MME      GEROUT
L+1    VFD      18/0,06/05,12/00
L+2    BCD      Slave Identification
L+3    Normal   Return
```

The location of this call is retained. Whenever a remote station transmits a control card requesting access to a specific slave program, the slave identification in the slave program area is examined for a match in identification. If a match is found, the station identification is placed in the right 12 bits of word directly after the MME GEROUT. The pointer to this MME is deleted from the table. The slave program can interrogate this call from time to time and, when it accepts the request, the slave program can send an appropriate message to the terminal directly.

Calling Sequence for Remote Terminal Type

The slave program requests the terminal type by:

```
L      MME      GEROUT
L+1    VFD      6/Type,12/0,06/06,H12/Station Identification
L+2    Terminal Not Connected
L+3    Normal Return
```

The terminal type is stored into L+1 when a 06 op code is encountered. The type of terminal is determined by the station identification.

If the terminal is not connected, return is to L+2.

Normal return is to L+3.

Restrictions

The MME GEROUT may not be used during courtesy call.

GEROLL--REINTIATE OR ROLLBACK PROGRAM

The MME GEROLL entry results in an output message to the operator, requesting the operator to initiate a restart of the program. If the operator's response is no restart, the program is aborted. If the operator's response is restart, the program is restarted at the last checkpoint within the current activity if such a checkpoint exists.

The user must place a checkpoint (MME GECHEK) at the start of his program; otherwise, if no checkpoint exists, the program is aborted (since GELOAD releases any library files as well as memory itself prior to any possible reinitiation of an activity).

When restart is attempted, positioning of magnetic tape files is accomplished using information available within the user's file control blocks at the time of the checkpoint. The positioning of linked drum and disc files is accomplished using information contained in the Slave-PAT at the time of the checkpoint.

After the checkpoint recovery has taken place, control is returned to the address specified by the contents of the address field of location 14 within the user's assigned memory. If at the time a checkpoint is taken (via GECHEK) the address field of this location is zero, the checkpoint routine will place the address of the MME GECHECK plus one into this location.

Note: Following any file switch, file release, or memory release procedures, the user should issue a MME GECHEK. This will ensure that peripheral unit assignments have not been disturbed between current instruction and last checkpoint.

Calling Sequence

```
        L       MME       GEROLL
```

Restrictions

Any tapes or files assigned to SYSOUT (P*) will not be repositioned.

The contents of the machine registers will be restored to the point immediately prior to the last checkpoint dump. Rollback will always take the positioning information given in the last checkpoint dump and pass over to selected dumps is not possible.

During rollback, the memory size currently allocated must equal the size of memory at the time of checkpoint. If not equal, the program will be aborted.

If a rollback abort and slave dump occurs with the value 3 in the Q-register, the indication is that no checkpoint dump was taken because of an invalid FCB address used as input to the checkpoint program.

GEUSER -- USER-SUPPLIED MME

The MME GEUSER entry supplies the user installation with the capability of adding its own MME for a specified use. Through this entry, an installation may write his own module (.MDMME) tailored to his own needs.

Entry

Entry to a routine in .MDMME is made through the Fault Interrupt Vector as a result of a MME GEUSER in a user program.

Calling Sequence

Specified by user installation.

Restrictions:

The user installation is responsible for maintaining this module. Guidelines for writing a module and some format restrictions of a module may be found in a listing of the UMME Module (.MDMME). At present, the MME GEUSER acts as a No Op. By using this MME, the user undertakes the proof of system failure.

The MME GEUSER may not be used during courtesy call.

GELOOP -- LOOP PROTECTION

The MME GELOOP provides slave program loop protection by dividing the execution time limits into segments. This is particularly useful during program debugging when a program may get into a loop and remain in the loop until the limits specified on the $ LIMITS card are exceeded.

Entry

Entry to this routine is made through the Fault Interrupt Vector as a result of a MME GELOOP in a user program.

Calling Sequence

L	MME GELOOP
L+1	Return

Initially: C(Q) required

```
0                                                              35
| Requested loop time (in seconds), right justified              |
```

Restrictions

Once used to specify a part of the total time limits, the MME must continue to be used for the remaining total time limit, or an I8 abort will be experienced. Note: The user may specify an infinite time (DEC -1) on a succeeding MME(s); in which case, the total limits, specified in the $ LIMITS card, will be used. Under no circumstance will GECOS allow the $ LIMITS card time to be exceeded; MME GELOOP cannot be used to gain additional processor time.

GEWAKE -- CALL-ME-LATER

The MME GEWAKE instruction is used when a program wishes to delay for some time interval before it becomes a candidate again for processor time. Candidacy occurs when the time interval, rounded to milliseconds, has elapsed, or when any outstanding I/O for the program has terminated, or when the program has been enabled by some other GECOS program, whichever comes first.

Entry

Entry to this routine is made from the fault interrupt vector as the result of the MME GEWAKE.

Calling Sequence

```
     L      MME      GEWAKE
     L+1    Return
```

Initially: C(Q) required

0	35
Time interval in 1/64 ms after which this program is enabled (right justified)	

Restrictions

The MME GEWAKE is not to be used during a courtesy call. There is no minimum or maximum time interval. However, if the time interval is greater than 2 seconds, the program may be swapped out of core. An attempt to specify a time greater than 5 minutes or a zero time interval will result in a program abort.

GEIDSE -- JOURNALIZATION AND SUBFILE PAGE RANGE

The MME GEIDSE instruction is used to call the IDS control module. It is restricted to exclusive use by IDS object-time routines.

.EMM -- ENTER MASTER MODE

The MME .EMM instruction is restricted to use by GECOS. An attempt by a user to execute this MME results in an abort of the user program.

GELBAR -- LOAD BASE ADDRESS REGISTER

The MME GELBAR instruction allows a slave program to reset the base address register to a smaller area within his allocated memory.

Entry

Entry to this routine is made from the fault interrupt vector as a result of the MME GELBAR.

Calling Sequence

```
L       MME      GELBAR

L+1     Return
```

Initially: C(Q) required

0	35
Processor time increment before interrupt	

The processor time increment must be less than the amount of time remaining before the timer runs out. The time was assigned when the slave program was allocated. If the increment is greater, its value is set to the remaining allocated time.

Initially: C(A) required

0 17	18 35
LOC1	

where LOC1 and LOC1+1 are:

17	18
BAR	LOC2
IC	I

and LOC2 is the location of the program registers stored prior to issuing the MME GELBAR. The BAR is the desired Base Address Register setting relative to slave zero. The Instruction Counter (IC) is relative to the new base where execution is to be continued. The Indicator (I) equals the setting desired.

When any fault, timer runout,or return after having serviced any system interrupt occurs, GECOS returns to Word 23_8 of the slave program. If Word 23_8 is zero, the program is terminated.

The information relative to GELBAR is stored in Words 17 through 19 (21_8 - 23_8) and Word 25 (31_8) of the Slave Program Prefix. (See Chapter 8.)

Restrictions

The MME GELBAR is not to be used during courtesy call.

GEFRC -- GEFRC ENTRY POINT

This MME is used as an entry point to the GEFRC routines that operate within master mode. Its use is restricted to GEFRC.

GEFSYE -- FILE SYSTEM ENTRY POINT

The MME GEFSYE is used to access the GECOS-III file system.

Entry

Entry to this routine is made from the fault interrupt vector as a result of the MME GEFSYE. Detailed descriptions of the MME are given in GE-625/635 GECOS-III File System Maintenance, CPB-1497.

The calling sequences vary depending upon the desired operation.

Calling Sequence To Create a Catalog

```
L       MME       GEFSYE

L+1     Zero      Courtesy call (or 0),ARGLIST

L+2     Zero      2, BUFFER

L+3     Return
```

The courtesy call should at least be a MME GEENDC; the ARGLIST is as follows:

```
ARGLIST   Zero      Return area,User name
          Zero      Catalog description,Permissions
          Zero      Options
```

The BUFFER is a BSS area of 355 words.

Calling Sequence To Create a File

```
L       MME       GEFSYE

L+1     Zero      Courtesy call (or 0),ARGLIST

L+2     Zero      3, BUFFER

L+3     Return
```

See above for descriptions of calling parameters.

Calling Sequence To Access a File

```
L       MME       GEFSYE

L+1     Zero      Courtesy call (or 0), ARGLIST

L+2     Zero      FUNCTION, BUFFER

L+3     Return
```

See previous page for a description of Courtesy call, ARGLIST, and BUFFER. The FUNCTION is either of the following:

Type 4 - Locate file and set busy

Type 5 - Locate file; do not set busy

Calling Sequence to Deaccess a File

```
L      MME      GEFSYE

L+1    Zero     Courtesy call (or 0),ARGLIST

L+2    Zero     6,BUFFER

L+3    Return
```

See descriptions above for BUFFER and Courtesy call definitions. The ARGLIST is as follows:

```
ARGLIST   Zero     Return area,Offset to PAT pointer
          Zero     Return area,Offset to PAT pointer
          OCT      777777777777
```

Calling Sequence To Purge a Catalog

```
L      MME      GEFSYE

L+1    Zero     Courtesy call (or 0), ARGLIST

L+2    Zero     8,BUFFER

L+3    Return
```

See descriptions above for BUFFER and Courtesy call definitions. The ARGLIST is as follows:

```
ARGLIST   Zero     Return area,User name
          Zero     Catalog description, 0
```

Calling Sequence To Purge a File

```
L      MME      GEFSYE

L+1    Zero     Courtesy call (or 0), ARGLIST

L+2    Zero     9,BUFFER

L+3    Return
```

See above for descriptions of the calling parameters.

Calling Sequence To Modify a Catalog

```
L       MME      GEFSYE

L+1     Zero     Courtesy call (or 0), ARGLIST

L+2     Zero     10, BUFFER

L+3     Return
```

See description above for BUFFER and Courtesy call definitions. The ARGLIST is as follows:

```
ARGLIST   Zero     Return area,User name
          Zero     Catalog description, Permissions
          Zero     Options, Newname
```

Calling Sequence To Modify a File

```
L       MME      GEFSYE

L+1     Zero     Courtesy call (or 0), ARGLIST

L+2     Zero     11, BUFFER

L+3     Return
```

See above for descriptions of the calling parameters.

Calling Sequence To Create, Modify, Query, or Delete Master Catalog * PRIVILEGED SLAVES ONLY

```
L       MME      GEFSYE

L+1     Zero     Courtesy call (or 0), ARGLIST

L+2     Zero     FUNCTION, BUFFER

L+3     Return
```

See description above for BUFFER and Courtesy call definitions. The ARGLIST is as follows:

```
ARGLIST   Zero     Return area, Catalog area
```

The FUNCTION is as follows:

Type 1 - Create system master catalog

Type 12 - Modify system master catalog

Type 13 - Delete system master catalog

Type 14 - Query system master catalog

Type 15 - Query and lock system master catalog

Calling Sequence To Get First Master Catalog Entry

```
L       MME       GEFSYE

L+1     Zero      Courtesy call (or 0), ARGLIST

L+2     Zero      16, BUFFER

L+3     Return
```

See description above for BUFFER and Courtesy call definitions. The ARGLIST is as follows:

```
ARGLIST   Zero      Return area, Record address
          Zero      0, Device name
```

Calling Sequence To Get Next Master Catalog Entry

```
L       MME       GEFSYE

L+1     Zero      Courtesy call (or 0), ARGLIST

L+2               17, BUFFER

L+3     Return
```

See description above for BUFFER and Courtesy call definitions. The ARGLIST is as follows:

```
ARGLIST   Zero      Return area, Record address
          Zero      Catalog name, Device name
```

Calling Sequence To Get Current Catalog/File Description

```
L       MME       GEFSYE

L+1     Zero      Courtesy call (or 0), ARGLIST

L+2               18, BUFFER

L+3     Return
```

See description above for BUFFER and Courtesy call definitions. The ARGLIST is as follows:

```
ARGLIST   Zero      Return area, Record Address
          Zero      Catalog/file description, Device name
```

Calling Sequence To Get First Catalog/File Description

```
L       MME       GEFSYE
L+1     Zero      Courtesy call (or 0), ARGLIST
L+2     Zero      19, BUFFER
L+3     Return
```

See above for descriptions of the calling parameters.

Calling Sequence To Get Next Catalog/File Description

```
L       MME       GEFSYE
L+1     Zero      Courtesy call (or 0), ARGLIST
L+2     Zero      20, BUFFER
L+3     Return
```

See above for descriptions of the calling parameters.

Calling Sequence To Get Specific Address

```
L       MME       GEFSYE
L+1     Zero      Courtesy call (or 0), ARGLIST
L+2     Zero      21, BUFFER
```

See description above for BUFFER and Courtesy call definitions. The ARGLIST is as follows:

```
ARGLIST   Zero      Return area, Record address
          Zero      SPADDR, Device name
```

Where SPADDR is the address of the record to read.

Restrictions

The MME GEFSYE should not be used during a courtesy call. It should be followed by a roadblock at the point at which processing should not continue without a completed request. Only GECOS system programs may use MME GEFSYE. An attempt by a user program to issue a MME GEFSYE will result in a program abort.

GEPRIO -- I/O PRIORITY

The MME GEPRIO instruction is used to assign priority to any I/O for a given file.

Entry

Entry to this routine is made through the Fault Interrupt Vector as a result of a MME GEPRIO in a user program.

Calling Sequence

```
        L       MME      GEPRIO

        L+1     Return

Initially:      C(Q) required
```

23	24 35
	File Code

Restrictions

The MME GEPRIO should not be used during a courtesy call. It is only applicable to tape and must only be used for one file code at a time.

8. SLAVE PROGRAM PREFIX

The Slave Program Prefix resides in the first 64 words of slave memory and is shown in Figure 6. As shown in the figure, the Slave Program Prefix contains the following areas:

- Fault Vector
- GECOS-Reserved Area
- GELOAD Communication Region
- GECOS Initiation and Termination Sequence
- Register Safe-Store Areas

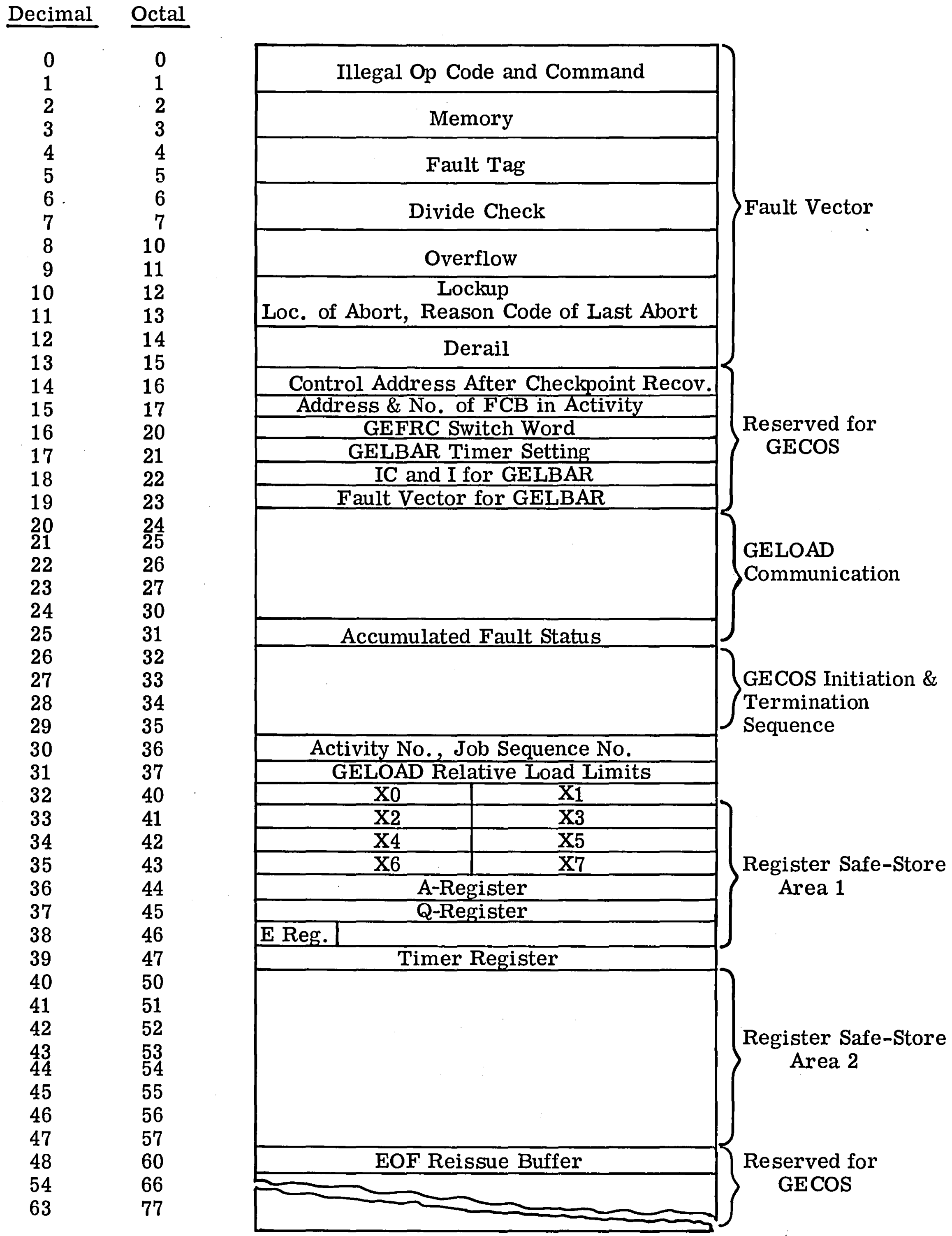
Prior to loading any activity, the entire Slave Program Prefix, with the exception of Words 26 (32_8) through 29 (35_8) is set to zero.

FAULT VECTOR

The Fault Vector comprises the first 14 words (Word 0-13) of the Slave Program Prefix. These words are used in pairs, one for each Fault Trap Cause (with the exception of the Lockup fault).

Word 0 and 1	-	Zero Op code fault Command fault
2 and 3	-	Memory fault
4 and 5	-	Fault tag modifier
6 and 7	-	Divide check fault
8 and 9	-	Overflow fault
10	-	Lockup fault
11	-	Location of abort and the reason code (see Appendix B) of last abort
12 and 13	-	Derail instruction

C (Bar) + Word:



Decimal	Octal		
0	0	Illegal Op Code and Command	Fault Vector
1	1		
2	2	Memory	
3	3		
4	4	Fault Tag	
5	5		
6	6	Divide Check	
7	7		
8	10	Overflow	
9	11		
10	12	Lockup	
11	13	Loc. of Abort, Reason Code of Last Abort	
12	14	Derail	
13	15		
14	16	Control Address After Checkpoint Recov.	Reserved for GECOS
15	17	Address & No. of FCB in Activity	
16	20	GEFRC Switch Word	
17	21	GELBAR Timer Setting	
18	22	IC and I for GELBAR	
19	23	Fault Vector for GELBAR	
20	24		GELOAD Communication
21	25		
22	26		
23	27		
24	30		
25	31	Accumulated Fault Status	
26	32		GECOS Initiation & Termination Sequence
27	33		
28	34		
29	35		
30	36	Activity No., Job Sequence No.	
31	37	GELOAD Relative Load Limits	
32	40	X0 | X1	Register Safe-Store Area 1
33	41	X2 | X3	
34	42	X4 | X5	
35	43	X6 | X7	
36	44	A-Register	
37	45	Q-Register	
38	46	E Reg.	
39	47	Timer Register	
40	50		Register Safe-Store Area 2
41	51		
42	52		
43	53		
44	54		
45	55		
46	56		
47	57		
48	60	EOF Reissue Buffer	Reserved for GECOS
54	66		
63	77		

Figure 6. Slave Program Prefix

Each pair has the following format:

	0 … 17	18 … 35
First word	C(IC)	C(IR)
Second word	Vector	

In the first word, GECOS stores the contents of the Instruction Counter (IC) and Indicator Register (IR) as they were when the fault occurred.

Except in the case of lockup fault (which always causes the program to be aborted), GECOS uses the second word of the pair to determine the action to be taken as a result of the fault:

1. If the second word contains zero, GECOS aborts the program.
2. If the second word is nonzero, GECOS transfers control to it after restoring all registers to their status at the time of the fault. (As stated above, GECOS always aborts after a lockup fault regardless of the contents of the second word.)
3. For divide check and overflow faults, refer to the . SETU. routine, GE-625/635 FORTRAN Input/Output Library, CPB-1137.

GECOS initially clears the second word to zero; following are two examples of how the slave program might use this word:

1. To ignore the fault for any pair (n and n+1), the second word (n+1) is set to:

```
            RET                 n
```

2. To process the fault with a routine at FAULT, the second word is set to:

```
            TRA                 FAULT
```

GECOS-RESERVED AREA

Words 14 through 19 are reserved for GECOS; at present, only words 14, 15, and 16 are used.

Word 14 (16_8) contains the address to which control is to return after checkpoint recovery.

Word 15 (17_8) contains information on the FCB in the activity. Bits 0-17 contain the address of one of the FCB's in the activity. This address is used by GEROLL to find files to be repositioned. Bits 18 through 35 show the number of FCB's in the activity and are used by GEFRC to update the linkage of the FCB's in activity on any OPEN or CLOSE calls.

Word 16 (20_8) is reserved for use as a GEFRC switch word. Currently, bit 18 is used as an interface between . GCLOS and . GR200 regarding output to SYSOUT files.

GELOAD COMMUNICATIONS

Words 20 (24_8), 21 (25_8), and 23 (27_8), and Word 25 (31_8) are used for GELOAD communication.

Word 20 (24_8) has the following format:

0 17	18	19 35
Address of Entry to Main Program		Not Used

Bit 18 = 0 for Normal (Highload)
Bit 18 = 1 for Lowload

Word 21 (25_8) contains the following:

0 17	18 35
Address of the Logical Unit Table	Address of the Destination Code Table

Word 23 (27_8) is used for the wrapup routine and has the following format:

0 17	18 35
Address of Wrapup Routine	Not Used

Word 25 (31_8) has the following format:

0 17		19		21	22	23		29	30 35
BAR from last MME GELBAR		FAULT		OV	EXOV	EXUN	MBZ	DIV	Fault Type

Each overflow, exponent overflow, exponent underflow, and divide check fault handled by GECOS causes the corresponding bit position of this word to be set to one:

Bit Position	Fault
19	Fault occurred within a MME GELBAR
21	Overflow
22	Exponent overflow
23	Exponent underflow
29	Divide check
30 - 35	Fault type inside MME GELBAR: 0 - MME 1 - Memory 2 - Fault tag 3 - Command 4 - Derail 5 - Lockup 6 - Zero op code 7 - Op not complete 8 - Overflow 9 - Divide check 10 - Timer runout

If the user supplies his own fault handling routines, these bits are not affected.

GECOS INITIATION AND TERMINATION SEQUENCE

Words 26 through 29 are used for the Initiation and Termination Sequence. To load GELOAD, the Allocator sets these words as follows:

Word	Contents
26 (32_8)	MME GECALL
27 (33_8)	BCI 1, GELOAD
28 (34_8)	ZERO
29 (35_8)	ZERO

ACTIVITY IDENTIFICATION

The Activity No.· and Job Sequence No. is contained in Word 30 (36_8).

0 5	6 17 35
Activity Number (in octal)	Job Sequence Number

GELOAD RELATIVE LOAD LIMITS

0 17	18 35
Lower Limit	Upper Limit

Lower Limit

The lower limit is the lowest allowable address of the activity to be loaded by GELOAD, relative to the origin of GELOAD.

Upper Limit

The upper limit is the highest allowable address of the activity to be loaded by GELOAD, relative to the origin of GELOAD.

This word is set by GECOS prior to loading GELOAD.

Upon completion of loading, GELOAD sets Word 31 (37_8) of the object memory to the next available lower and upper memory addresses.

REGISTER SAFE-STORE AREAS

There are two register safe-store areas. Both are blocks of 8 words each and are used by GECOS for safe-storing the slave program's registers.

Words 32 (40_8) through 39 (47_8) are used for safe storing slave program registers as follows:

Word (octal)	Content	
	Bits 0-17	Bits 18-35
40	X0	X1
41	X2	X3
42	X4	X5
43	X6	X7
44	A Register	
45	Q Register	
46	E Reg (Bits 0-7)	
47	Timer Register	

GECOS-RESERVED AREA

A portion of the $ IDENT card (col. 13-72) is placed into Word 54 (66_8) through 63 (77_8). The data is right -justified with three leading blanks in Word 48 (66_8).

9. TERMINATION

Three types of termination can occur when running a program under GECOS.

- Normal Termination
- Abnormal Termination
- Disaster Termination

The termination procedure varies depending upon the type of termination, as shown in Figure 7.

NORMAL TERMINATION

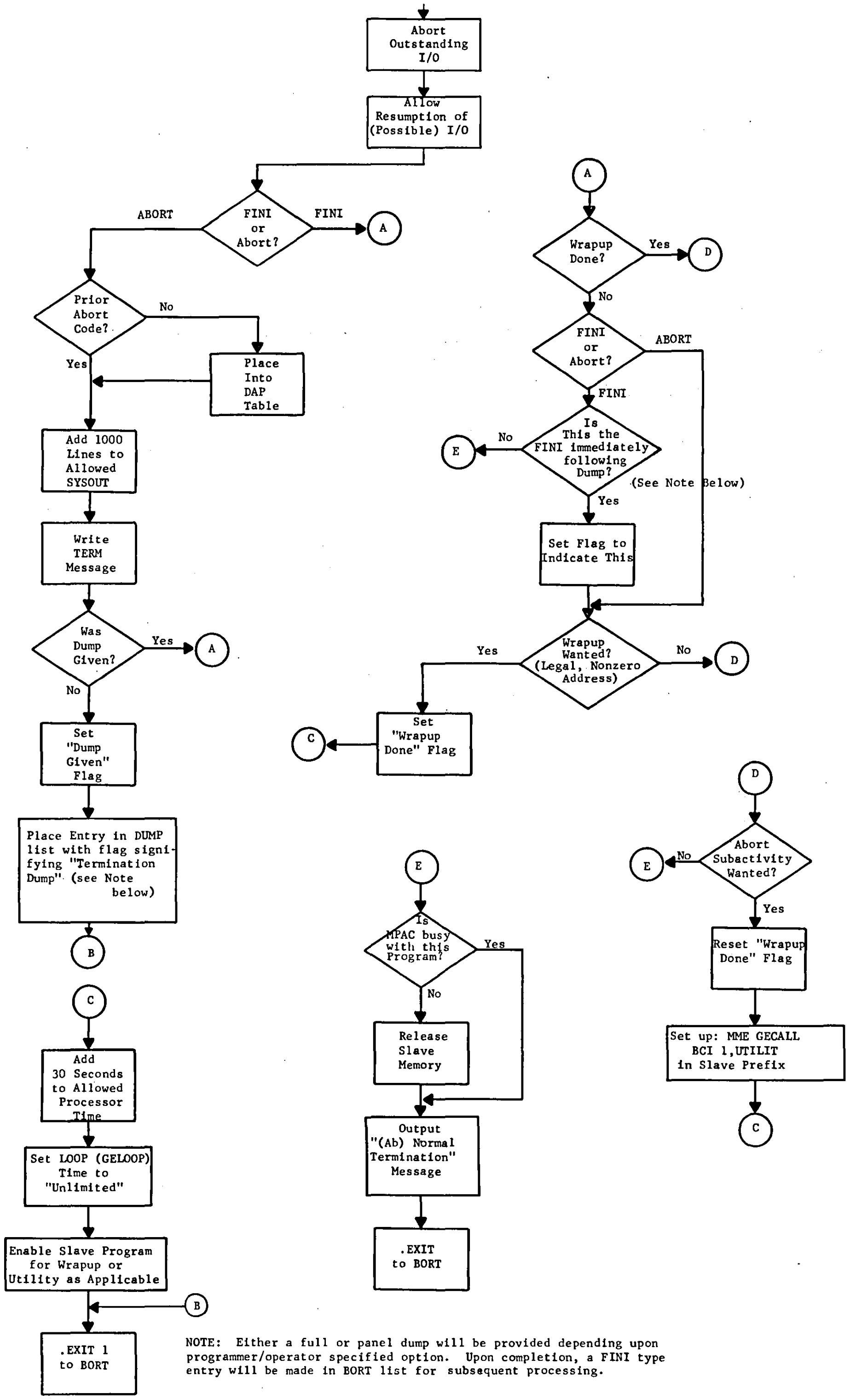
Normal activity occurs when a MME GEFINI is executed by a user program. If an activity terminates normally, the user will not get a dump, wrapup, or an Abort Subactivity, regardless of whether or not his Slave Program Prefix contains a wrapup address or whether or not his input deck contains a $ ABORT card. An accounting record is sent to the accounting file, the execution report is generated, peripheral files are released, and memory is released.

If the activity is the last activity of the job (delimited by a $ ENDJOB card), an *EOJ message is typed on the console and the job is released for print or punch output. If not the last activity, subsequent activities are scheduled for allocation.

ABNORMAL TERMINATION

Abnormal termination may be caused by (1) GECOS-detected errors, (2) operator KILL/ TERM requests, (3) hardware-generated faults, or (4) slave program requests (MME GEBORT). Any of these result in an immediate interruption of the activity. All uninitiated I/O requests are lost; those I/O requests that are in progress are allowed to terminate normally.

The location of the fault or error and the reason code are placed into Word 11 of the Slave Program Prefix. A *ABT message is typed on the control console. If bit 0 of the Program Switch Word is on, a postmortem dump is produced. If Word 31 of the Slave Program Prefix contains a wrapup address, word 31 is cleared and the user is given control at that address. (This word is normally initialized by the .SETU routine, if an Execute activity.) If no wrapup address is found, bit 12 of the Program Switch Word is examined. If set on, a UTILITY Subactivity is executed. Upon return from this subactivity, the accounting and execution records are produced and peripherals and memory are released.


Abort Outstanding I/O
Allow Resumption of (Possible) I/O
FINI or Abort?
ABORT
FINI
A
Prior Abort Code?
No
Yes
Place Into DAP Table
Add 1000 Lines to Allowed SYSOUT
Write TERM Message
Was Dump Given?
Yes
A
No
Set "Dump Given" Flag
Place Entry in DUMP list with flag signifying "Termination Dump" (see Note below)
B
C
Add 30 Seconds to Allowed Processor Time
Set LOOP (GELOOP) Time to "Unlimited"
Enable Slave Program for Wrapup or Utility as Applicable
B
.EXIT 1 to BORT
A
Wrapup Done?
Yes
D
No
FINI or Abort?
ABORT
FINI
Is This the FINI immediately following Dump?
No
E
(See Note Below)
Yes
Set Flag to Indicate This
Wrapup Wanted? (Legal, Nonzero Address)
Yes
No
D
Set "Wrapup Done" Flag
C
E
Is MPAC busy with this Program?
Yes
No
Release Slave Memory
Output "(Ab) Normal Termination" Message
.EXIT to BORT
D
Abort Subactivity Wanted?
No
E
Yes
Reset "Wrapup Done" Flag
Set up: MME GECALL BCI 1,UTILIT in Slave Prefix
C

NOTE: Either a full or panel dump will be provided depending upon programmer/operator specified option. Upon completion, a FINI type entry will be made in BORT list for subsequent processing.

Figure 7. Termination Procedure

If termination was initiated by a KILL/TERM request, the job is terminated as though a $ ENDJOB followed. If termination was not initiated by a KILL/TERM request and a $ ENDJOB is not the activity delimiter, the job is returned, with an Abort indicator set, for subsequent activity allocation. In this case, only subsequent language processing activities (GMAP, FORTRAN, etc.) will be scheduled for allocation until a $ BREAK card is encountered, at which time normal job scheduling is resumed. The conditional control cards $ BREAK, $ GOTO, $ IF, and $ WHEN, along withthe $ Label cards (see Chapter 5) may be used to explicitly control conditional activity execution if the user so desires.

In the case of a MME GEBORT, the program need only supply the reason code, and GECOS will provide the rest. The instruction counter (IC) and reason code are inserted in slave word 11_{10}, as shown in Figure 8.

0 — 17	18 — 23	24 — 35
IC	MBZ	Reason Code

Figure 8. Format of Word 11_{10}

If the job aborted, an entry will be made in the abort queue. The entry in the queue will then be processed as shown in Figure 7.

Abnormal Termination During Allocation

When the first pass through the job control file is completed, a test is made to determine if any errors were found during allocation. If so, the job is deleted, and a *DLT ssss IMPROPER DECK SETUP message is typed. Explicit reasons for deletion are printed on the execution report.

When a job is returned for allocation after execution of at least one activity and no subsequent activities are executable (because of fatal compile errors or activity abortion, for example), an *END sssss message is typed on the console.

If, because of hardware malfunction or unusual deck setup, a fault occurs during allocation, a *TILT, DLT sssss message is typed, a disaster dump is produced (on printer PR2, if dedicated, and on the error and accounting file, if no PR2 is dedicated), and the job is deleted from the system. The disaster dump should be referred to the site Systems Applications personnel.

DUMP, WRAPUP, AND ABORT SUBACTIVITY

If an activity is terminated abnormally (either through a MME GEBORT or an abort) any, all, or none of the following may occur. The occurrence is at the user's option.

- Dump (only if requested via a control card specifying the DUMP option--see Chapter 5, "System Call Cards").
- Wrapup (which may also terminate normally or abnormally).
- Abort Subactivity (which may terminate normally or abnormally).

The console log will show the first abort code; the execution report will show the codes in the order in which they occurred.

Dump

A dump will result only if the user has explicitly requested a dump by either of two procedures:

1. Specified the DUMP option on the control card which defines the activity, or
2. Set (via MME GERETS or GESETS) bit 0 of the Program Switch Word to 1.

When the no dump option (NDUMP) is executed following an abort, a panel dump of the registers will be produced, but no core dump will be given.

Wrapup

Following the possible dump, an optional wrapup is allowed. This will occur if a legal wrapup address has been specified by the user in the upper half of Word 27_8 (23_{10}) of the Slave Program Prefix. Wrapup may terminate normally or it also may undergo abnormal termination. If an abnormal termination occurs during wrapup, a dump will not occur. If a dump is desired, it must be done during wrapup via a MME GESNAP. The user is allowed the balance of the time specified (or implied) on the $ LIMITS card (plus 30 seconds) for wrapup, as shown in Figure 7.

In addition to supplying a legal wrapup address in the Slave Program Prefix, the user must also supply a wrapup routine if it is not supplied for him by the software. In wrapup, the address (bits 0-17) of Word 23_{10} is checked for validity. If the address is out-of-bounds, no wrapup will be done.

Abort Subactivity

An Abort Subactivity may be specified by the user via a $ ABORT control card in his input deck. The subactivity will be initiated if abnormal termination occurs and at least 8k of memory is available for the UTILITY program. If at least 8k is not available, an M4 abort will occur. Again, if the Abort Subactivity undergoes abnormal termination, a wrapup may result. (See GE-625/635 Utility, CPB-1422.) A dump will not occur.

Time Limits for Dump, Wrapup, and Abort Subactivity

The maximum time allowed for any or all of the above (including any associated dumps in the case of Wrapup and Abort Subactivity) is the time specified (or implied) on the $ LIMITS card (plus 30 seconds for each wrapup and/or abort subactivity).

DISASTER TERMINATION

If during the activity termination process, a fault or GECOS-detected error occurs, a PROGRAM DISASTER message is typed on the console and the termination is suspended. No further activity will occur on this job until a system dump and restart is performed. This condition is the result of a hardware or software error and represents an irrecoverable fault.

10. ACCOUNTING INFORMATION

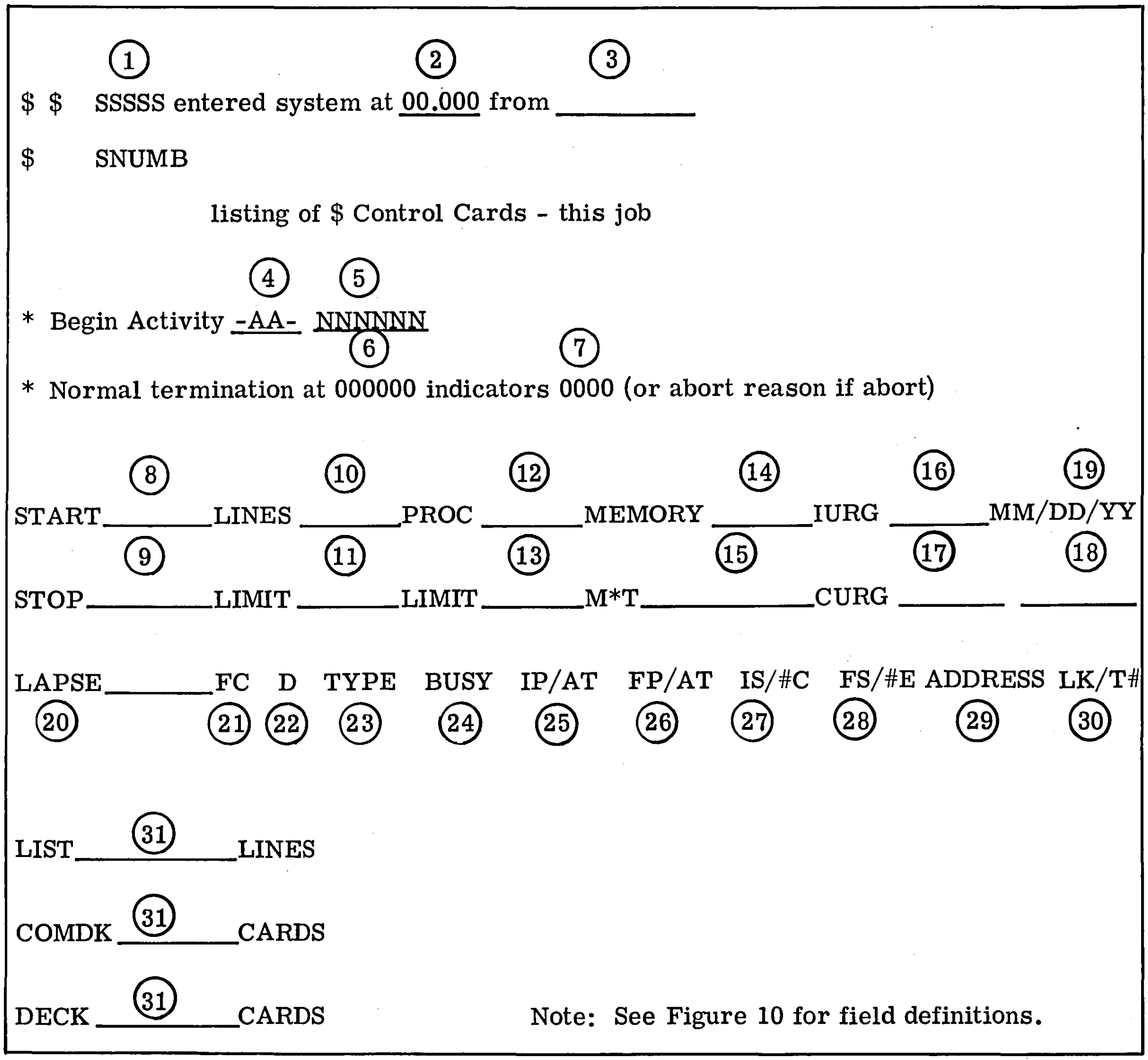
The accounting information for an activity is produced during termination of that activity. It consists of information supplied in the following form:

- Execution Report
- Accounting File Logical Records

Both the lines on the execution report and the binary accounting cards are described in this chapter.

EXECUTION REPORT

The format of fields on the execution report is given in Figure 9. The definition of these fields is given in Figure 10.



```
         ①                     ②       ③
$ $  SSSSS entered system at 00.000 from ________

$    SNUMB

              listing of $ Control Cards - this job

                   ④    ⑤
* Begin Activity -AA-  NNNNNN
                         ⑥          ⑦
* Normal termination at 000000 indicators 0000 (or abort reason if abort)

         ⑧         ⑩         ⑫          ⑭        ⑯        ⑲
START_______LINES_______PROC_______MEMORY_______IURG_______MM/DD/YY
         ⑨         ⑪         ⑬         ⑮         ⑰        ⑱
STOP________LIMIT_______LIMIT______M*T__________CURG_______ _______

LAPSE_______FC  D  TYPE  BUSY  IP/AT  FP/AT  IS/#C  FS/#E ADDRESS LK/T#
  ⑳         ㉑  ㉒   ㉓    ㉔     ㉕     ㉖     ㉗     ㉘      ㉙      ㉚

        ㉛
LIST__________LINES

          ㉛
COMDK_________CARDS

          ㉛
DECK__________CARDS                Note: See Figure 10 for field definitions.
```

Figure 9. Format of Fields of Execution Report

①	SSSSS	- SNUMB
②	00.000	- Time, hours
③	______	- Input source (e.g., CD RDR, IMCV, REMOTE)
④	-AA-	- Activity number
⑤	NNNNNN	- Activity type (e.g. FORTRA)
⑥	000000	- Location of termination
⑦	0000	- Content of indicators at termination
⑧	0.000	- Time job started, hours
⑨	0.000	- Time job stopped, hours
⑩	Lines	- Number of lines of output
⑪	Limit	- Limit on lines of output for this activity
⑫	Proc	- Processor time, hours
⑬	Limit	- Limit on processor time, hours, for this activity
⑭	Memory	- Core allocated to this activity (including slave service area)
⑮	M*T	- Product of (time used core), K-seconds
⑯	IURG	- Initial urgency
⑰	CURG	- Current urgency
⑱	______	- System identification (from $ SYID card in Startup deck)
⑲	MM/DD/YY	- Date
⑳	Lapse	- Elapsed time (Stop - Start)
㉑	FC	- File code
㉒	D	- Disposition code
㉓	Type	- Media type (e.g.TAPE, DRUM, DISC, READ, PRINT), A "P" following DISC or DRUM denotes a permanent file.
㉔	BUSY	- Amount of time channel was busy, ms, if ≤99999; hrs, if >99999.

Figure 10. Definition of Fields on Execution Report

㉕ IP/AT - Initial file position at start of activity

TAPE R/FF	PRINT/PUNCH/READ/PPT
Where R = Records passed FF = Files passed	xx.xxx = Time of allocation, hours

DISC/DRUM
NNN = Blocks passed (320-word blocks)

㉖ FP/AT - Final file position at end of activity

TAPE R/FF	PRINT/PUNCH/READ/PPT
Where R = Records passed FF = Files passed	xx.xxx = Time of deallocation, hours

DISC/DRUM
NNN = Blocks passed (320-word blocks)

㉗ IS/#C - IS = DISC/DRUM Initial Size
NN * = Blocks
NNR = Links (random)
NNL = Links (linked)

/#C = TAPE/PRINT/PUNCH/READ/PPT
NNNN = No. Connects

㉘ FS/#E - FS = DISC/DRUM Final Size
NN * = Blocks
NNR = Links (random)
NNL = Links (linked)

/#E = TAPE/PRINT/PUNCH/READ/PPT
NN = No. errors

㉙ ADDRESS - i-cc-dd

IOC-channel-device no.

Figure 10. (cont'd.)

㉚ LK/T# - Link no. or tape no.
LK/T# (if mass store) (if mag tape)

㉛ ——— - No. of:

Lines printed
Compressed deck cards punched
Object cards punched

Optional fields - if no output of this type was generated, then the field will not be generated.

Figure 10. (cont'd.)

ACCOUNTING FILE RECORDS

The accounting file is maintained in standard system format by GECOS on either mass storage or magnetic tape, whichever is designated by Startup. Each logical record contains a source code identifying the type of information in the records. The format of the activity accounting file record is shown in Figure 11.

0					
1	Record Size	Record Type = 1			
2	System Identification				
3	Month	Day	Year		
4	SNUMB	17$_8$			
5	52$_8$	Activity No.	EOA Type	Abort Code	
6	Activity Mnemonic				
7	Start Time (clock pulses)				
8	Stop Time (clock pulses)				
9	Core Urgency	No. of Activity Compressions	Current Urgency	Not Used	Initial Urgency
10	SYSOUT Limit	SYSOUT Count			
11	Initial No. Memory Blocks	Activity Seq. No.			
12	Product of Memory x Time (k*ms)				
13	Processor Time Used (clock pulses)				
14	Processor Time Limit (clock pulses)				
15	No. Peripheral Entries	No.Report Entries			
16	Col. 13-72 of $ IDENT card				
26	Device Type	GECALL Time (clock pulses)			
27	Device Type	Pushdown Time (clock pulses)			
28	Device Type	SSA I/O Time (clock pulses)			
29	Device Type	GESYOT Time (clock pulses)			
30	Peripheral File Entry (See Figure 12) (5 words per entry)				
35	Peripheral File Entries, as required				
n					

Figure 11. Format of Activity Accounting File Records (Source code 1)

35 3/105

0	Device Type	Address	M	S	F		DISP	File Code
+1	Alternate Address	Channel Use Time (ms)						
+2	No. Connects	No. Errors						
+3	Starting Record	D	Starting File					
	13	File Serial No. if *MNT						
+4	Ending Record	D	Ending File					

S = 1, Tape in Standby, ready message issued
F = 1, File name present
D = density (=1, high = 0, low)
M = 1, mount (see word+3)

Magnetic Tape

0	Device Type	Address	Not Used	DISP	File Code
+1	Zero	Channel Use Time (ms)			
+2	No. Connects	No. Errors			
+3	Time Of Allocation				
+4	Time Of Deallocation				

Unit Record Devices

0	Device Type	Address		DISP	File Code			
+1	Catalog Hash	C		M	P		Channel Use Time (ms)	
+2	Initial No. Links	Final No. Links						
+3	Initial First Link	Initial Rel. Block						
+4	Final First Link	Final Rel. Block						

C = 1, catalog present
M = 1, mode = random
= 0, mode = linked
P = 1, permanent file
0, temporary file

Mass Storage

Figure 12. Format of Peripheral File Entries

When a job enters the system, a logical record is generated containing input statistics (source code 5), as shown in Figure 13. Certain peripheral errors encountered during the execution of the job result in error statistic records on the accounting file (source code 3), as shown in Figure 14. When an activity terminates, a logical record is generated containing the accounting information for the activity (source code 1). (See Figure 11.) Depending on the number of peripherals used by the activity, additional logical records may be generated to contain the overflow of information (source code 8). When a time-sharing user logs off, a logical record is generated containing accounting information for the user identification (source code 7). When the output of the job has been completed for a destination, a logical record is generated containing the output statistics (source code 2), as shown in Figure 15.

0			
+1	Record Size	Record Type = 5	
+2	System Identification		
+3	Month	Day	Year
+4	SNUMB		
+5	GEIN TYPE (CR,IMCV,Remote,TS)		
+6	Job Start Time (clock pulses)		
+7	Job Stop Time (clock pulses)		
+8	Input Record Count (Card count)		
+9	Memory Size		
+10	Processor Time Used		
+11	IOC Time for Input Device		
+12	IOC Time *J File		
+13	IOC Time J* File		

Destination: bit 30 = 1 - Printer
bit 31 = 1 - Remote J*
bit 32 = 1 - Punch
bit 33 = 1 - Remote No. 1
bit 34 = 1 - Remote No. 2
bit 35 = 1 - Remote No. 3

Figure 13. Format of Accounting Record for Input Statistics (Source Code 5)

0											
+1	Record Size	Record Type = 3									
+2	System Identification										
+3	Month	Day	Year								
+4	SNUMB followed by ignore character, if slave										
	Prog. no., if master	Zero, if master									
+5	BCD Act. no., if slave Zero, if master	Type Code (oct.)	A	B	C	Unused					
+6	Time Adjustment for Internal Clock (1/64 ms)										
+7	Time of Interrupt - Internal Clock (1/64 ms)										
+8	Time of Connect - Internal Clock (1/64 ms)										
+9	No. of connects on Device (includ. this I/O)										
+10	No. Errors on Device (including this I/O)	Error Ratio	Unused								
+11	Status Word 1										
	D	E	Maj. St.	SS		F	G		IOC Error		Rec. Residue
+12	I/O Command (second if disc or drum)										
	Dev. Com.	Dev. No.	Ch. No.	IOC Comm	D	Rec. Cnt.					
+13	Seek Address if disc or drum										
	File Serial No. if tape										

A = 1, if GPROVD
B Unused
C GPROVD Special Options:
= 1, CR, Tape Data Alert
= 2, Tape U option
D Sync bit
E Power bit
E = 1, if lost interrupt
G = 1, if initiation interrupt

Figure 14. Format of Error Accounting Record (Source code 3)

<table>
<tr><td>0</td><td colspan="8"></td></tr>
<tr><td>+1</td><td colspan="3">Record Size</td><td colspan="5">Record Type = 2</td></tr>
<tr><td>+2</td><td colspan="8">System Identification</td></tr>
<tr><td>+3</td><td colspan="8">Date from .CRDAT</td></tr>
<tr><td>+4</td><td colspan="7">SNUMB</td><td>Destination</td></tr>
<tr><td>+5</td><td colspan="8">Time of Day (clock pulses)</td></tr>
<tr><td>+6</td><td colspan="2">No. Reports</td><td colspan="6">No. Records</td></tr>
<tr><td>+7</td><td>If on-line Dev. Type</td><td>If on-line IOC/Ch. No.</td><td>0</td><td>A</td><td>B</td><td>D</td><td>0</td><td>Station Identification</td></tr>
</table>

A = 1, if this destination is the last destination for output for this SNUMB

B = 1, if this destination is remote

Destination:
bit 30 = 1 - Printer
bit 31 = 1 - Remote J*
bit 32 = 1 - Punch
bit 33 = 1 - Remote No. 1
bit 34 = 1 - Remote No. 2
bit 35 = 1 - Remote No. 3

Figure 15. Format of Accounting Record for Output Statistics (Source code 2)

When a printer PR2 is not available and is not dedicated, the GESNAP dump is output to the accounting tape. The format of these records is shown in Figure 16. The restart record format is shown in Figure 17.

0	Block Serial No. (inserted by IOS)	Zero	
+1	Record Size	Record Type~~s~~ = 6	
+2	System Identification		
+3	Month	Day	Year
+4	SNUMB	17	
+5	Formatted printer output		
n-1	Formatted printer output		
Word 41 ≥n	Slew code		

Figure 16. Format of Accounting Record for GESNAP Records (Source code 6)

0		
+1	Block Serial No.	Record Size
+2	4	Zero
+3	3	Record Type = 9 or 10
+4	Restart Record	
	00N	
+6		

Record Type 9 - Normal restart record indicating restart occurred

Record Type 10 - Next record is suspect. Data saved may be invalid

N = 1 for normal restart. N is incremented by 1 for each Type 10 record written (to a maximum of 3)

Figure 17. Format of Accounting Records for Restart (Source code 9 or 10)

11. USE OF IOS

The use of the Input/Output Supervisor (IOS) as it applies to the initiation and termination of input and output (I/O) is discussed in detail in this chapter. Included in this discussion are the ground rules associated with using IOS, the select sequence in requesting I/O, and the means by which the user may gain control after specific I/O has terminated (courtesy call).

INITIATION OF INPUT/OUTPUT

When a user program wishes to initiate I/O, it executes a MME GEINOS which is followed by a select sequence specifying additional information pertaining to the I/O operation. When the I/O Supervisor is entered as a result of the MME GEINOS, the file code in the select sequence is used as a basis for determining physical device and channel. If the channel is not busy, IOS will load the communication cells (subsequently to be examined by IOC) and will issue a Connect instruction which will cause the IOC to initiate the request I/O operation.

If, when IOS receives a request to perform an I/O operation, it finds that the channel upon which the request was made is busy, it will queue the request and return control to the requesting program. If no I/O entry space is available in the slave service area, the program will be forced to relinquish within IOS. When I/O has terminated, the user program will again request I/O entry space.

TERMINATION OF INPUT/OUTPUT

Whenever an I/O operation terminates (as signaled by a termination or initiation interrupt), control will be given to the termination portion of IOS after all registers and indicators for the interrupted program have been saved. If the termination was successful (that is, no error condition existed), IOS will return the status of the peripheral device to the program which requested the termination I/O operation.

After IOS has processed the termination interrupt, it will determine if a courtesy call was requested. If it finds that a courtesy call was requested, it will remove the entry from the I/O channel queue and enter it into a courtesy call queue to be dispatched later. If no courtesy call was requested, the entry will be removed from the I/O channel queue and released. If there is another queued request which can be started for the channel, the pertinent I/O action is initiated. Control is returned to the Dispatcher. At that time, any queued courtesy call requests can be paid.

The information on the preceding page has presented a considerably condensed description of the sequence of events that takes place in the execution of a typical I/O operation.

GROUND RULES FOR IOS

The maximum I/O requests queued for each peripheral device is dependent upon I/O entry space available in the program slave service areas. Requests in a slave service area can be for any peripheral assigned to the program.

For those channels which can control more than one device (for example, magnetic tape or disc), I/O requests are normally serviced in the order in which they are received. Requests for files on parallel-access devices may be serviced out of the order in which they are received.

Seek commands addressed to different file units are overlapped automatically by IOS to improve the total throughput of the disc channel.

Inhibition of interrupts are kept to a minimum to reduce response time to such occurrences as peripheral termination.

Returns from I/O requests are made directly to the requesting program. After responding to a peripheral interrupt, IOS always returns to the Dispatcher so that a program may be selected for execution.

Routines to handle the most frequently occurring I/O commands and circumstances are coded for speed, even at the possible expense of increasing execution time in abnormal cases.

In addition to the above ground rules, the following items concerning use of file codes, mass storage files, and write-then-read messages are important to the user of IOS.

- File Codes. A file code is a two-letter symbolic designation used universally to refer to files in the GE-625/635 software system. If the programmer is writing in GMAP (symbolic) language but not using GEFRC, the file code must be provided by the programmer using a BCI pseudo-operation. The file code is contained in the calling sequence following the MME GEINOS which initiates each I/O operation. If the programmer is using GEFRC to perform his I/O operation, GEFRC automatically synthesizes a calling sequence which contains an appropriate file code (specified in the FILCB macro).

  In FORTRAN source language, the file code is created as a function of the file number referred to in the Read or Write commands in the FORTRAN source language. For instance, Write (1) would imply file code 01. In COBOL source language, the file name is related to the file code in the Environment Division using the Assigned To clause.

When a user prepares his program for execution, he prepares $ File control cards (See Chapter 5). These relate the specific file that is to be used in each case to the file code by which it has been referenced internally. IOS translates the file code references at execution time to the actual physical file assignments allocated.

"Random" file

- Mass Storage Files. Two types of mass storage files are treated by IOS. The first is the familiar random access file in which a segment of contiguous storage is allocated on the disc or drum to contain the information of some unique file. IOS may refer to one or more such files for each program. It will make each file appear as though it begins at block 0 and extends upward to an allowable limit. IOS, with the assistance of hardware features, prevents a user from writing beyond the area allocated to him on a mass storage device.

"Linked" file

For many applications it is desirable to be able to interchange disc, drum, and magnetic tape as storage media. To achieve this interchangeability, a type of file referred to as a "linked" file is provided. This type of file causes the mass storage devices to perform as if they were magnetic tape, if the following restrictions are observed:

> Standard system format includes:
>
> Variable block size, 320-word maximum; block sequence number and size of block for each data block; high density, binary mode; variable-length records with record size control word; standard header and trailer labels.) See GE-625/635 File and Record Control manual, for details on standard system format.

- Write then Read. In the multiprogramming environment of the GE-625/635, it is necessary occasionally to type a message on the typewriter which requires an operator reply. If such messages were typed just as they were encountered at execution time, it would often be difficult for the operator to keep up with the multitude of questions which might present themselves to him within a short time interval.

To alleviate this problem, IOS recognizes an altered Write Typewriter I/O command consisting of (130000000002_8) which will specify that there is a Read Typewriter command to be executed immediately following the successful termination of a preceding Write Typewriter command. This pseudo command will allow the operator to respond to each message as it is presented to him.

If after 30 seconds the operator has not responded to a question which has been typed, the console will automatically terminate its Read command with a unique substatus indication to the requesting program (see Appendix C). At this time the requesting program can reissue the command and continue to do so until the operator responds.

SELECT SEQUENCE

In general, a request for I/O is made by means of the instruction MME GEINOS followed by a select sequence of three or five words, depending on the number of peripheral commands in the sequence. Normally, the select sequence contains three words: Operation Word, Identification Word, and Return Word.

A request to perform I/O using a 3-word select sequence may be viewed schematically as shown in Figure 18.

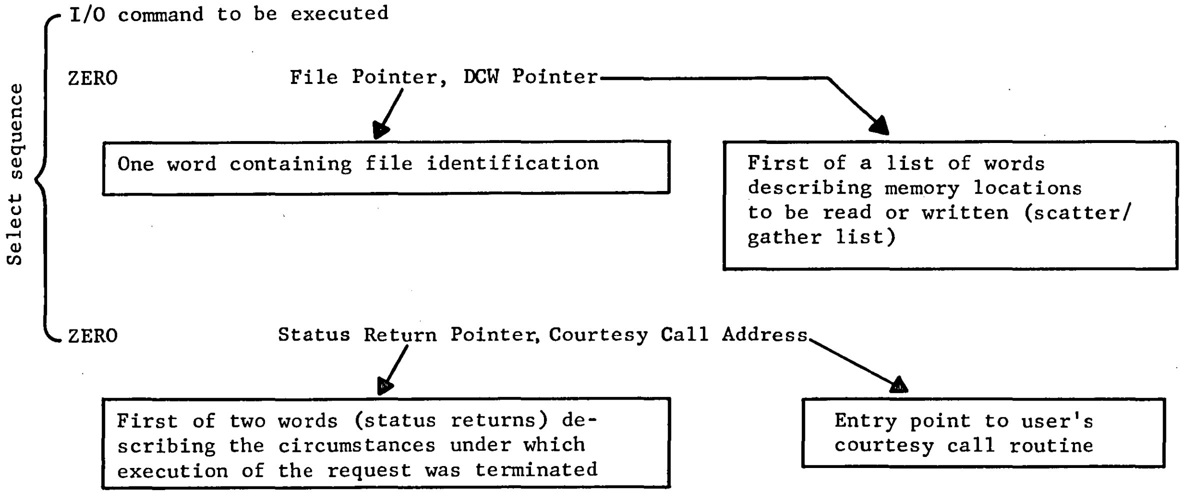

Select sequence
I/O command to be executed
ZERO
File Pointer, DCW Pointer
One word containing file identification
First of a list of words describing memory locations to be read or written (scatter/gather list)
ZERO
Status Return Pointer, Courtesy Call Address
First of two words (status returns) describing the circumstances under which execution of the request was terminated
Entry point to user's courtesy call routine

Figure 18. A 3-Word Select Sequence Requesting I/O

The first word of the select sequence, the operation word, specifies what actual operation is to be performed. The second, the identification word, indirectly identifies which peripheral device is involved and what data, if any, are to be transferred. The third word, the return word, tells where IOS must return the status return pointer and the courtesy call entry point if one is required.

Certain I/O requests involve two related operations which must be performed sequentially. These are Seek, then Read (or Write) Disc; Select, then Read (or Write) Drum; and Write, then Read Typewriter. In these cases, the Select sequence is five words long.

In a 3-word select sequence, then, the order of words is as follows:

- Operation word 1
- Identification word 1
- Return word

A 5-word select sequence consists of the following:

- Operation word 1
- Identification word 1
- Operation word 2

- Identification word 2
- Return word

In Operation Word 1, if the IOC Command is zero and the Count field contains a 2, the select sequence is five words long.

Operation Word

The operation word has the following format:

0 5	6 17	18 22	23	24 29	30 35
Device Command	MBZ	IOC Command	M B Z	Control	Count

The octal configurations of the various Device commands recognized by the GE-625/635 software are given in Figure 19.

		Device Octal Code
All Devices		
REQS	Request status	00
RESS	Reset status	40
Card Reader		
RCB	Read Card Binary	01
RCD	Read Card Decimal	02
RCM	Read Card Mixed	03
Card Punch		
WCB	Write Card Binary	11
WCD	Write Card Decimal	12
WCDE	Write Card Decimal Edited	13
Printer		
WPR	Write Printer	10[a]
WPRC	Write Printer (multirecord)	10[a]
WPRE	Write Printer Edited	30[a]
WPREC	Write Printer Edited (multirecord)	30[a]
Magnetic Tape		
RTB	Read Tape Binary	05
RT9	Read Tape 9	03
RTD	Read Tape Decimal	04
RRTD	Reread Tape Decimal	06
WTB	Write Tape Binary	15
WT9	Write Tape 9	13
WTD	Write Tape Decimal	14
RRTB	Reread Tape Binary	07
WEF	Write End-of-File	55

[a] Printer Slew commands will not be recognized, so that GECOS can have the ability to assign the output either directly to the printer or to another device for subsequent printing. When slewing is to be performed, the escape character and appropriate slew character must appear in the line to be printed.

Figure 19. Octal Configuration of the GE-625/635 Software Device Commands

WFM	Write File Mark	15[b]
WFMD	Write File Mark Decimal	14[b]
ERASE	Erase	54
BSR	Backspace Record (s)	46
BSF	Backspace File (s)	47
FSR	Forward space Record (s)	44
FSF	Forward space File (s)	45
REW	Rewind	70
REWS	Rewind and Standby	72
SLD	Set Low Density	61
SHD	Set High Density	60
<u>Disc</u>		
SDIA	Seek Disc Address	34[c]
RDIC	Read Disc Continuous	25
WDIC	Write Disc Continuous	31
WDICV	Write Disc Continuous and Verify	33
<u>Drum</u>		
SDRA	Select Drum Address	34[c]
RDR	Read Drum	25
WDR	Write Drum	31
WDRV	Write Drum and Verify	33
DRCV	Drum Compare and Verify	11
<u>DATANET-30</u>		
WDN	Write DATANET	10
RDN	Read DATANET	01
<u>Perforated Tape</u>		
RDPT	Read Perforated Tape	02[d]
WPT	Write Perforated Tape	11
WPTE	Write Perforated Tape Edited	31
WPTSC	Write Perforated Tape, Single Character	16
WPTDC	Write Perforated Tape, Double Character	13
<u>Console Typewriter</u>		
RTYP	Read Typewriter	03
WTYP	Write Typewriter	13
WTYPR	Write Typewriter and Return to Read	13

[b] A file mark is written by a combination of a "write tape" Device command and the "write single-character record" IOC command.

[c] When requesting disc (or drum) I/O involving data transfer, the select sequence contains two operation words; the Device command in operation word 1 must be Seek Disc (or Drum) Address, and that in operation word 2 must be the appropriate Read or Write command.

[d] Before issuing a Read Perforated Tape command, the program should first issue an REQS or RESS to insure that a reader plug board is present. If a Read Perforated Tape command is issued and there is no reader plug board present, tape movement will begin and will continue until the end of the input reel is reached, at which time a "Device Attention--Out of Paper" status will be sent.

Figure 19. (cont'd) Octal Configuration of the GE-625/635 Software Device Commands

The various IOC Command codes and their meanings are as follows:

- 00001 = Continuous Mode Non-Data-Transfer (used for backspacing or forward-spacing records or files on magnetic tape and for all other non-data-transfer operations including Request Status and Reset Status)

- 00010 = Card Punch (used whenever a binary or decimal card is to be punched on the 100-cpm punch)

- 00100 = Write Single-Character Record (used to write an end-of-file or tape mark on magnetic tape)

- 00000 = Unit Record Transfer (used for all operations not defined above)

The Control field must be zero except in the following cases:

When writing an end-of-file or tape mark on magnetic tape, the control field contains the binary representation of the character to be written as a single-character record.

The Count field must be zero except in the following cases:

When the Continuous Mode Non-Data-Transfer (00001) IOC command code is used, the count field must be 000001 unless the operation is "backspace or forward space n records," in which case the count field contains n as a binary integer. For this latter purpose, 0 is interpreted as if it were 64.

For disc or drum I/O, the count field (initialized by GMAP) in operation word 1 must contain the value 2 (to indicate that there are two commands in the Select Sequence). The same is true for typewriter I/O whenever the Select sequence contains a Write and Read command.

Identification Word

The identification word has the following format:

0 17	18 35
File Pointer	DCW Pointer

In identification word 1, the File Pointer is the address of the word containing the File Code used to identify the file to which the I/O request pertains. If identification word 2 is present, its file pointer, regardless of actual value, is assumed to be the same as that in identification word 1.

The format of the File Code word is as follows:

0 23	24 35
MBZ	File Code

The File Code is the two-character code used for purposes of file identification.

The DCW (Data Control Word) Pointer contains the address of the first of one or more DCW words which specifies and is used in the control of data transmission. If no data transmission is involved, there are no DCW words; in such a case, the DCW pointer must be zero. If the WPRC or WPREC multirecord command is used for the printer, the DCW Pointer contains the address of the first DCW or of a string of n DCW's. There must be n DCW's where n is the number of records indicated in the command. Each DCW must have a Transmit and Disconnect action code.

Note: The Seek Disc Address and Select Drum Address commands involve the transmission of an address as data, and therefore require a DCW word and a DCW pointer.

The format of a DCW word is as follows:

0 17	18 21	22 23	24 35
Data Address	MBZ	Action Code	Word Count

The Data Address contains the location of the first word to be transmitted or, when the action code specifies a DCW transfer, the location of the next DCW word.

In the case of the DCW word associated with a Seek Disc Address or Select Drum Address command, the word referred to in the data address field must, for random files, be initialized with the desired relative block address each time that I/O is requested on that file; the relative block address is converted to an absolute address by IOS at the time the I/O is initiated and stored back into the word indicated by the data address field. For linked files, all maintenance of the block address is performed by IOS.

Note: When reading or writing linked files, a standard record length of 320 words will always be assumed. Since a block on disc accommodates 40 words, IOS increments the block address by 8 (except across link boundaries) whenever a linked file is accessed. For linked drum files, the block address increment is 5, since a block contains 64 words. For random files, however, transmission can start at any block of 40 words on disc, or of 64 words on drum, by means of the relative block address.

The Action Codes and their meanings are as follows. (The DCW pseudo-operations are listed in parentheses following the Action Code meanings.)

00--Transmit and Disconnect. Move the number of words to or from memory specified by the count and then terminate the I/O command (IOTD).

01--Transmit and Proceed. Move the number of words specified by the count to or from memory, then continue to the next DCW and do as it indicates (IOTP).

10--Transfer to DCW. Don't move any data. Go to the DCW at the address specified in this DCW and do as it indicates (TDCW).

11--Nontransmit and Proceed. Skip the number of words specified in the count of this DCW, then go to the next DCW and do as it indicates (IONTP).

The following restrictions apply to Action Codes:

- The DCW referred to by the DCW Pointer associated with a Seek Disc Address or Select Drum Address command must contain the action code 00.
- The first DCW in a series must not contain the action code 10.
- The action code 10 must not be used in a DCW associated with drum I/O unless it is executed between 64-word blocks.
- In no case may a DCW containing the action code 10 refer to another DCW with the action code 10.
- DCW's for a multirecord command must have an action code 00.

The Word Count contains a binary integer indicative of the number of words to be transmitted under control of the DCW. A count of 0 is interpreted as if it were 4096.

The following restriction applies to Word Count:

The DCW referred to by the DCW pointer associated with a Seek Disc Address or Select Drum Address command must contain a word count of 1.

Return Word

0 17	18 35
Status Return Pointer	Courtesy Call Address

Status Return Pointer. This field contains the address of the first of two sequential words into which the status returns will be placed when I/O has terminated. The first word is zeroed by IOS when the I/O request is accepted. Note that if the requested I/O action is Request Status or Reset Status, the status conditions will be returned to the user activity exactly as IOS receives them from the IOC. Otherwise, the Exception Processing Routine (GEPR) will have taken all possible recovery action on any indicated errors before it returns the final status to the program. In any case, no status returns will be made if the Status Return Pointer is zero.

If the user wishes to exercise special control over the types of error action in response to a nonready status on a peripheral, he may elect to use one of three types of GEPR Override available.

Normal GEPR Override. If the user wishes to suppress all error recovery, normal GEPR Override may be requested. This is done by setting the first status return word to -1 (all ones) prior to issuing the MME GEINOS. IOS will flag the request queue and zero the first status return words in the normal fashion when the request is accepted. When the I/O terminates, the presence of the flag in the request queue causes status to be returned directly to the user without interpretation or error recovery action by GEPR. This is applicable to all devices.

GEPR Override Case 1. If the user wishes to suppress only specifically defined errors, GEPR Override Case 1 may be requested. For all other errors, GEPR will perform normal recovery. To request, the first status return word is set to -2 prior to issuing the MME GEINOS. This function is applicable to magnetic tape devices (Data Alert, Lateral or Longitudinal Parity Error) and to card reader devices (Data Alert and Invalid Character).

GEPR Override Case 2. For specifically defined errors, status will be returned to the user if "U" is one option that would normally be given to the operator after all normal recovery had been taken. For all other errors, GEPR will type the pertinent error message. To request, the first status return word is set to -3 prior to issuing the MME GEINOS. This function is applicable to magnetic tape (Data Alert, Lateral or Longitudinal Parity Error) and disc/drum (Data Alert, Check Character Alert).

GEPR Override Case 15. If the user wishes to suppress error recovery of specified printer and punch errors (Attention on printer or punch; Data Alert, and punch alert on punch), the first status word must be set to -16 prior to issuing the MME GEINOS.

GEPR Abort. Any value of -1 through -16 in the first status return word is legitimate but not necessarily defined GEPR Override request value. A user who experiences an error or exception status on a device for which his GEPR Override function request is undefined will have normal recovery action taken.

If, in addition to a GEPR Override request, the second status return word is set to -1 (all ones), the second status return word will be altered at I/O termination time to give the next DCW pointer in bits 0-17 (replacing the data address). This request for alteration of status return word 2 is implemented to satisfy needs of the 9SA program and would not normally be desired by other users.

It should be pointed out that status returns in response to Rewind and to Rewind and Standby commands are meaningless other than to signify that the I/O action has started. Status is not provided when any of these actions terminate.

In addition, when an I/O request involves two or more I/O actions (for example, Select, then Read Drum), the status returns pertain to the request as a whole.

The formats of the two status return words are as follows:

0	1	2 5	6 11	12 15	16	17	18 23	24 29	30 35
T	POWER	Major Status	Substatus	MBZ	I	ABORT	IOC Status	MBZ	Record Count Residue

0 17	18 20	21	22 23	24 35
	Char. Count	R/W		Word Count Residue

Action

First Status Word

The Termination Indicator (T) will be set to 1 when the I/O (and any pertinent recovery action) has terminated.

For a complete discussion of the Power bit, Major Status, and Substatus, see Appendix C.

The Interrupt Indicator (I) will be set to 0 if the status was returned by the IOC as a result of a termination interrupt but will be 1 in the case of an initiation interrupt.

The latter occurs in the following cases:

The command was Request Status or Reset Status.

The command was Rewind, Rewind and Standby, Set Low Density, or Set High Density with respect to magnetic tape.

The command Backspace Record or Backspace File was addressed to a magnetic tape at the load point and was consequently rejected.

An attempt was made to initiate I/O for the Perforated Tape Reader/Punch when it was in a Device Attention condition.

On the printer, the previous operation resulted in a slew with a top-of-page echo or an attempt was made to initiate I/O when the printer was in a Device Attention condition.

The Abort Indicator is set to 1 when GEPR cannot recover the error reflected in this pair of status words. Slave programs are aborted after status is returned.

The IOC Status will always be 0 except possibly in response to a Request Status in which case this status will be returned by IOS as it is received from the IOC.

The Record Count Residue pertains to magnetic tape or linked file backspace or forward-space operations only. It reflects the number of records remaining that were not spaced over because either the beginning of tape or an end-of-file mark was detected. (Note: an end-of-file mark on tape will have been counted as a record.)

Second Status Word

The second status return word reflects the condition of the last DCW processed by the IOC when the I/O terminated. The fields in this word are as follows:

- The Data Address Residue is the address +1 of the last complete or partial word of data transmitted.
- The Character Count reflects the number of characters transmitted in the last word if it was a partial word; it will be zero if the last word was complete.
- The Read/Write Control (R/W) is used by the IOC for control purposes. Normally it is meaningless to the user program but can be used by hardware diagnostic routines.
- The Action Code is that which was provided in the DCW.
- The Word Count Residue reflects the number of words still left to be transmitted under control of the DCW when I/O terminated.

For a more detailed explanation, see Appendix C.

COURTESY CALL ADDRESS

This field contains the address at which the user wants to gain control when the specified I/O has terminated. The courtesy call address is queued in a courtesy call list at the time of processing the interrupt and is dispatched when interrupt processing has terminated.

Certain restrictions apply:

> It is illegal to use any of the following instructions within a courtesy call: MME GEROAD, GEFADD, GERELS, GESNAP, GEFINI, GEBORT, GEMORE, GEFCON, GEFILS, GESETS, GERETS, GERELC, GECALL, GESAVE, GERSTR, GEMREL, GECHEK, GEROUT, and GEROLL. Depending upon its implementation MME GEUSER may also be prohibited in a courtesy call (See Chapter 7.)

The appearance of any of these instructions within a courtesy call will cause the program to be aborted.

Termination of a courtesy call must be made by means of the instruction MME GEENDC.

The length of time permitted to a courtesy call is limited to 200 microseconds on the GE-635 and 400 microseconds on the GE-625, exclusive of the time required for IOS to process I/O requests. If this time limit is exceeded, the program will be aborted.

Example of IOS Usage

The following example may be helpful in explaining the usage of IOS.

Mode - binary

File Code - TA

Address - DATA

Physical Record Length - 500 (decimal) words

Status Return - STATR

No Courtesy Call

```
         MME         GEINOS
         RTB
         ZERO        FA, DCWP
         ZERO        STATR


FA       BCI         1, 0000TA
STATR    BSS         2             (or DEC      0, 0)
DCWP     IOTD        DATA, 500
DATA     BSS         500
```

12. ALLOCATION

When a new job is received, the job control file (J* File), containing all control cards, is scanned to determine whether the job deck was made-up correctly and which, if any, magnetic tape reels should be retrieved. During this scan, the number of executions and number of $ BREAK cards is accumulated for later decision making. The worst case is determined for memory requirements and magnetic tape needs; processor time and output line limits are accumulated. These values are then optionally used to set urgency and to gauge job length.

At the end of the initial scan of the J* File, the job is deleted if any errors were found, is put in limbo if special resources need retrieval, or is made a candidate for allocation if neither of the above is true. If the job is a lengthy one and is not an urgent job, it is bypassed.

The first time an activity is considered for allocation, the J* File is scanned from its current position to the next activity delimiter to accumulate exact memory and peripheral requirements. The $ File control card information is extracted and condensed into a peripheral requirement summary by type of device. In addition, a peripheral detail entry is constructed for each file specified by a control card or implied because of the type of activity. The peripheral summary is used to perform a gross resource test before any detailed allocation is attempted. The detail entries are used to eliminate the need for performing a character scan of variable fields each time allocation is tried. Keeping a queue of jobs waiting for resources and attempting allocation more than once for each activity prevents hardware delays and maintains a high level of throughput.

Actual allocation is done by matching the details of peripherals needed to the resources available, as described in the System Configuration Tables (SCT), until all needs are satisfied. During this process a Peripheral Assignment Table (PAT) is generated. The PAT allows a user-specified file code to be associated with an entry in the SCT and, at I/O time, with a specific piece of peripheral hardware.

When allocation is successfully completed, an SSA image is written to the first 640 words of the job data file (J* File), and an entry is made in the core allocation queue so the activity can be loaded and executed. At that time, the job status is set to "allocation complete" and is not considered again for allocation until the current activity has terminated.

In allocating files to devices that permit a choice of unit (such as magnetic tape and mass storage), the choices are based on channel of least usage for magnetic tape, and file unit of least usage for mass storage. Usage is defined as the number of files (regardless of size) currently allocated to the channel or unit being considered.

If a file serial number (FSN) is present on the $ TAPE control card, the scan is for a unit in Standby status on which the specified reel may be mounted. If a FSN is not present, the scan is for a unit in Ready status (scratch tape mounted and write-permit ring in) so that no operator action is necessary.

When a tape unit of proper status is not available, a second scan is made to pick a device which can be used. In this scan Ready or Standby status is not considered, but the device must be on line with power on and must be logically assignable but not already assigned as determined by the SCT. If a named device is designated at startup time and is specified as being dedicated, it is only allocated when requested by name.

13. GECOS FILE SYSTEM

The GECOS File System provides automatic management of secondary storage space. From the user's viewpoint, the file system is essentially a means whereby he can store files by name and retrieve them easily. The user need have no concern as to which device a given file is on and also need have little or no concern as to the type of device.

Beyond facilitating basic file storage and retrieval, however, the file system provides several important features:

- Selective access control
- Security protection
- Hierarchical catalog/file structures
- Communication between the batch, remote batch, and time-sharing "worlds" of GECOS-III.

These features are optional, in the sense that if a particular user does not need them, the file system allows him to proceed in the most simple, straightforward manner in regard to them. He may, on the other hand, utilize the file system in as sophisticated a fashion as his needs require.

From the system viewpoint, the file system may be considered the heart of GECOS-III, as it provides:

Permanent on-line data base common to the batch, remote-batch, and time-sharing modes of operation, and

Efficient utilization of high-speed secondary storage space.

The common data base allows a truly integrated multimode system, since the several processing modes not only co-exist, but also communicate with each other. One application of this capability might be to allow a large batch job to generate or update a file (perhaps based upon other files entered from time-sharing terminals) and have that file available for inquiry by time-sharing users.

Another even more interesting application allows a time-sharing user to generate a job for batch processing. The user program in the batch world might be too large to execute conveniently in the time-sharing mode, or may be an existing program for which modification for direct execution is not desirable.

The advantages of unified, efficient management of secondary storage space, especially for on-line shared files, are obvious.

The file system provides a simple, high-level means of defining physical file space and controlling access to it. As such, it has nothing to do with file content. Actual input/output is performed via the standard I/O facilities provided by GECOS, e.g., GEFRC and GMAP. Thus, creating a file refers to using the file system (FILSYS) activity to define a specific extent of space on a particular device. This space is thereby associated with a given file name and file creator.

There are two modes of user-communication with the file system. The batch and remote-batch user communicates via the FILSYS-activity language, which is documented in GE-625/635 File System (FILSYS), CPB-1513. The time-sharing user, at a remote terminal, communicates by means of ~~an accessing~~ subsystem, which is part of the Time-Sharing System and is documented in GE-625/635 Time-Sharing System EDITOR, CP~~B-151~~0 and GE-625/635 Time-Sharing System BASIC Language, CPB-1510.

the ACCESS

TEXT

1515

Both?

Also, p 86.
page CPB-1501

APPENDIX A
TYPICAL DECK SETUPS

Figures 20 through 26 illustrate some representative examples of deck order and the arrangement of the control cards described in Chapter 5.

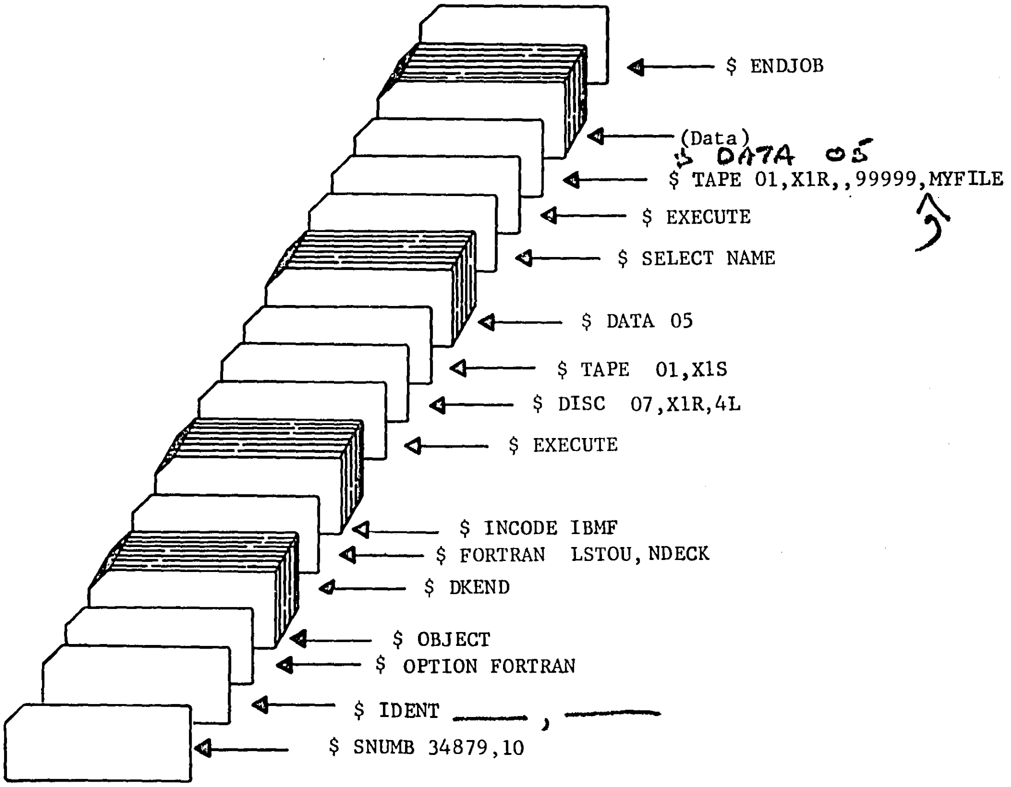


Figure 20. Execution of Two Object Programs

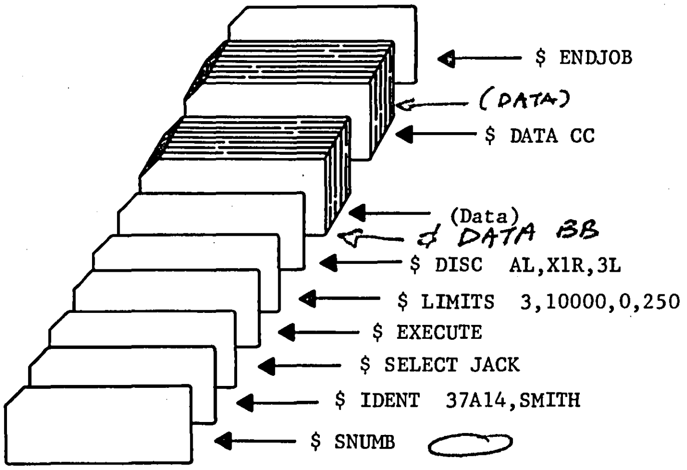


Figure 21. Example of $ SELECT Card Use

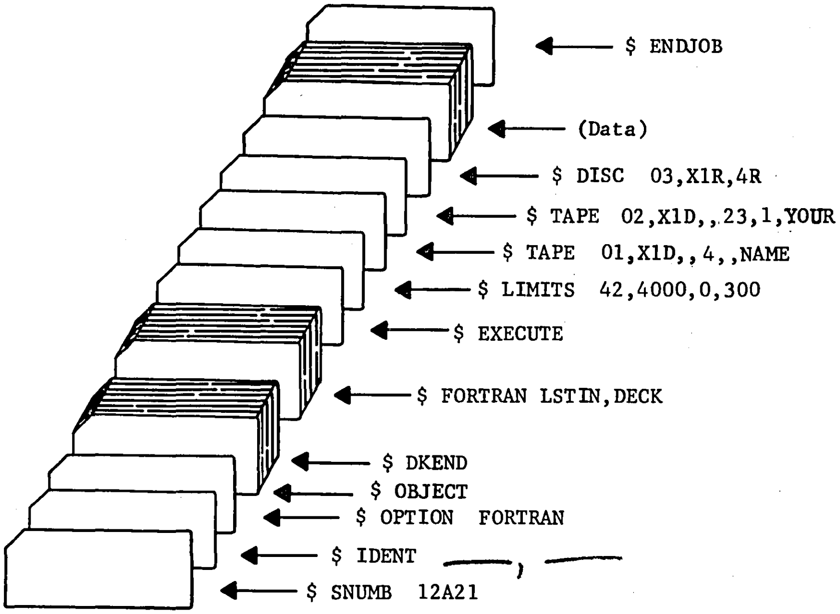


Figure 22. Example of Mixed Object and Source Decks

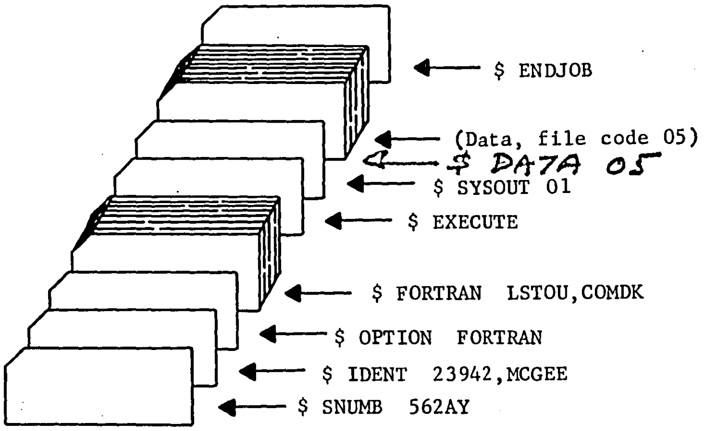


Figure 23. Use of $ SYSOUT Control Card

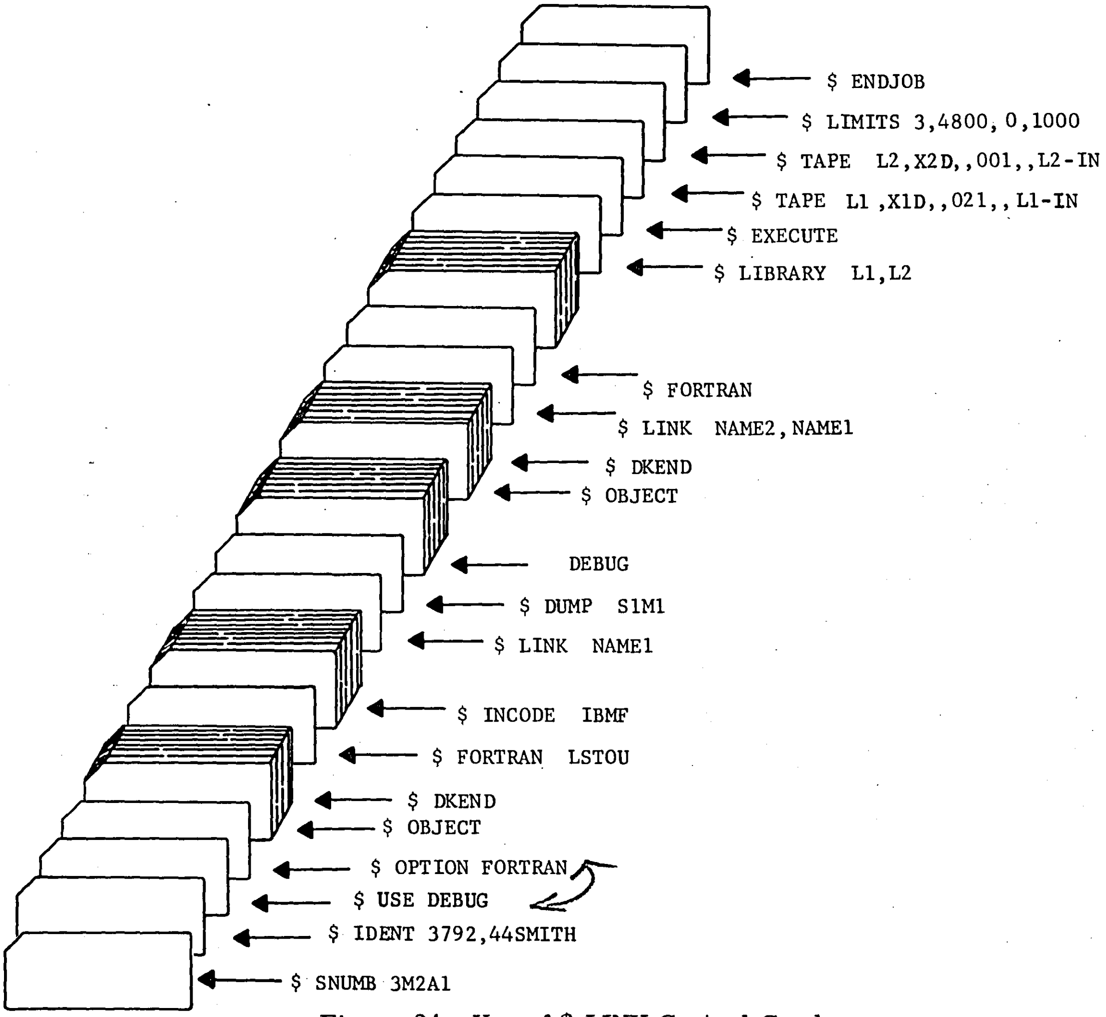


Figure 24. Use of $ LINK Control Card

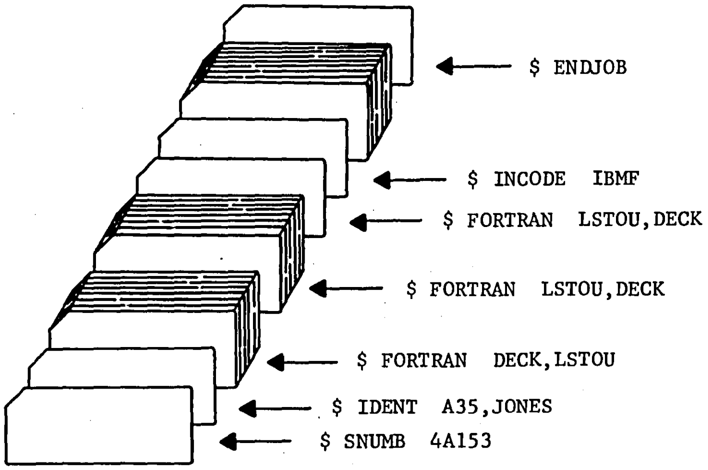


Figure 25. Example of Compile Without Execution

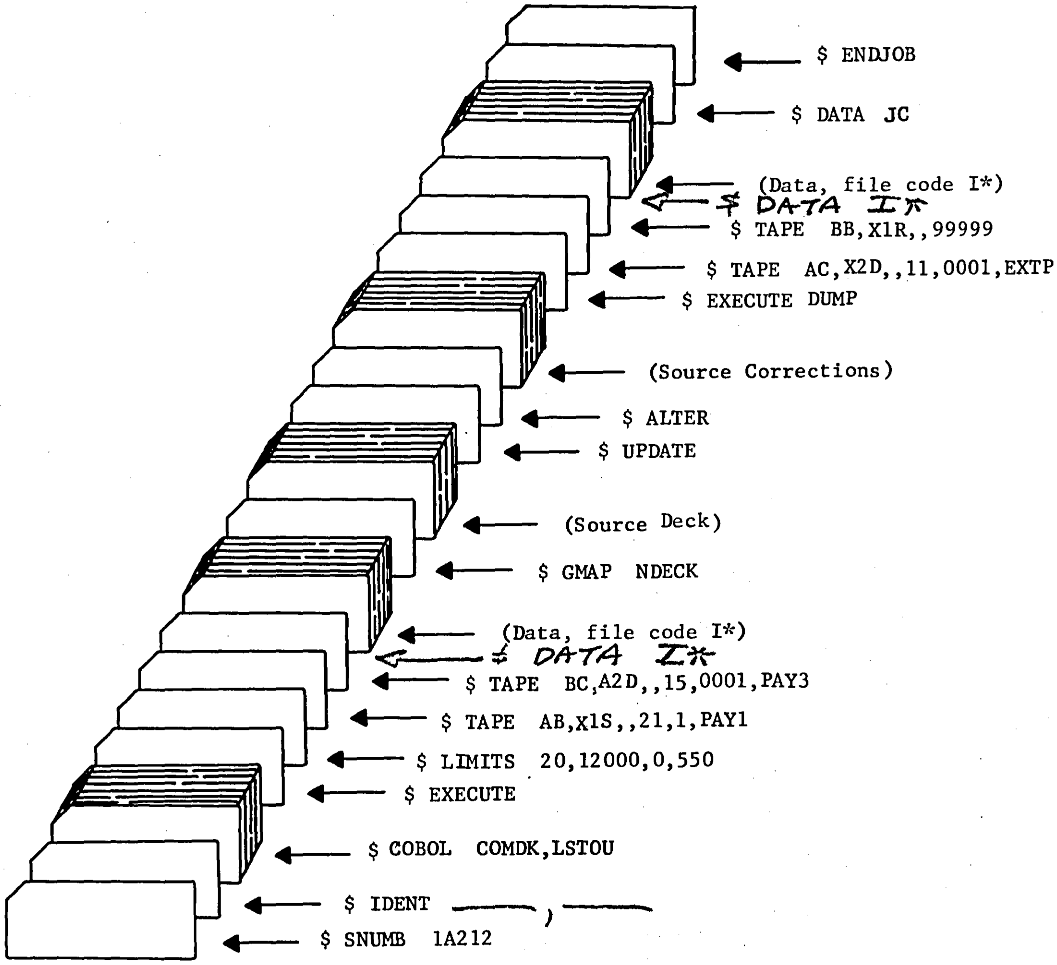


Figure 26. Use of $ ALTER Control Card

APPENDIX B. TERMINATE MESSAGES AND ABORT AND DELETE REASON CODES FOR GECOS-II AND GECOS-III

This Appendix correlates the abort and delete reason codes for GECOS-II with the abort and delete messages for GECOS-III. In addition, a complete list of abort and delete reason codes are given in the following section.

CORRELATION OF GECOS-II AND GECOS-III CODES AND MESSAGES

DELETE CODES

GECOS-II	GECOS-III
J0	JOB DELETED BY OPERATOR REQUEST
J0	JOB DELETED NO ACTIVITY
--	JOB DELETED DUPL. SNUMB
J0	JOB DELETED NEED $ IDENT $ CARD #nn
A4	JOB DELETED NO DEVICE $ CARD #nn
A4	JOB DELETED FILE SIZE $ CARD #nn
J6	JOB DELETED NO SUCH NAME $ CARD #nn
JI	JOB DELETED 9SA NEEDED $ CARD #nn
--	JOB DELETED PFILE ERR #mm $ CARD #nn

where mm = 01 - NAME NOT IN MASTER CATALOG
= 02 - PERMISSION DENIED
= 03 - INCORRECT FILE DESCRIPTION
= 14 - INCORRECT PASSWORD

GECOS-II	GECOS-III
J6	JOB DELETED NO FILE CODE $ CARD #nn
J6	JOB DELETED NO FILE NAME $ CARD #nn
A2	JOB DELETED NTAPE EXCESS $ CARD #nn
AM	JOB DELETED CORE DROUTH $ CARD #nn
--	JOB ENDED, R* TABLE OVERFLOW
A2	ALLOCATION IMPOSSIBLE-xxxx

where xxxx = type of device

GECOS-II	GECOS-III
--	ALLOCATION DENIED PFILE ERR #nn

where nn = card number in the J* file which contains all the control cards and is printed at the beginning of each activity.

ABORT CODES

GECOS-II	GECOS-III
	NO REASON CODE PRESENT
	CANNOT LOAD SSA MODULE
	BAD SSA MODULE CHECKSUM
	CANNOT FIND SSA MODULE
	TOO MANY POP-UPS OF SSA
I8	PROCESS TIME EXHAUSTED
I2	GEROAD IN COURTESY CALL
I6	GEENDC WHILE NOT IN C. C.
	PRIMARY MAILBOX ERROR
	CONNECT OR TERM. ERROR
	IOC DATA-SERVICE ERROR
I7	NON-EXISTENT I/O ADDRESS
	NOT IN MASTER MODE
	MEMORY PARITY IN I/O
	NOT CONTROL PROCESSOR
	PROTECTED AREA I/O
	MEMORY TIME-OUT IN I/O
	OTHER IOC/MEM ERROR
F0	MEMORY ADDRESS FAULT
I1	IMPROPER MME ADDRESS
F1	FAULT TAG ENCOUNTERED
F2	COMMAND FAULT
F3	DERAIL FAULT
F4	LOCKUP FAULT
F6	MEMORY PARITY FAULT
F7	ZERO OPERATION CODE
F9	OVERFLOW/UNDERFLOW FAULT
I0	DIVIDE CHECK FAULT
F8	OP-NOT-COMPLETE FAULT
EP	IOC LIMITS ERROR
I3	FILE CODE NOT DEFINED
I5	NO GEINOS FILE POINTER
I7	ACCESS OUT OF FILE LIMIT
K2	INVALID STATUS RET. PTR.
K3	INVALID FILE POINTER
K4	INVALID DCW POINTER
K5	INVALID COURTESY ADDRESS
K6	INVALID I/O COMMAND-FILE
K7	TWO TDCWS IN SUCCESSION

GECOS-II	GECOS-III
K1	INVALID I/O FOR DEVICE
X1	OPERATOR REQUESTED TERM
X2	OPERATOR REQUESTED KILL
M6	I/O LIMIT CALL/SAVE/RSTR
M6	I/O ERROR CALL/SAVE/RSTR
M5	NO PAT FOR GECALL FILE
M1	WRONG CALL/SAVE DEVICE
M1	NON-RANDOM GECALL FILE
N8	NO SPACE IN GESAVE FILE
M6	CALL/RSTR CHECKSUM ERROR
	LOW GERSTR ORIGIN USED
N4	NO GESAVE WORD COUNT
M3	CALL NAME NOT IN CATALOG
I7	DATA OUT OF FILE RANGE
	IMPROPER GECOS CALL
M0	GEMORE - ZERO FILECODE
M0	INVALID GEMORE PARAMETER
	GEMORE - NO ROOM IN PAT
M7	GEMORE - NOT MASS STORE
R5	GEROUT - INVALID OP CODE
	GEROUT PAT EXHAUSTED
RA	REMOTE TERMINAL REC SIZE
RL	UNDEF REMOTE STATION ID
RH	INVALID GEROUT PROG. ID
0>	SYSOUT LIMITS EXCEEDED
	SYSOUT RECORD SIZE ERROR
	SYSOUT SEEK ERROR
	SYSOUT ALLOCATION ERROR
	L(SYOT STATUS) > LIMITS
	L(SYOT BUFFER) > LIMITS
	SYOT BUFFER END > LIMITS
EP	IRRECOVERABLE I/O ERROR
KU	INVALID SEEK DCW
	IMPROPER MEDIA CODE
N3	ROLLBACK NOT POSSIBLE
	TSS REQUESTED TERM
	GEMORE - DUPL. FILECODE
	INVALID I/O SCT POINTER
	PERM WRITE W/O PERMIT
	PERM READ W/O PERMIT

REASON CODES (CC) USED IN SYSTEM ABORT AND DELETE MESSAGES AND SOURCE OF THE ABORT

Code	Source	Explanation
0>	GECOS-II (SYSOUT)	Excessive output of GESYOT.
01		Improper GESYOT call sequence code.
A>	GECOS-II (TALC)	More than 638 links requested on $ DISC/$ DRUM linked file.
A1		Peripheral Assignment Table (PAT) buffer size exceeded.
A2		Allocation impossible because of channel requirements (devices requested are in excess of those available on a given channel).
A3		Secondary tape buffer size exceeded.
A4		Peripheral requirements in excess of actual resources have presented an impossible allocation task.
A5		Abeyance table size exceeded.
A6	GERTS-II	More than 6 output files requested.
A7	GECOS-II (GEIN)	Identical logical unit designators have been used for mass storage requests on the same type of device.
A8	GECOS-II (TALC)	Allocation impossible because of channel requirements (as previously defined by one or more peripherals in abeyance).
A9		Named device not configured, "RLSE'd," or no such name.
AD		Zero links requested on $ DISC/$ DRUM card or improper deck setup.
B1	BMC (See <u>GE-625/635 Bulk Media Conversion</u>, CPB-1096.)	Variable error - Improper use of variables on a $ OUTPUT or $ INPUT card. Check for spelling, use of MIXED for other than card input, use of MIXL on input, etc.
B2		Variables on the $ MULTI control card are incorrectly used.
B3		Improper request for transliteration. Either the media combination is in error or codes (IBMF, GE225, etc.) are improperly interpreted. Transliteration applies only to the card reader; assigning it to any other device will cause a B3 abort.
B4		Input or output device code is not acceptable to BMC.
B5		A partial header label has been encountered on magnetic tape input.
B6		User has not provided an Sxxxxx or Uxxxxx option (input only), and an error has occurred.
B7		The number (xxxxx) of acceptable input errors as specified in Uxxxxx or Sxxxxx has been exceeded.
B8		Input error prevents BMC from continuing.
B9		An entry made to an overlay is erroneous; probable hardware failure.

Code	Source	Explanation
C0		A requested input/output file is not contained in the file address table; an internal problem in the compiler.
C1	COBOL	Drum address table (*3 File) is full. Normally, this problem may be corrected by increasing the size of the *3 File with a $ DISC or $ DRUM card. Maximum limits for the *3 File is 50 random links.
C2		Fixed portion of the data name table is filled to the point that the largest record (01) contained on the overflow file cannot be read back into memory. Normally, this problem may be corrected by including a $ LIMITS card to extend memory limits, thus enabling extension of the data name table.
C3		Invalid internal list structure of the Data Division, possibly caused by invalid level structures for some data items which have been processed prior to this point.
C4	COBOL	Substantive Stack build error in Report Writer. Occurs if the substantive stack is empty and the overflow indicator is not on when the Substantive Stack Popup routine is called.
C5		Invalid internal list structures of the Report Writer, which is an internal Compiler problem.
C6		Fixed portion of the Report Table is full. See GE-625/635 COBOL, CPB-1007, for ways in which changes can be made to the Report Section (report descriptions) of the source program so that abort is eliminated.
C7		Fixed and variable portion of the Report Table is full. See GE-625/635 COBOL, CPB-1007, for ways in which changes can be made to the Report Section (report descriptions) of the source program so that abort is eliminated.
C8		Invalid internal list structures of the Report Writer Analyzers, which is an internal Compiler problem.
C9		Report Writer Generator error; Report Generator is unable to build its COBOL lists (internal language) using the Analyzer build routines. An internal Compiler problem.
CA		Invalid internal list structures of the Report Writer Generators, which is an internal Compiler problem.
CB	COBOL/FORTRAN	Invalid (not 17_8 or 23_8) end-of-file mark has been encountered in a COBOL or FORTRAN object program.
D1	IDS	IDS has aborted program for cause, as indicated by its abort message. (See Integrated Data Store, CPB-1093.)
D2		User error.
EF	PRODUCT	Input tape in improper format for PRODUCT.
E(n)	GECOS-II (GEPR)	Undefined GEPR override option requested. (n) = absolute value of first status word when the I/O request was initiated.
EP	GECOS-II (GEPR)	Irrecoverable I/O Error. (IC will contain location of status return pointer; indicators will contain courtesy call address, if any.)
EX	Operator-initiated	EXECUTE button at processor T & O Panel depressed (Operator-initiated abort). Typeout will appear at the console.

Code	Source	Explanation
F0	Machine-Detected Fault	Memory address fault (attempt to access memory beyond BAR).
F1		Fault tag (tag field -40)
F2		Command fault (slave execute of CIOC, RMCM, SMCM, RMFP, SMFP, SMIC).
F3*		Derail fault (Op code = 002)*.
F4		Lockup fault (the processor is in a program lockup which inhibits recognizing an Execute Interrupt or Interrupt-type fault for greater than 16 milliseconds)
F5		Connect fault (Control processor, in master mode, executed CIOC addressed to a memory port containing a processor (the latter processor incurs the connect fault)
F6		Parity fault (A parity error was detected in a word which was read from a core location)
F7		Zero Op Code fault
F8		Op Not Complete fault
F9		Overflow/Underflow fault
GF	GEFRC	GEFRC has aborted program for cause as indicated by its abort message (see GE-625/635 Operator's Reference Manual, CPB-1045)
G1		Input Header Label
G2		Error in Input Tape Header label on tape
G3		Error in Output Tape Header label
G4		Error in old label on output tape
G5		End-of-tape while writing header label
G6		Blank tape rather than input trailer label
G7		Block count error in input tape trailer label
G8		End-of-tape detected on multifile reel. (Note: A multifile reel cannot be continued on a second tape)
G9	GECOS-II (FCON)	New FCB address is the same as present one in the MME GEFCON sequence.
I0	GECOS-II (IOS)	Divide Check fault
I1		Invalid address in MME
I2		Improper use of MME during courtesy call
I3		File code not in Peripheral Assignment Table (PAT), or user attempts to execute a MME GEINOS with a file code specifying a SYSOUT file
I4		Invalid PAT pointer
I5		No file pointer in GEINOS select sequence
I6		GEENDC used when not in courtesy call
I7		Attempt to access outside file limits
I8		Allowable run time exceeded
I9		Too long in courtesy call

* See CPB-1469, Appendix B, for a list of derail codes and their meanings.

Code	Source	Explanation
J0	GECOS-II (GEIN)	No activity defined. Will occur only in the last activity of a job
J1		$ PRODUCT card does not contain a program name
J2		$ DATA card does not contain a file code
J3		$ INCODE request improper
J4		$ SYSOUT card does not contain a file code
J5		Invalid option of a $ system call card ($ GMAP, $ IDS, etc.)
J6		Invalid $ file card
J7		Maximum number of files exceeded
J8		Invalid $ NTAPE card
J9		Input read error
JA		Job has exceeded Input Media (GEIN) storage size or: Job has been aborted through no fault of its own so that links may be released and system hangup prevented. Resubmit job.
JB		Checksum error on binary card
JC		More than one IDS file per activity
JD		$ PERM name not found in catalog
JE		Check character alert on system storage file
JF		Illegal device type found in catalog
JG		Catalog logical device designator does not match that of configuration table
JH		System card reader inoperable (card jam, etc.)
JI		9SA not configured
JJ		A $ PPT card encountered in input deck; $ PPTR or $ PPTP must be used
JK		Job has zero storage request on $ LIMITS card
JL		A $ REMOTE card has been encountered by GEIN but the DATANET-30 processor is not configured
JM		Invalid data on permanent file
JN		A $ SELECT card has been used within a Select activity. "Nesting" of $ SELECT cards is illegal
K1	GECOS-II (IOS)	Improper device code in PAT or CONFIG table
K2		Invalid status return point for I/O
K3		Invalid file pointer for I/O
K4		Invalid DCW pointer for I/O
K5		Invalid courtesy call address
K6		Invalid Op code, linked file I/O
K7		Two successive TDCW's
L1	GELOAD	Missing subroutine required
L2		NOGO option exercised or Program and Loader overlap
L3		Fatal error encountered during loading (fatal loading errors are listed in GE-625/635 General Loader, CPB-1008)。
M0	GECOS-II (CALL)	Parameter A in GEMORE calling sequence is not 0, 1, 2, or 3
M1		Illegal device type in GECALL/GERSTR
M2		Entry not present in GECALL/GERSTR disc catalog
M3		Entry not present in GECALL/GERSTR tape catalog
M4		Memory limit exceeded in GECALL/GERSTR
M5		File code missing in PAT for GECALL/GERSTR
M6		Input error in reading within GECALL/GERSTR

Code	Source	Explanation
M7	GECOS-II (MORE)	Parameter A in the GEMORE calling sequence specifies additional links but the Slave PAT specifies a nondisc or nondrum device or: GEMORE request for additional links results in a total exceeding 638 links
M8	PRODUCT	Production Library file has not been configured at Startup
M9	GECOS-II (SYOT/DSYT)	Calling sequence to GESYOT destroyed between the MME GESYOT and processing of the MME
MA	GECOS-II (RELS)	Calling sequence to MME GERELS contains (1) zero files to be released, (2) 20 or more files to be released, (3) program number ≥ 73_8 or (4) file word contains characters other than two low-order characters in slave mode.
N1		Fault Tag used
N2	GECOS-II (FILS)	File code not in Peripheral Assignment Table (PAT) for MME GEFILS
N3	GECOS-II (CHEK/ROLL)	Illegal action during rollback
N4	GECOS-II (SAVE)	Request for GESAVE exceeds memory limits or is zero
N5		Illegal device or no PAT for GESAVE
N6		Catalog full in GESAVE
N7		I/O error in GESAVE
N8		Insufficient number of links for GESAVE
N9		Too many DCW's required for GESAVE
P0	GMAP	Maximum of 63 levels of MACRO expansion exceeded
Q1	FORTRAN	Logical Unit Table overflow
Q2		Missing Logical Unit Table
Q3		No space for Logical Unit 6 Buffer
Q4		Machine error or unexpected error to FORTRAN Compiler
R1	ALGOL	Fatal ALGOL error. An error number preceding the R1 designation categorizes the error (See GE-625/635 ALGOL, CPB-1087)
R2	GERTS-II	Unsuccessful attempt to write data to the DATANET-30 in the direct access mode - MME GEROUT.
R3		Remote operator abort (GERTS)
R4		GERTS too busy to accept any new requests; resubmit job.
R5		Illegal op code for MME GEROUT.
R6		Illegal file code in MME GEROUT.
R7		Invalid media code for op code 1 - MME GEROUT.
R8		Invalid media code for op code 7 - MME GEROUT.
R9		Unsuccessful attempt to write data to disc - op code 1 or 7 - MME GEROUT.
RA		Excessive word count in direct access data block
RB		Excessive character count in direct access data block
RC		GERTS is not on the system; DATANET-30 is not configured
RD		Excessive word count for data in op code 1 or 7 of MME GEROUT

Code	Source	Explanation
RE	GERTS-II	Excessive output on remote file
RF		Illegal address in MME GEROUT calling sequence or associated data block
RG		Attempt to write to an unopened file - MME GEROUT
SC	SORT	SORT abort; reason indicated on execution report
SE	System Editor	System Editor has encountered an irrecoverable error
SM	COBOL	COBOL Linkage not in stack
U1	UTILITY (See CPB-1422)	Control deck error
U2		Comparison error
U3		Hardware error
V1	JOVIAL	Compiler error abort. The JOVIAL statement is given in a corresponding error message on the listing.
V2	JOVIAL	Object program abort. The JOVIAL Compiler generates a MME GEBORT when replacing an erroneous JOVIAL statement
X1	Operator-initiated abort	Operator deleted job from control stack
X2		Operator aborted job in execution

APPENDIX C. MAJOR STATUS AND SUBSTATUS CODES

The Status Return Pointer field of the Return Word (see Chapter 11) indicates to the programmer the address of the two status return words which carry back information supplied by IOS on the termination of the current I/O request. The first of these status return words contains a Major Status field (bits 2-5) and a Substatus field (bits 6-11), which serve to identify the nature of the termination of the request.

Those status returns listed with an * in this Appendix are the only ones that a user activity can expect to see in the status return words, unless a Request Status (REQS) or a Reset Status (RESS) command is issued. If the activity issues an REQS or an RESS command, it must be prepared to receive any of the status/substatus conditions listed herein. No error recovery action is performed when either an REQS or an RESS is issued.

In most cases when error recovery action is taken, appropriate error statistics are typed and written on SYSOUT for later processing and statistical analysis.

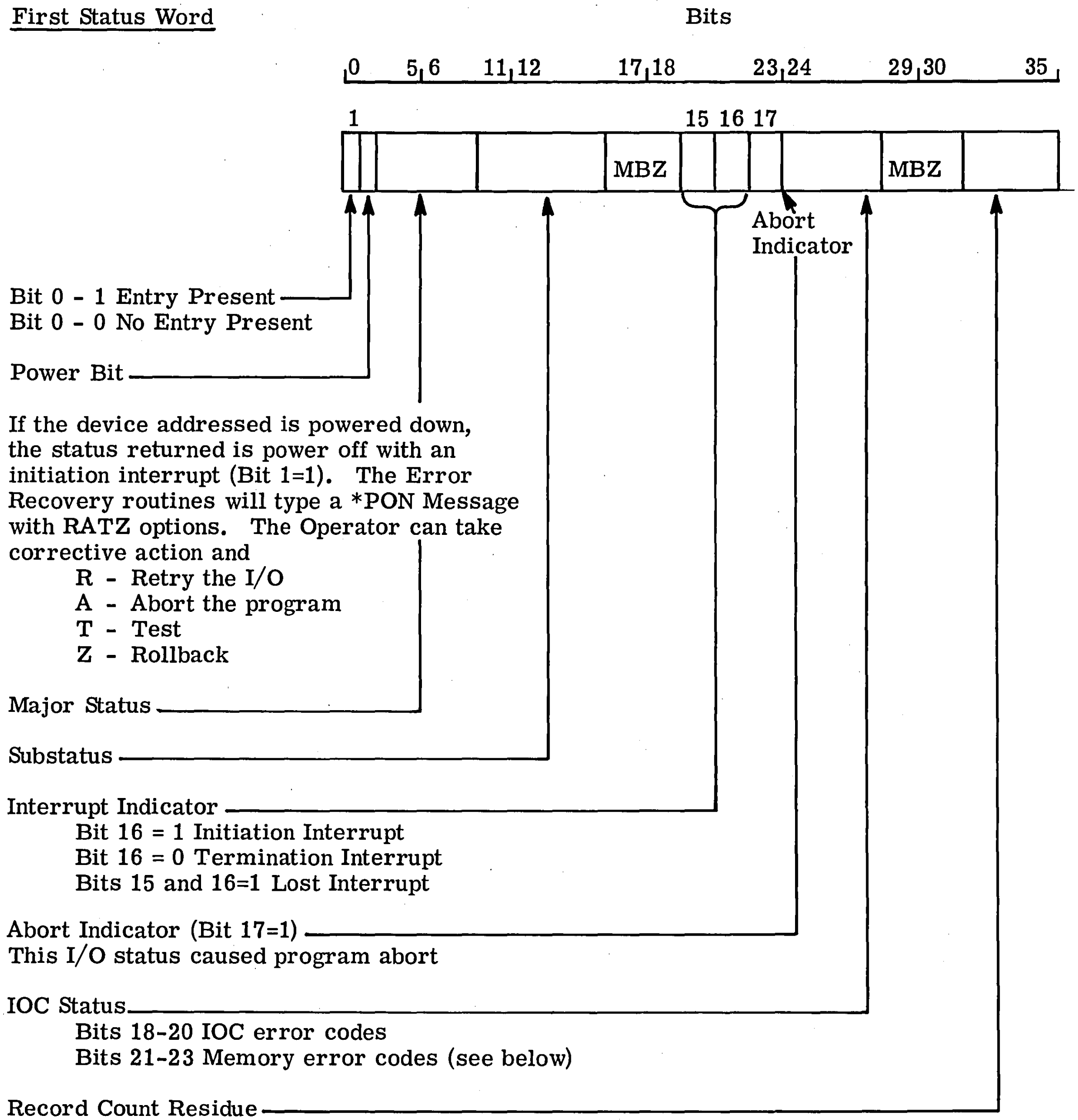
Throughout this Appendix, wherever it states under "Action Taken by GEPR" that the operator is given the option of aborting the activity, this implies that he will also take any applicable action necessary to initiate servicing of the malfunctioning hardware. Note also that the operator option, "Call in a hardware test routine," is not implemented at present.

FORMAT OF STATUS RETURN WORDS

The following describes the format of the first and second status return words for all peripherals and gives explanations and recovery procedures for power-off status and IOC error status.

FORMAT OF STATUS RETURN WORDS

First Status Word



Bit 0 - 1 Entry Present
Bit 0 - 0 No Entry Present

Power Bit

If the device addressed is powered down, the status returned is power off with an initiation interrupt (Bit 1=1). The Error Recovery routines will type a *PON Message with RATZ options. The Operator can take corrective action and

- R - Retry the I/O
- A - Abort the program
- T - Test
- Z - Rollback

Major Status

Substatus

Interrupt Indicator
Bit 16 = 1 Initiation Interrupt
Bit 16 = 0 Termination Interrupt
Bits 15 and 16=1 Lost Interrupt

Abort Indicator (Bit 17=1)
This I/O status caused program abort

IOC Status
Bits 18-20 IOC error codes
Bits 21-23 Memory error codes (see below)

Record Count Residue

IOC Error and Memory Codes. One of the following IOC error codes may appear in Bits 18-20 of the first status return word:

Bit			
18	19	20	Explanation
0	0	0	No Error
0	0	1	Primary mailbox parity error
0	1	0	Secondary mailbox parity error
1	0	0	Peripheral data parity error
1	0	1	Data error
1	1	0	Overflow address (exceeds 2^{18} - 1)
1	1	1	Illegal address

One of the following Memory codes may appear in Bits 21-23 of the first status return word:

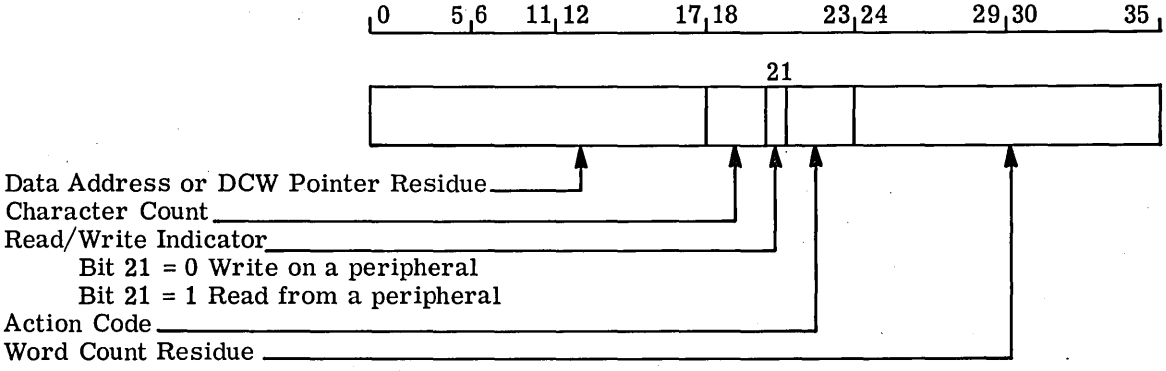
Bit			
21	22	23	Explanation
0	0	0	No error
0	0	1	Nonexistent address
0	1	0	Not master memory for this IOC
0	1	1	Parity error in memory
1	0	0	Not the control processor for this IOC
1	1	0	Protection area
1	1	1	Cycle not properly completed

Note: An unrecoverable IOC error can be a programmer error.

GECOS checks the legality of the pointer within the select sequence but the IOC checks the addresses within the DCW. For example, the programmer can designate an address in a DCW that is outside the limits of his program and this will cause an IOC error status. * GECOS error recovery (GEPR) will retry the I/O and give the operator options for further action.

Second Status Word

Bits


0
5
6
11
12
17
18
23
24
29
30
35
21
Data Address or DCW Pointer Residue
Character Count
Read/Write Indicator
Bit 21 = 0 Write on a peripheral
Bit 21 = 1 Read from a peripheral
Action Code
Word Count Residue

The Word Count is zero if terminate occurred because the Word Count was exhausted. If terminate occurred before the Word Count was exhausted:

On a Read. The Word Count residue is the number of words that were not transmitted.

On a Write. The Word Count residue will have been decremented n (where n is some small number depending upon whether IOC-B or IOC-C) more than the number not transmitted due to IOC look ahead.

*A cause of this error could be an unintended DCW count of zero. A count of zero in a DCW is interpreted as 4096_{10} by the IOC; the DCW count is added to the data address by the IOC for the limits check.

The Data Address is the location of the first word to be transmitted under the current DCW. If a terminate has occurred because the Word Count is exhausted, the Data Address field contains the location +1 of the last word transmitted. If terminate has occurred but word count is not exhausted:

On a Read. The residual data address is the location +1 of the last word or partial word transmitted.

On a Write. Same as Read, except location +n (where n = a small number and depends upon whether IOC-B or IOC-C is present.

Action Codes. One of the following Action Codes may appear in Bits 22 and 23 of the second status return word:

Bit			
22	23	Pseudo Op	Explanation
0	0	IOTD	Transmit and Disconnect. Move the number of words specified by the count to or from memory and then terminate I/O command.
0	1	IOTP	Transmit and Proceed. Move the number of words specified by the count to or from memory, then continue to the next DCW and do as it indicates.
1	0	TDCW	Transfer to DCW. Do not move any data. Go to the DCW at the address specified in this DCW.
1	1	IONTP	Nontransmit and Proceed. Skip the number of words specified in the count of this DCW and do as it indicates.

STATUS CODES

The status codes associated with each peripheral are summarized in Figure 27 and discussed in detail in the following.

	Status Codes		
Peripheral	Major	Substatus	Explanation
Card Reader (see also p. 213)	0000	000000	Channel Ready*[a]
	0010		Attention
		XXXXX1	Hopper/Stacker Alert
		XXXX1X	Manual Halt

[a] Only these status returns (indicated by an *) will normally be seen by the user unless REQS or RESS is issued.

Figure 27. Summary of Peripheral Major and Substatus Codes

	Status Codes		
Peripheral	Major	Substatus	Explanation
Card Reader (cont'd.)		XXX1XX	Last Batch
(see also p. 213)		XXX1X1	Last Batch[b]
		0X1XXX	Feed Alert
		X1XXXX	Card Jam
		1X0XXX	Read Alert
		1X1XXX	Sneak Feed*
	0011		Data Alert
		000001	Transfer Timing
		000X10	Validity Check*[a]
		0001X0	Dual Read Failure[b]
		001000	No Read Command[b]
	0100	000000	End of File*
	0101	000000	Command Reject
		000001	Invalid Command[b]
		000010	No Card Committed to Stacker[b]
		000100	Late Read Command[b]
Card Punch	0000	000000	Channel Ready*
(see also p. 220)			
	0010		Attention
		0XXXX1	Hopper/Stacker (or Pocket for CP-21) Alert
		0XXX1X	Manual Halt
		0XX1XX	Chadbox Full
		0X1XXX	Feed Alert
		01XXXX	Card Jam
	0011		Data Alert
		000000 or 000XX1	Transfer Timing
		000X1X	Transmission Parity Alert
		0001XX	Punch Alert
	0101	000000	Command Reject
Perforated Tape	0000		Channel Ready
Reader/Punch		XXXXXX	Reader Plugboard No.
(see also p. 224)			
	0010		Attention*[a]
		0000X1	Out of Tape
		00001X	Manual Halt*
		00011X	Missing Sprocket*
	0011		Data Alert
		00000X	Transfer Timing Alert*
		00XX10	Parity Alert*
		00X1X0	Punch Tape Warning*
		001XX0	Invalid Character Count*

[a] Only these status returns (indicated by an *) will normally be seen by the user, unless REQS or RESS is issued.

[b] Applicable only to CR-21.

Figure 27. Summary of Peripheral Major and Substatus Codes (Cont'd.)

	Status Codes		
Peripheral	Major	Substatus	Explanation
Perforated Tape Reader/Punch (cont'd)	0100	000000	End of File*
(see also p. 224)	0101		Command Reject
(see also p. 228)		0000X1	Invalid Operation Code
		00001X	Invalid Device Code
Printer (see also p. 228)	0000	000000	Channel Ready*[a]
	0010		Attention
		00XXX1	Out of Paper*
		00XX1X	Manual Halt*
		00X1XX	VFU Tape Alert*
		001XXX	Check*
	0011		Data Alert
		XXX01X	Alert Before Printing Began*
		000011	Transfer Timing Alert*
		XXX100	Alert After Printing Began*
		0X1XXX	Paper Low
		X1XXXX	Slew Alert
		1XXXXX	Top of Page*
	0101		
		XX0001	Invalid Operation Code
		X1000X	Slew Alert on Last Slew Operation*
		1X000X	Top-of-Page Echo on Last Slew
Magnetic Tape (see also p. 233)	0000	000000	Channel Ready*
		000X1	Last Tape Handler Write Inhibited*
		0001X	Last Tape Handler on Load Pt*
		0001XX	ASA 9-Track Tape Handler*
	0001	000000	Device Busy
	0010		Attention
		0XXXX1	Tape Write Inhibited
		0XXX1X	No Such MTU
		0XX1XX	MTU Standby
		0X1XXX	MTU Check
		01XXXX	Blank Tape on Write

[a] Only these status returns (indicated by an *) will normally be seen by the user, unless REQS or RESS is issued.

[b] Applicable only to CR-21.

Figure 27. Summary of Peripheral Major and Substatus Codes (Cont'd.)

	Status Codes		
Peripheral	Major	Substatus	Explanation
Magnetic Tape (cont'd) (see also p. 233)	0011		Data Alert
		000001	Transfer Timing Alert
		XXXX10	Blank Tape on Read*[a]
		X00011	Bit Detected During Erase
		XXX1X0	Transmission Parity Alert
		XX1XX0	Lateral Parity Alert*
		X1XXX0	Longitudinal Parity Alert*
		1XXXX0	End Tape Mark*
	0100		End of File
		XXXXXX	Single Data Character ≠ 77_8*
		111111	Data Alert = 77_8*
	0101		Command Reject
		00X0X1	Invalid Operation Code
		XXX01X	Invalid Device Code
		00X1XX	Parity Alert on Operation/Device
		X01XXX	Magnetic Tape at Load Point*
		X1XXX0	Attempted Read after Write
		1XXXX0	Invalid Command to this Unit
Magnetic Drum (see also p. 242)	0000		Channel Ready
		0XXXXX	Block Address*
	0010	000000	Attention
	0011		Data Alert
		000001	Transfer Timing Alert
		0X0X10	Transmission Parity Alert
		0X01X0	Invalid Select Data
		010XX0	Check Character Alert*
	0100		End-of-File*
	1111	000000	End-of-Logical File*

[a] Only these status returns (indicated by an *) will normally be seen by the user, unless REQS or RESS is issued.

[b] Applicable only to CR-21.

Figure 27. Summary of Peripheral Major and Substatus Codes (Cont'd.)

	Status Codes		
Peripheral	Major	Substatus	Explanation
Magnetic Drum (cont'd) (see also p. 242)	0101		Command Reject
		00XXX1	Invalid Operation Code
		00XX1X	Invalid Device Code
		00X1XX	Parity Alert on Operation/Device
		001XXX	Invalid Command Sequence
Disc Storage Unit (see also p. 246)	0000	000000	Channel Ready*[a]
	0001		Device Busy
		000000	Overridable
		000001	Not Overridable
	0010	000000	Attention
	0011		Data Alert
		000001	Transfer Timing Alert
		XXXX10	Transmission Parity Alert
		0XX1X0	Invalid Control Character
		XX1XX0	Internal Alert
		X1XXX0	Check Character Alert*
		1XXXX0	(Not Implemented)
		11XXX0	(Not Implemented)
	0100	000000	End-of-File*
	1111	000000	End-of-Logical File*
	0101		Command Reject
		0000X1	Invalid Operation Code
		00001X	Invalid Device Code
Console Typewriter (see also p. 251)	0000	000000	Channel Ready*
	0010	000000	Attention
	0011		Data Alert
		000001	Transfer Timing Alert
		0XXX10	Transmission Parity Alert
		0XX1X0	Operator Input Alert
		0X1XX0	Operator Distracted*
		01XXX0	Incorrect Format*
		100000	Message Length Alert
	0101		Command Reject
		000001	Invalid Operation/Device Code
		000011	Parity Alert on Operation Code

[a] Only these status returns (indicated by an *) will normally be seen by the user, unless REQS or RESS is issued.

[b] Applicable only to CR-21.

Figure 27. Summary of Peripheral Major and Substatus Codes (Cont'd.)

CARD READER

Major Status: *Channel Ready (0000) Substatus: None (000000)

Meaning:

If this status is received as an initiation interrupt in response to REQS or RESS, the card reader is ready to accept a new command.

If this status is received as a termination interrupt, it indicates the command execution was error free and the card reader is ready to accept a new command.

Action taken by GEPR:

On Initiation and Termination: Status returns are passed back to user program.

Major Status: Attention (0010) Substatus: Hopper/Stacker Alert (XXXXX1)

Meaning:

1. Input card hopper is empty, or
2. Output card stacker is full, or
3. Both (1) and (2) exist.

Action taken by GEPR:

On Initiation: A message is typed to the operator instructing him to place more cards in the input hopper or remove cards from the output stacker. When the attention condition has been corrected and the card reader is in "ready," the previously rejected command will be reissued.

On Termination: The first word of the card image (columns 1-6) is scanned in memory. If this word contains the Standard End-of-File card format (***EOF in columns 1-6), an End-of-File major status (0100) is returned to the user. If word in memory is not of this format, a Channel Ready major status (0000) is returned. If the last card read was not an End-of-File card and the program gives another card read command, this will result in an initiation interrupt. The operator will then be able to either refill the input hopper with the remainder of the input deck or place the End-of-File card, which he previously forgot, in the input hopper resulting in an End-of-File major status being returned to the user.

Major Status: Attention (0010) Substatus: Manual Halt (XXXX1X)

Meaning:

The operator has depressed the HALT switch. He must depress the OPERATE/RESET to resume normal operations.

CARD READER

Action taken by GEPR:

On Initiation: The initiation of the command will be deferred. It will be executed when a special interrupt is received as a result of the operator depressing the OPERATE/RESET switch.

Major Status: Attention (0010)   Substatus: Last Batch (XXX1XX)
Last Batch, CR-21 (XXX1X1)

Meaning:

This substatus will never be seen by the user program when GEPR is used. It arises from one of the following conditions:

1. The input card hopper has gone empty while the LAST BATCH switch was on, or

2. While the HOPPER/STACKER ALERT switch was on, the operator depressed the LAST BATCH switch and a REQS/RESS command was issued.

Action taken by GEPR:

On Initiation and Termination: This substatus is treated exactly as if a Hopper/Stacker Alert substatus had been received.

Major Status: Attention (0010)   Substatus: Feed Alert (0X1XXX)

Meaning:

A new card failed to feed from the input hopper within the prescribed time.

Action taken by GEPR:

On Termination: No manual backspacing is required. A message is typed to the operator giving him the following options:

1. Take corrective action and then retry the command, or

2. Call in a hardware test routine, or

3. Abort the program, or

4. Initiate a rollback procedure.

Major Status: Attention (0010)   Substatus: Card Jam (X1XXXX)

Meaning:

A card's trailing edge did not arrive at a photocell within the prescribed time period after its leading edge had passed this photocell. (After about 2 seconds following detection of this condition, the Card Reader will begin to cycle down and power will be removed from the device.)

Action taken by GEPR:

On Initiation: No manual backspacing is required. A message is typed to the operator giving him the following options:

1. Take corrective action and then retry the command, or
2. Call in a hardware test routine, or
3. Abort the program, or
4. Initiate a rollback procedure.

On Termination: Manual backspacing is required. A message is typed to the operator giving him the following options:

1. Take corrective action, manually backspace the device, and then retry the command, or
2. Call in a hardware test routine, or
3. Abort the program, or
4. Initiate a rollback procedure.

Major Status: Attention (0010) Substatus: Read Alert (1X0XXX)

Meaning:

The actual time required for the card to pass through the read station was not within the acceptable time range.

Action taken by GEPR:

On Termination: No manual backspacing is required. A message is typed to the operator giving him the following options:

1. Take corrective action, manually backspace the device, and then retry the command, or
2. Call in a hardware test routine, or
3. Abort the program, or
4. Initiate a rollback procedure.

CARD READER

Major Status: *Attention (0010) Substatus: Sneak Feed (1X1XXX)

Meaning:

Previous to receipt of the current command, one or more cards have passed through the reader without a command being given.

Action taken by GEPR:

On Initiation: The status is returned to the user program.

Major Status: Data Alert (0011) Substatus: Transfer Timing Alert (000001)

Meaning:

The IOC did not receive (read) data characters at a rate compatible with the transfer rate of the card reader subsystem.

Action taken by GEEPR:

On Termination: Manual backspacing is required. A message is typed to the operator giving him the following options:

1. Take corrective action, manually backspace the device, and then retry the command, or
2. Call in a hardware test routine, or
3. Abort the program, or
4. Initiate a rollback procedure.

Major Status: *Data Alert (0011) Substatus: Validity Check (000X10)

Meaning:

During a Read Card Hollerith command, a character was recognized as not being a valid character (GE-625/635 Character Set). An ignore character (octal 17) is automatically stored in the memory position of the invalid character. The remainder of the card is read and stored normally.

Action taken by GEPR:

On Termination: A message is typed to the operator giving him the following options:

1. Take corrective action, manually backspace the device, and then retry the command, or
2. Return status to user program.

Major Status: Data Alert (0011) Substatus: Dual Read Failure, CR-21 (0001X0)

Meaning:

This condition occurs when the two outputs of the dual read head (CR-21) do not compare for a card column. In the binary mode, the detection of this error will cause two ignore characters (17_8) to be stored in memory where card column information is stored. One ignore character is stored in the Hollerith mode when this error is detected. In either mode of operation, the rest of the card is read and stored normally.

Action taken by GEPR:

On Termination: Manual backspacing is required. A message is typed to the operator giving him the following options:

1. Take corrective action, manually backspace the last card read and then retry the command, or
2. Call in a hardware test routine, or
3. Initiate a rollback procedure, or
4. Abort the program.

Major Status: Data Alert (0011) Substatus: No Read Command CR-21 (001000)

Meaning:

A card fed by a Feed command has entered the Read station before a Read command was received.

(The Read command must be received within 9 milliseconds of the preceding Feed Command.) No data is transmitted.

Action taken by GEPR:

On Termination: Manual backspacing is required. A message is typed to the operator giving him the following options:

1. Take corrective action, manually backspace the last card read, and then retry the command, or
2. Call in a hardware test routine, or
3. Initiate a rollback procedure, or
4. Abort the program.

CARD READER

Major Status: *End of File (0100) Substatus: None (000000)

Meaning:

The input hopper has gone empty and the last card read was End-of-File card (***EOF).

Action taken by GEPR:

On Termination: The End-of-File major status (0100) is passed back to user.

Major Status: Command Reject (0101) Substatus: None (000000)

Meaning:

The received command is being rejected because it is of an unacceptable configuration or has a parity error.

Action taken by GEPR:

On Initiation: The rejected command will be reissued N times by GEPR. N, which is easily changed by an EQU card, will be 3. If recovery is not successful, the operator is given the following options:

1. Reissue N more times, or
2. Call for a hardware test routine, or
3. Initiate a rollback procedure, or
4. Abort the program.

Major Status: Command Reject (0101) Substatus: Invalid Command, CR-21 (000001)

Meaning:

This condition occurs when the operation code is not recognized for any reason, an invalid bit configuration or a parity error.

Action taken by GEPR:

On Initiation: The requested command is reissued N times by GEPR. N, which is easily changed by an EQU card, will be 3. If recovery is not successful, the operator is given the following options:

1. Reissue N times, or
2. Call for a hardware test routine, or
3. Initiate a rollback procedure, or
4. Abort the program.

Major Status: Command Reject (0101) Substatus: No Card Committed to Stacker, CR-21 (000010)

Meaning:

The condition occurs when a Stack command is received and no card is in transit to the stacker gate. (A Stack command must be received within 6 milliseconds following the Terminate signal as that card left the Read head.)

Action taken by GEPR:

Either of two situations may have occurred when this status return is given. (1) A card may have been read and the Stacker command was not received in time or (2) a card was not read but a Stacker command was inadvertently issued by the user. At present,* the action taken by GEPR is not correct; either condition will result in GEPR reissuing the rejected command N times even though the second condition is a program error. N, which is easily changed by an EQU card, will be 3. If recovery is not successful, the operator is given the following options:

1. Reissue N times, or
2. Call for a hardware test routine, or
3. Initiate a rollback procedure, or
4. Abort the program.

Major Status: Command Reject (0101) Substatus: Late Read Command, CR-21 (000100)

Meaning:

A Read command was received after a card has entered the read head. If the movement of the card was initiated by a Feed Card Command, see Data Alert, No Read Command, (001000).

*Change Letter 9

CARD PUNCH
CARD READER

Action taken by GEPR:

On Termination: At present*, the action taken by GEPR is not correct for the type of error this status indicates. The card should be manually placed in the input hopper for re-reading. GEPR now reissues the rejected command N times. Because this error resulted from a Late Read command, the rejected command is a Read command. In all likelihood when, and if, the command is executed correctly, from 1 to N cards will have been passed by the card reader without having been read to core. N is defined as 3 by an EQU card. If recovery is not successful, the operator is given the following options:

1. Reissue N more times, or
2. Call for a hardware test routine, or
3. Initiate a rollback procedure, or
4. Abort the program.

CARD PUNCH

Major Status: *Channel Ready (0000) Substatus: None (000000)

Meaning:

If this status is received as an initiation interrupt in response to REQS or RESS, the card punch is ready to accept a new command.

If this status is received as a termination interrupt, it indicates the command execution was error free and the card punch is ready to accept a new command.

Action taken by GEPR:

On Initiation and Termination: Status Returns are passed back to user program.

Major Status: Attention (0010) Substatus: Hopper/Stacker (or Pocket, for CR-21) Alert (0XXXX1)

Meaning:

The input hopper is empty; the output stacker is full; or both. For the CR-21, a card has been sent to the wrong output stacker.

*Change Letter 9

Action taken by GEPR:

On Initiation: No manual backspacing is required. A message is typed to the operator informing him of the condition. The interrupted command will be reissued as soon as the attention condition is cleared.

On Termination: This condition is ignored and GEPR returns a Channel Ready status to the user because this I/O operation was completed normally. This attention condition will be detected as an initiation interrupt on the next I/O request to this device.

Major Status: Attention (0010) Substatus: Manual Halt (0XXX1X)

Meaning:

The operator has depressed the HALT switch. He must depress the OPERATE/RESET to resume normal operations.

Action taken by GEPR:

On Initiation: GEPR will resume normal operation when the attention condition is cleared.

On Termination: This condition is ignored and GEPR returns a Channel Ready status to the user because this I/O operation was completed normally. This attention condition will be detected as an initiation interrupt on the next I/O request to this device.

Major Status: Attention (0010) Substatus: Chadbox Full (0XX1XX)

Meaning:

Chad is the waste material resulting from a card being punched. When the receptacle for these pieces of chad is full, this status is returned, and the operator must empty the chadbox before normal punching can be resumed.

Action taken by GEPR:

On Initiation: No manual backspacing is required. A message is typed to the operator informing him of the condition. The interrupted command will be reissued as soon as the attention condition is cleared.

On Termination: This condition is ignored and GEPR returns a Channel Ready status to the user because this I/O operation was completed normally. This attention condition will be detected as an initiation interrupt on the next I/O request to this device.

CARD PUNCH

Major Status: Attention (0010)   Substatus: Feed Alert (0X1XXX)

Meaning:

A card has failed to feed from the input hopper into the card punch mechanism. The card currently being punched is completed normally.

Action taken by GEPR:

On Initiation: No manual backspacing is required. A message is typed to the operator giving him the following options:

1. Take corrective action and then retry the command, or
2. Call in a hardware test routine, or
3. Abort the program, or
4. Initiate a rollback procedure.

On Termination: This condition is ignored and GEPR returns a Channel Ready status to the user because this I/O operation was completed normally. This attention condition will be detected as an initiation interrupt on the next I/O request to this device.

Major Status: Attention (0010)   Substatus: Card Jam (01XXXX)

Meaning:

One or more cards were improperly loaded in the input hopper or a card has failed to feed at the proper time from one station to another. The card currently being punched will immediately be terminated and will not be completed normally. (After about 2 seconds following detection of this condition, the card punch will begin to cycle down and power will be removed from this peripheral.)

Action taken by GEPR:

On Initiation: No manual backspacing is required. A message is typed to the operator giving him the following options:

1. Take corrective action and then retry the command, or
2. Call in a hardware test routine, or
3. Abort the program, or
4. Initiate a rollback procedure.

On Termination: Manual backspacing is required. A message is typed to the operator giving him the following options:

1. Take corrective action, manually backspace the device, and then retry the command, or
2. Call in a hardware test routine, or
3. Abort the program, or
4. Initiate a rollback procedure.

Major Status: Data Alert (0011) Substatus: Transfer Timing Alert (000000) or (000XX1)

Meaning:

The IOC did not send (write) data characters at a rate compatible with the transfer rate of the Card Punch Subsystem.

Action taken by GEPR:

On Termination: Manual backspacing is required. A message is typed to the operator giving him the following options:

1. Take corrective action, manually backspace the device, and then retry the command, or
2. Call in a hardware test routine, or
3. Abort the program, or
4. Control is return to the user program.

Major Status: Data Alert (0011) Substatus: Transmission Parity Alert (000X1X)

Meaning:

Incorrect parity was detected on a data character received from the IOC. This can occur only during a write operation.

Action taken by GEPR:

On Termination: Manual backspacing is required. A message is typed to the operator giving him the following options:

1. Take corrective action, manually backspace the device, and then retry the command, or
2. Call in a hardware test routine, or
3. Initiate a rollback procedure, or
4. Abort the program.

Major Status: *Data Alert (0011) Substatus: Punch Alert (0001XX)

Meaning:

A hole count of what is actually punched on the card is compared with the calculated hole count. This status indicates the two counts do not agree.

TAPE READER/PUNCH
CARD PUNCH

NOTE: Since this check is made as the card passes from the first post-punch station to the second post-punch station, a subsequent card will be undergoing punching at the time this error is detected.

Action taken by GEPR:

Data Alert status (0011) with punch alert substatus (0001XX) is returned to the user program. The user may use his own recovery action, such as tell the operator to discard the last two cards that were punched and then reissue the two commands, or he may choose to ignore the punch alert substatus. (See NOTE under Meaning)

Major Status: Command Reject (0101) Substatus: None (000000)

Meaning:

The received command is being rejected because it is of an unacceptable configuration or has a parity error.

Action taken by GEPR:

On Initiation: The rejected command will be reissued N times by GEPR. This N, which is easily changed by an EQU card, will be 3. If recovery is not successful, the operator is given the following options:

1. Reissue N more times, or
2. Call for a hardware test routine, or
3. Initiate a rollback procedure, or
4. Abort the program.

PERFORATED TAPE READER/PUNCH

Major Status: *Channel Ready (0000) Substatus: Reader Plugboard No. (XXXXXX)

Meaning:

When Channel Ready status is returned for any command addressed to the Perforated Tape Reader/Punch, the substatus field will contain the 6-bit number which has been wired into the Reader Plugboard. If no Reader Plugboard is present, a $(77)_8$ substatus is sent from the peripheral.

Action taken by GEPR:

On Initiation and Termination: Status returns are passed back to user program.

Major Status: *Attention (0010) Substatus: Out of Tape (0000X1)

Meaning:

Either the punch or the reader has run out of tape, or the tape has broken. This condition can also occur if the tension is too tight between the punch head and the supply reel.

Action taken by GEPR:

On Initiation and Termination: Status returns are passed back to user program.

Major Status: *Attention (0010) Substatus: Manual Halt (00001X)

Meaning: One or more of the following switches has been actuated:

1. Manual HALT switch (both punch and reader are inoperable).
2. READER TAPE LOAD switch (only the reader is inoperable).
3. PUNCH TAPE LOAD switch (only the punch is inoperable).
4. ON/OFF LINE switch set to OFF LINE.

Action taken by GEPR:

On Initiation and Termination: Status returns are passed back to user program.

Major Status: *Attention (0010) Substatus: Missing Sprocket (00011X)

Meaning:

A data hole has been detected without detecting a sprocket hole. This may be caused by chad in the sprocket hole or the tape being out of registration.

Action taken by GEPR:

On Initiation and Termination: Status returns are passed back to the user program.

TAPE READER/PUNCH

Major Status: *Data Alert (0011) Substatus: Transfer Timing Alert (00000X)

Meaning:

The IOC did not send (write) or receive (read) data characters at a rate compatible with the transfer rate of the system.

Action taken by GEPR:

On Termination: Since there is no programmed method of backspacing the perforated tape reader/punch and since any Data Alert is a termination interrupt and has resulted in the movement of perforated tape, GEPR is unable to perform any recovery for Data Alert conditions. GEPR will return the status to the user program and allow him to initiate his own recovery action.

Major Status: *Data Alert (0011) Substatus: Parity Alert (00X10)

Meaning:

A parity error has been detected on a data character being read according to the wiring of the PARITY HUB on the plugboard. The use of the PARITY HUB will indicate whether or not parity is to be checked, and whether parity is to be odd or even.

While punching, this indicates a parity error is detected on a data character received from the IOC. The character in error is not punched.

Action taken by GEPR:

On Termination: Since there is no programmed method of backspacing the perforated tape reader/punch and since any Data Alert is a termination interrupt and has resulted in the movement of perforated tape, GEPR is unable to perform any recovery for Data Alert conditions. GEPR will return the status to the user program and allow him to initiate his own recovery action.

Major Status: *Data Alert (0011) Substatus: Punch Tape Warning (00X1X0)

Meaning:

The take-up reel on the punch is full, or the input reel on the punch has about N feet of unused tape remaining. N is variable depending upon the adjustment of a sensing arm. The punch tape trailer length (N) should be standardized depending upon the maximum record length at an installation, but should never be less than 25 feet.

Action taken by GEPR:

On Termination: GEPR will return the status to the user program and allow him to initiate his own recovery action. The user must be prepared to end the tape and to inform the operator of appropriate action to be taken.

Major Status: *Data Alert (0011) Substatus: Invalid Character Count (001XX0)

Meaning:

When reading in the double character mode, an end data transfer is received by the Perforated Tape Subsystem after the first character has been accepted by the external user. The second character is stored, the tape is halted, and the Read command is Terminated with the status of Data Alert, invalid character count.

Action taken by GEPR:

On Termination: Since there is no programmed method of backspacing the perforated tape reader/punch and since any Data Alert is a termination interrupt which has resulted in the movement of perforated tape, GEPR is unable to perform any recovery for Data Alert conditions. GEPR will return the status to the user program and allow him to initiate his own recovery action.

Major Status: *End of File (0100) Substatus: None (000000)

Meaning:

A character wired into the END OF FILE HUB on the reader plugboard has been detected. This character will be sent to the IOC unless it has also been wired into the DELETE HUB.

Action taken by GEPR:

On Termination: Status Returns are passed back to user program.

Major Status: Command Reject (0101) Substatus: Invalid Operation Code (0000X1)

Meaning:

Either the operation code is unrecognizable or a parity error exists on the operation code.

Action taken by GEPR:

On Initiation: The rejected command will be reissued N times by GEPR. N, which is easily changed by an EQU card, will be 3. If recovery is not successful, the operator is given the following options:

1. Reissue N more times, or
2. Call for a hardware test routine, or
3. Initiate a rollback procedure, or
4. Abort the program.

PRINTER
TAPE READER/PUNCH

Major Status: Command Reject (0101) Substatus: Invalid Device Code (00001X)

Meaning:

Either the device code is unrecognizable (refers to the punch while the physical system contains only a reader) or there is a parity error detected on an otherwise legitimate device code.

Action taken by GEPR:

On Initiation: The rejected command will be reissued N times by GEPR. N, which is easily changed by an EQU card, will be 3. If recovery is not successful, the operator is given the following options:

1. Reissue N more times, or
2. Call for a hardware test routine, or
3. Initiate a rollback procedure, or
4. Abort the program.

PRINTER

Major Status: *Channel Ready (0000) Substatus: None (000000)

Meaning:

If this status is received as an initiation interrupt in response to an REQS or RESS, the printer is ready to accept a new command.

If this status is received as a termination interrupt, it indicates the command execution was error free and the printer is ready to accept a new command.

Action taken by GEPR:

On Initiation and Termination: Status returns are passed back to user program.

Major Status: *Attention (0010) Substatus: Out of Paper (00XXX1)

Meaning:

This status may be caused by either the detection of torn paper or the first top of page after the paper low condition is detected.

Action taken by GEPR:

On Initiation: Status returns are passed back to user program.

On Termination: This status is always changed to an initiation interrupt and is returned to the user as such. Detection of this status as a termination interrupt always results in aborting the command before anything is printed.

Major Status: *Attention (0010)   Substatus: Manual Halt (00XX1X)

Meaning:

Either of the following has occurred:

1. The manual HALT switch has been actuated, or
2. The POWER ON has been actuated while the printer was not in the Power On state, or
3. The printer yoke has been raised from the print hammers.

This condition is reset manually by actuating the "OPERATE/RESET" switch.

Action taken by GEPR:

On Initiation and Termination: Status returns are passed back to user program.

Major Status: *Attention (0010)   Substatus: VFU Tape Alert (00X1XX)

Meaning:

One of the following has occurred:

1. VFU tape horizontal parity error, or
2. VFU tape is not present, or
3. VFU tape is not properly installed, or
4. Holes are punched in both channel 5 (start automatic slew) and channel 6 (stop automatic slew) of the same vertical line position on the VFU tape.

Action taken by GEPR:

On Termination: Status Returns are passed back to user program.

Major Status: *Attention (0010)   Substatus: Check (001XXX)

Meaning:

One or more of the following has occurred:

1. Hammer driver fuse failures, or
2. Paper slew fuse failure, or
3. Incomplete printout detected because not all characters received from IOC have been printed, or
4. Print wheel out of sequence.

PRINTER

Action taken by GEPR:

On Initiation and Termination: Status returns are passed back to user program.

Major Status: *Data Alert (0011) Substatus: Alert Before Printing Began (XXX01X)

Meaning:

This condition is detected when:

1. Parity error is detected on a print line data character sent from the IOC.
2. Printer buffer overflow is sensed when more than 136 printable characters (including spaces) are received from the IOC before either a slew character or an end data transfer signal is received by the printer.
3. Internal parity error is detected on a character within the internal storage buffer of printer.
4. A transfer timing alert condition exists.

Action taken by GEPR:

On Initiation: The rejected command will be reissued N times by GEPR. N, which is easily changed by an EQU card, will be 3. If recovery is not successful, the status is returned to the user program.

Major Status: *Data Alert (0011) Substatus: Transfer Timing Alert (000011)

Meaning:

The IOC did not send (write) data characters at a rate compatible with the transfer rate of the system. Since data transfer occurs before printing starts, transfer timing alert will always be sent with Alert Before Printing Began substatus.

Action taken by GEPR:

On Initiation: The rejected command will be reissued N times by GEPR. N, which is easily changed by an EQU card, will be 3. If recovery is not successful, the operator is given the following options:

1. Reissue N more times, or
2. Call for a hardware test routine, or
3. Initiate a rollback procedure, or
4. Abort the program, or
5. Return status to the user program.

Major Status: *Data Alert (0011)   Substatus: Alert After Printing Began (XXX100)

Meaning:

This substatus is sent when a parity error has been detected on a data character within the printer buffer and physical printing has actually been started.

Action taken by GEPR:

On Termination: Status returns are passed back to user program.

Major Status: Data Alert (0011)   Substatus: Paper Low (0X1XXX)

Meaning:

A program that uses GEPR need not concern itself with this return. This substatus is sent when the printer is about to run out of paper. When this condition is sensed, between 22 and 26 inches of unused paper still remain.

Action taken by GEPR:

On Termination: This condition is ignored,and GEPR returns a Channel Ready status to the user. At the first top of page, after the paper low condition is detected, a Device Attention status is detected and is passed back to the user program.

Major Status: Data Alert (0011)   Substatus: Slew Alert (X1XXXX)

Meaning:

This substatus is sent when two top of page indications (a punch in channel 7 of the VFU tape) are sensed within a single slew operation.

Action taken by GEPR:

On Termination: A message is typed providing the operator with the following options:

1. Take corrective action and then retry the command, or
2. Call for a hardware test routine, or
3. Initiate a rollback procedure, or
4. Abort the program.

PRINTER

Major Status: *Data Alert (0011) Substatus: Top of Page (1XXXXX)

Meaning:

Paper is slewed to the top of page as a result of a slew command other than an explicit slew-to-top-of-page command. This is often used by a programmer to initiate the printing of page headings.

Action taken by GEPR:

On Termination: Status returns are passed back to user program.

Major Status: Command Reject (0101) Substatus: Invalid Operation Code (XX0001)

Meaning:

Either the operation code is unrecognizable or a parity error exists on an otherwise legitimate operation code.

Action taken by GEPR:

On Initiation: The rejected command will be reissued N times by GEPR. N,which is easily changed by an EQU card, will be 3. If recovery is not successful, the operator is given the following options:

1. Reissue N more times, or
2. Call for a hardware test routine, or
3. Initiate a rollback procedure, or
4. Abort the program.

Major Status: *Command Reject (0101) Substatus: Slew Alert on Last Slew Operation (X1000X)

Meaning:

The last command resulted in a termination interrupt with Data Alert status and Slew Alert substatus.

Action taken by GEPR:

On Termination: A message is typed informing the operator of the Alert condition and providing the following options:

1. Take corrective action and then retry the command, or
2. Call for a hardware test routine, or

3. Initiate a rollback procedure, or

4. Abort the program.

Major Status: *Command Reject (0101) Substatus: Top-of-Page Echo on Last Slew Operation (1X000X)

Meaning:

The last command resulted in a termination interrupt with Data Alert status and Top-of-Page substatus.

Action taken by GEPR:

On Initiation: Status returns are passed back to user program.

MAGNETIC TAPE

Major Status: *Channel Ready (0000) Substatus: None (000000)

Meaning:

If this status is received as an initiation interrupt in response to an REQS or RESS, the magnetic tape subsystem is ready to accept a new command.

If this status is received as a termination interrupt, it indicates the command execution was error free and the magnetic tape subsystem is ready to accept a new command.

Channel Ready major status may have one or both of the following substatus conditions accompanying it.

Action taken by GEPR:

On Initiation and Termination: Status returns are passed back to user program.

Major Status: *Channel Ready (0000) Substatus: Last Tape Handler Write Inhibited (000X1)

Meaning:

The tape on the last tape unit addressed has had the "write enable" ring removed from it.

Action taken by GEPR:

On Initiation and Termination: Status returns are passed back to user program.

MAGNETIC TAPE

Major Status: *Channel Ready (0000) Substatus: Last tape Handler on Load Point (000X1X)

Meaning:

The tape on the last tape unit addressed is currently positioned at the load point. It is ready to read or write the first physical record on tape.

Action taken by GEPR:

On Initiation and Termination: Status returns are passed back to user program.

Major Status: *Channel Ready (0000) Substatus: ASA 9-Track Tape Handler (0001XX)

Meaning:

If this status is received as an initiation interrupt in response to an REQS or RESS, the magnetic tape subsystem is ready to accept a new command.

If this status is received as a termination interrupt, it indicates the command execution was error free and the magnetic tape subsystem is ready to accept a new command.

Note: For future ASA magnetic tape units, the upper three bits of the substatus indicate density.

Action taken by GEPR:

On Initiation and Termination: Status returns are passed back to the user program.

Major Status: Device Busy (0001) Substatus: None (000000)

Meaning:

This status will never be seen by a program using IOS. The addressed tape unit is rewinding or, if this is a cross-barred tape system, the addressed tape unit is executing a command from the second controller.

Action taken by GEPR:

On Initiation: IOS will reinitiate the I/O action after all other requests queued for that channel have been handled.

Major Status: Attention (0010) Substatus: Tape Write Inhibited (0XXXX1)

Meaning:

A write command has been addressed to a tape unit on which the tape reel does not have a "write enable" ring.

Action taken by GEPR:

On Initiation: No manual backspacing is required. A message is typed to the operator giving him the following options:

1. Take corrective action and then retry the command, or
2. Call in a hardware test routine, or
3. Initiate a rollback procedure, or
4. Abort the program.

Major Status: Attention (0010) Substatus: No Such MTU (0XXX1X)

Meaning:

A command has been addressed to a tape unit that does not exist within the system.

Valid logical tape unit numbers are from 0 through 15; any physical unit may be set to any unique logical tape unit number.

Action taken by GEPR:

On Initiation: No manual backspacing is required. A message is typed to the operator giving him the following options:

1. Take corrective action and then retry the command, or
2. Call in a hardware test routine, or
3. Initiate a rollback procedure, or
4. Abort the program.

Major Status: Attention (0010) Substatus: MTU Standby (0XX1XX)

Meaning:

The addressed tape unit is physically present but is not in the Ready state, it is in Standby. This can be caused by:

1. A rewind/standby command.

2. The operator has manually put the unit in Standby.
3. A previous write operation resulted in the Blank tape on Write, Device Attention being set.

Action taken by GEPR:

On Initiation: No manual backspacing is required. A message is typed to the operator giving him the following options:

1. Take corrective action and then retry the command, or
2. Call in a hardware test routine, or
3. Initiate a rollback procedure, or
4. Abort the program.

On Termination: This unrecoverable error is due to either operator error or hardware malfunction. GEPR will give the operator the option of calling in a hardware test routine* or aborting the program.

Major Status: Attention (0010) Substatus: MTU Check (0X1XXX)

Meaning:

Some malfunction (such as broken tape) within the addressed tape unit has resulted in an inoperable condition that makes further operation physically impossible.

Action taken by GEPR:

On Initiation: A message is typed to the operator giving him the following options:

1. Take corrective action and then retry the command, or
2. Call in a hardware test routine, or
3. Initiate a rollback procedure, or
4. Abort the program.

On Termination: This inoperable condition is caused by some hardware malfunction; recovery is possible, with care. GEPR will give the operator the above options.

Major Status: Attention (0010) Substatus: Blank Tape on Write (01XXXX)

Meaning:

A write operation had actually been started on the addressed handler but the read after write check was unable to detect any characters.

*At present, inoperable.

Action taken by GEPR:

On Termination: This inoperable condition is caused by some hardware malfunction, recovery is not possible. GEPR will give the operator the option of calling in a test routine or aborting the program.

Major Status: Data Alert (0011) Substatus: Transfer Timing Alert (000001)

Meaning:

The IOC did not send (write) or receive (read) data characters at a rate compatible with the transfer rate of the subsystem.

Action taken by GEPR:

On Initiation: The rejected command will be reissued N times by GEPR. N, which is easily changed by an EQU card, will be 3. If recovery is not successful, the operator is given the following options:

1. Reissue N more times, or
2. Call for a hardware test routine, or
3. Initiate a rollback procedure, or
4. Abort the program.

Major Status: *Data Alert (0011) Substatus: Blank Tape on Read (XXXX10)

Meaning:

After the receipt of a read command, 120 inches of tape has been passed over without the detection of any character previously being written on the tape.

Action taken by GEPR:

On Termination: Since this condition may be the result of either a hardware failure or of trying to read a new tape that has never been written on, this status/substatus is returned to the user program so that he may take whatever action is appropriate to his specific needs.

Major Status: Data Alert (0011) Substatus: Bit Detected During Erase (XXXX11)

Meaning:

This status is sent when a bit is detected in that portion of tape that should have been erased as the result of either an erase or a write-end-of-file command.

MAGNETIC TAPE

Action taken by GEPR:

On Termination: The Data Alert, Bit detected during erase, status/substatus is changed to a Channel Ready status and is returned to the user. If there is truly a noise bit on the tape, this will be successfully handled when the tape is read (see action for Longitudinal Parity Alert).

Major Status: Data Alert (0011) Substatus: Transmission Parity Alert (XXX1X0)

Meaning:

Incorrect parity was detected on a data character received from the IOC. This can occur only during a write operation.

Action taken by GEPR:

On Initiation: The rejected command will be reissued N times by GEPR. N, which is easily changed by an EQU card, will be 3. If recovery is not successful, the operator is given the following options:

1. Reissue N more times, or
2. Call for a hardware test routine, or
3. Initiate a rollback procedure, or
4. Abort the program.

Major Status: *Data Alert (0011) Substatus: Lateral Parity Alert (XX1XX0)

Meaning:

During a write or a read operation either a missing character is detected or lateral (character) parity is detected.

Action taken by GEPR:

See action taken for Longitudinal Parity Alert.

Major Status: *Data Alert (0011) Substatus: Longitudinal Parity Alert (X1XXX0)

Meaning:

During a write or a read operation, the recorded check character does not agree with the calculated check character.

Action taken by GEPR:

On Termination:

For Reading: When GEPR encounters a parity error while reading, the tape is backspaced and the read is tried again. If after trying the read N times (N will be 10, but may be changed by an EQU card), a parity alert is still detected, the tape is backspaced once more and the record is reread with a GEPR-created instruction having a word count of 3. If the DCW residue contains a number X(X≠0), then the record just read was noise.

The noise record is ignored and with the tape now positioned beyond the noise record, the read tape command is reissued and the following physical record is read.

If the DCW residue contains a zero then the record just read was not noise but is a valid data record with parity. The tape is backspaced and the read command reissued. If parity still exists, after repeating the cycle M times, the operator is given the option of retrying another M times (M is 3, but is easily changed by an EQU card) or of having the status/ substatus returned to the user program. The user program may now ignore the record or initiate his own recovery action.

For Writing: When GEPR encounters a parity error while writing, the tape is backspaced and the write is retried. This cycle is repeated N times (N will be 1, but may be changed by an EQU card). If still unsuccessful the tape is erased (about 9 inches) and the cycle retried N times. If still unsuccessful, the erase is repeated for a total of 10 erasures (or about 90 inches of tape) and the write is retried another N times. If parity still exists the operator is given the option of:

1. Calling a hardware test routine, or
2. Initiate a rollback procedure, or
3. Return status to user program, or
4. Abort the program.

Major Status: *Data Alert (0011) Substatus: End Tape Mark (1XXXX0)

Meaning:

During a write operation, the tape handler has detected the end-of-tape reflective foil. This write operation will proceed to its normal termination, and any subsequent write commands will be executed normally. The portion of tape beyond the end-of-tape foil should be used only for end-of-tape labels and checkpoint memory dumps.

Action taken by GEPR:

On Initiation: This status is returned to the user if, during some error recovery attempt, the End Tape Mark status is detected by GEPR. The user should understand that the Write he is attempting has, in fact, not been done.

On Termination: Status returns are passed back to user program.

MAGNETIC TAPE

Major Status: *End of File (0100) Substatus: Single Data Character ($\neq 77_8$) (XXXXXX)

Meaning of Status Return:

During the execution of a read, backspace, or forwardspace command, a single character record accompanied by its longitudinal check character was detected. If the operation was completed with no Data Alert condition, the substatus character will be the bit configuration of the single character end-of-file record for a file written in the binary mode, or an octal 17 for a file written in the decimal mode.

NOTE: The write single character record (WFM-write file mark) can be used with files written and read only in the binary mode.

An end-of-file (WEF=55) may be used with either BCD or binary files. A 7-track tape handler will write 17_8 in even parity; a 9-track tape handler will write 23_8 in odd parity. Octal 77 should not be used as an end-of-file character. And end-of-file character 00_8 is used exclusively as a checkpoint and should not be used by the user.

Action taken by GEPR:

On Termination: Status returns are passed back to user program.

Major Status: *End of File (0100) Substatus: Data Alert ($=77_8$) (111111)

Meaning:

A data alert condition was detected during the reading of an end-of-file record.

NOTE: If a programmer specifies 77 as a legitimate end-of-file character, he will have no way of knowing whether a data alert condition exists or whether this is truly his special end-of-file character for files written in the binary mode.

Action taken by GEPR:

On Termination: Status returns are passed back to user program.

Major Status: Command Reject (0101) Substatus: Invalid Operation Code (00X0X1)

Meaning:

The operation code, although having correct parity, is not of a recognizable configuration.

Action taken by GEPR:

On Initiation: This unrecoverable error is in the user program. There is no possible recovery action that GEPR can take. The program is aborted, and an error message is typed to inform the operator of the error condition and the action taken.

Major Status: Command Reject (0101) Substatus: Invalid Device Code (XXX01X)

Meaning:

The device code, although having correct parity, is not of a recognizable configuration.

Action taken by GEPR:

On Initiation: This error is due to a hardware malfunction. There is no possible recovery action that GEPR can take. The program is aborted, and an error message is typed to inform the operator of the error condition and the action taken.

Major Status: Command Reject (0101) Substatus: Parity Alert on Operation/Device Code (00X1XX)

Meaning:

A parity error is detected on either the operation code, the device code, or both.

Action taken by GEPR:

On Initiation: The rejected command will be reissued N times by GEPR. N, which is easily changed by an EQU card, will be 3. If recovery is not successful, the operator is given the following options:

1. Reissue N more times, or
2. Call for a hardware test routine, or
3. Initiate a rollback procedure, or
4. Abort the program.

Major Status: *Command Reject (0101) Substatus: Magnetic Tape on Load Point (X01XXX)

Meaning:

A backspace record or backspace file command was attempted while the tape was positioned at the load point.

Action taken by GEPR:

On Initiation: Status returns are passed back to user program.

DRUM
MAGNETIC TAPE

Major Status: Command Reject (0101) Substatus: Attempted Read after Write (X1XXX0)

Meaning:

A program issued a read or forward space instruction after a write instruction had just been issued to this tape unit with no intervening instructions.

Action taken by GEPR:

On Initiation: This unrecoverable error is in the user program. There is no possible recovery action that GEPR can take. The program is aborted and an error message is typed to inform the operator of the error condition and the action taken.

Major Status: Command Reject (0101) Substatus: Invalid Command to this Unit (1XXXX0)

Meaning:

The command was issued for an ASA 9-track magnetic tape handler and is invalid because a 7-track handler was addressed.

Action taken by GEPR:

On Initiation: This unrecoverable error is in the user program. There is no possible recovery action that GEPR can take. The program is aborted and an error message is typed to inform the operator of the error condition and the action taken.

MAGNETIC DRUM

Major Status: *Channel Ready (0000) Substatus: Block Address (0XXXXX)

Meaning:

If this status is received as an initiation interrupt in response to an REQS or RESS, the magnetic drum is ready to accept a new command.

If this status is received as a termination interrupt, it indicates the command execution was error free and the magnetic drum is ready to accept a new command.

The substatus will always contain the 5 low-order bits of the block address which is a measure of the actual angular position of the drum at the instant the command was executed.

Action taken by GEPR:

On Initiation and Termination: Status returns are passed back to user program.

Major Status: Attention (0010) Substatus: None (000000)

Meaning:

This status indicates the magnetic drum is unable to accept and instruction because of some inoperable condition.

Action taken by GEPR:

On Initiation and Termination: A message is typed to the operator giving him the following options:

1. Take corrective action and then retry the command, or
2. Call in a hardware test routine, or
3. Initiate a rollback procedure, or
4. Abort the program.

Major Status: Data Alert (0011) Substatus: Transfer Timing Alert (000001)

Meaning:

The IOC did not send (write) or receive (read) data characters at a rate compatible with the transfer rate of the subsystem.

Action taken by GEPR:

On Initiation: The rejected command will be reissued N times by GEPR. N, which is easily changed by an EQU card, will be 3. If recovery is not successful, the operator is given the following options:

1. Reissue N more times, or
2. Call for a hardware test routine, or
3. Initiate a rollback procedure, or
4. Abort the program.

Major Status: Data Alert (0011) Substatus: Transmission Parity Alert (0X0X10)

Meaning:

Incorrect parity was detected on a data character received from the IOC. This can occur only during a write operation.

Action taken by GEPR:

On Initiation: The rejected command will be reissued N times by GEPR. N, which is easily changed by an EQU card, will be 3. If recovery is not successful, the operator is given the following options:

1. Reissue N more times, or
2. Call for a hardware test routine, or

3. Initiate a rollback procedure, or

4. Abort the program.

Major Status: Data Alert (0011) Substatus: Invalid Select Data (0X01X0)

Meaning:

The start address received during the select command operation was invalid because it specified a nonexistent address.

Action taken by GEPR:

On Initiation: The rejected command will be reissued N times by GEPR. N, which is easily changed by an EQU card, will be 3. If recovery is not successful, the program will be aborted.

Major Status: *Data Alert (0011) Substatus: Check Character Alert (010XX0)

Meaning:

During the reading of a data block from drum, the calculated check character does not agree with the previously recorded check character.

Action taken by GEPR:

On Termination: Retried N times by GEPR. N, which is easily changed by an EQU card, will be 3. The operator is given the following options:

1. Retry N more times, or

2. Return status/substatus to the user program.

Major Status: *End-of-File (0100) Substatus: None (000000)

Meaning:

The data block accessed by the last executed command in the addressed drum was the last allowable block as specified in the Select Drum command. (For linked files, the block is set to a value of 5 if there are sufficient blocks left within the allocated file.) For random files, the block count is set to the number of blocks remaining in the file, as specified from the given Select address, unless the value is greater then 4096. If greater than 4096, the block count is reduced to 4096.

Major Status: *End of Logical File (1111)   Substatus: None (000000)

Meaning:

This status is not sent by hardware but is rather an IOS-implemented return. An attempt was made to write (or read) past the end of the allocated linked file.

Action taken by GEPR:

On Termination: No action taken by GEPR, Status returns are passed back to the user program by IOS.

Major Status: Command Reject (0101)   Substatus: Invalid Operation Code (00XXX1)

Meaning:

The received command is being rejected because the operation code is of an unacceptable configuration.

Action taken by GEPR:

In Initiation: This unrecoverable error is in the user program. There is no possible recovery action that GEPR can take. The program is aborted, and an error message is typed to inform the operator of the error condition and the action taken.

Major Status: Command Reject (0101)   Substatus: Invalid Device Code (00XX1X)

Meaning:

The received command is being rejected because the device code is of an unacceptable configuration.

Action taken by GEPR:

On Initiation: This error is due to a hardware malfunction. There is no possible recovery action that GEPR can take. The program is aborted, and an error message is typed to inform the operator of the error condition and the action taken.

Major Status: Command Reject (0101)   Substatus: Parity Alert on Operation/Device Code (00X1XX)

Meaning:

The received command is being rejected because a parity error has been detected on either the operation or device code of an otherwise acceptable command.

Action taken by GEPR:

On Initiation: The rejected command will be reissued N times by GEPR. N, which is easily changed by an EQU card, will be 3. If recovery is not successful, the operator is given the following options:

1. Reissue N more times, or
2. Call for a hardware test routine, or
3. Initiate a rollback procedure, or
4. Abort the program.

Major Status: Command Reject (0101) Substatus: Invalid Command Sequence (001XXX)

Meaning:

The previously issued Select Drum command has not been successfully completed but a Read Drum, Write Drum, Write Drum and Verify, or Drum Compare and Verify command has been received that requires the successful completion of the Select Drum.

Action taken by GEPR:

On Initiation: The rejected command will be reissued N times by GEPR. N, which is easily changed by an EQU card, will be 3. If recovery is not successful, GEPR initiates an abort.

DISC STORAGE UNIT

Major Status: *Channel Ready (0000) Substatus: None (000000)

Meaning:

If this status is received as an initiation interrupt in response to an REQS or RESS, the disc storage unit is ready to accept a new command.

If this status is received as a termination interrupt, it indicates that the command execution was error free and the disc storage unit is ready to accept a new command.

Action taken by GEPR:

On Initiation and Termination: Status returns are passed back to user program.

Major Status: Device Busy (0001) Substatus: Overridable (000000)

Meaning of Status Return:

This status means that the addressed unit is in the process of executing an off-line operation.

The Overridable substatus indicates that a Seek File command is being executed and this command execution may be overridden by the reception of any other Seek command. This substatus can only be sensed in response to an REQS or RESS.

Action taken by GEPR:

On Initiation: None

Major Status: Device Busy (0001) Substatus: Not Overridable (000001)

Meaning:

The Not Overridable substatus indicates an off-line command other than a Seek is being executed by the addressed file unit. This command cannot be overridden by any other Seek command. This substatus will never be seen by a program using IOS.

Action taken by GEPR: None

Major Status: Attention (0010) Substatus: None (000000)

Meaning:

The disc storage unit is unable to accept an instruction because of some inoperable condition.

Action taken by GEPR:

On Initiation and Termination: A message is typed to the operator giving him the following options:

1. Take corrective action and then retry the command, or
2. Call in a hardware test routine, or
3. Initiate a rollback procedure, or
4. Abort the program, or

Major Status: Data Alert (0011) Substatus: Transfer Timing Alert (000001)

Meaning:

The IOC did not send (write) or receive (read) data characters at a rate compatible with the transfer rate of the subsystem.

Action taken by GEPR:

On Initiation: The rejected command will be reissued N times by GEPR. N, which is easily changed by an EQU card, will be 3. If recovery is not successful, the operator is given the following options:

1. Reissue N more times, or
2. Call for a hardware test routine, or
3. Initiate a rollback procedure, or
4. Abort the program.

Major Status: Data Alert (0011) Substatus: Transmission Parity Alert (XXXX10)

Meaning:

Incorrect parity was detected on a data character received from the IOC. This can occur only during a Write operation.

Action taken by GEPR:

On Initiation: The rejected command will be reissued N times by GEPR. N, which is easily changed by an EQU card, will be 3. If recovery is not successful, the operator is given the following options:

1. Reissue N more times, or
2. Call for a hardware test routine, or
3. Initiate a rollback procedure, or
4. Abort the program.

Major Status: Data Alert (0011)   Substatus: Invalid Control Character (0XX1X0)

Meaning:

Six control characters are used to transmit the physical device address of the data to be read or written on disc. Before all six control characters have been received, an end data transfer signal was sent by the IOC to the Disc Storage Unit.

Action taken by GEPR:

On Termination: This unrecoverable error is in the user program. There is no possible recovery action that GEPR can take. The program is aborted and an error message is typed to inform the operator of the error condition and the action taken.

Major Status: Data Alert (0011)   Substatus: Internal Alert (XX1XX0)

Meaning:

Any one of the following can result in this substatus:

1. Parity error on character read from buffer.
2. An inconsistency in control characters.
3. An error condition is detected within one of the file units.

Action taken by GEPR:

On Initiation: The rejected command will be reissued N times by GEPR. N, which is easily changed by an EQU card, will be 3. If recovery is not successful, the operator is given the following options:

1. Reissue N more times, or
2. Call for a hardware test routine, or
3. Initiate a rollback procedure, or
4. Abort the program.

Major Status: *Data Alert (0011)   Substatus: Check Character Alert (X1XXX0)

Meaning:

During the reading of a data block from disc, the calculated check character does not agree with the previously recorded check character.

DSU

Action taken by GEPR:

On Termination: GEPR will retry the sequence (both the seek and the following read or write) N times. N, easily changed by an EQU card, will be 3. If recovery is not successful, the operator is given the following options:

1. Reissue N more times, or

2. Return the status to the user program.

NOTE: Substatuses (1XXXX0) and (11XXX0) are possible, but not probable, since the associated commands which could result in these statuses are not currently implemented.

Major Status: *End of File (0100) Substatus: None (000000)

Meaning:

The last command execution performed by the addressed disc storage unit resulted in the last consecutive block being accessed.

Action taken by GEPR:

On Termination: Status returns are passed back to user program.

Major Status: *End of Logical File (1111) Substatus: None (000000)

Meaning:

This status is not sent by hardware but is rather an IOS-implemented return. An attempt was made to write (or read) past the end of the allocated linked file.

Action taken by GEPR: None.

On Termination: Status returns are passed back to user program by IOS.

Major Status: Command Reject (0101) Substatus: Invalid Operation Code (0000X1)

Meaning:

The received command is being rejected because either the operation code is of an unacceptable configuration or a parity error was detected on an otherwise acceptable operation code.

Action taken by GEPR:

On Initiation: The rejected command will be reissued N times by GEPR. N, which is easily changed by an EQU card, will be 3. If recovery is not successful, the operator is given the following options:

1. Reissue N more times, or
2. Call for a hardware test routine, or
3. Initiate a rollback procedure, or
4. Abort the program.

Major Status: Command Reject (0101) Substatus: Invalid Device Code (00001X)

Meaning:

The received command is being rejected because either the device code is of an unacceptable configuration or a parity error was detected on an otherwise acceptable device code.

Action taken by GEPR:

On Initiation: The rejected command will be reissued N times by GEPR. N, which is easily changed by an EQU card, will be 3. If recovery is not successful, the operator is given the following options:

1. Reissue N more times, or
2. Call for a hardware test routine, or
3. Initiate a rollback procedure, or
4. Abort the program.

CONSOLE TYPEWRITER

Major Status: *Channel Ready (0000) Substatus: None (000000)

Meaning:

If this status is received as an initiation interrupt in response to an REQS or RESS, the typewriter is ready to accept a new command.

If this status is received as a termination interrupt, it indicates the command execution was error free and the console typewriter is ready to accept a new command.

TYPEWRITER

Action taken by GEPR:

On Initiation and Termination: Status returns are passed back to user program.

Major Status: Attention (0010) Substatus: None (000000)

Meaning:

This status indicates the typewriter is unable to accept an instruction because of some inoperable condition.

Action taken by GEPR:

On Initiation and Termination: The typewriter is inoperable. GEPR will initiate a "Write Alarm" to the console which will cause the sounding of an audio alarm. The sounding of this alarm notifies the operator of the inoperable state of the typewriter. The audio alarm will be sounded periodically until the typewriter becomes operable and the I/O is recovered.

Major Status: Data Alert (0011) Substatus: Transfer Timing Alert (000001)

Meaning:

The IOC did not receive (read) data characters at a rate compatible with the transfer rate of the subsystem.

Action taken by GEPR:

On Termination: The command is reissued by GEPR until it is accepted. Since the cause of this condition is due to a timing problem, retry should eventually correct the alert. After N times, GEPR will sound the audio alarm to notify the operator that there is a malfunction somewhere between the IOC and the console. N,which is easily changed by an EQU card, will be 25.

Major Status: Data Alert (0011) Substatus: Transmission Parity Alert (0XXX10)

Meaning:

Incorrect parity was detected on a data character received from the IOC. This can occur only during a write operation.

Action taken by GEPR:

On Termination: The command will be reissued N times. N, easily changed by an EQU card, will be 3. Each time the command is retried, an exclamation point (!) will be typed for that character that has a parity error and the message will be terminated. If the situation persists, the operator can recognize that a hardware problem exists by the frequency of exclamation points.

If recovery is not successful, a write alarm (audio alarm) will be issued to inform the operator of a malfunction somewhere between the IOC and the console.

Major Status: Data Alert (0011) Substatus: Operator Input Alert (0XX1X0)

Meaning:

While typing a message, the operator realizes that what he has typed is in error. He wishes to have the message ignored. By depressing the OPERATOR INPUT ERROR button, the operator causes this status/substatus to be returned to GEPR.

Action taken by GEPR: None.

On Termination: IOS will reissue the read command.

Major Status: *Data Alert (0011) Substatus: Operator Distracted (0X1XX0)

Meaning:

An interval of 30 seconds has elapsed between the receipt of a Read command and the first data character, or between successive characters, or between the last character to be typed and the actuation of the END OF MESSAGE button.

Action taken by GEPR:

On Termination: Status returns are passed back to user program.

Major Status: *Data Alert (0011) Substatus: Incorrect Format (01XXX0)

Meaning:

An escape character (s) is followed by a nonpermissible character in a message received from the IOC to be typed; or a control character, 17_8 = (?), or 77_8 = (!), is not preceded by the proper number of escapes (77_8).

NOTE: In order to type a (?) or a (!) it is necessary to precede that character with two escapes.

Action taken by GEPR:

On Termination: Status returns are passed back to user program.

TYPEWRITER

Major Status: Data Alert (0011) Substatus: Message Length Alert (100000)

Meaning:

The operator has typed more than the number of characters specified by the DCWs referred to by the Read command.

Action taken by GEPR:

On Termination: GEPR will reissue the command N times. If recovery is not successful, GEPR will initiate an abort. N, easily changed by an EQU card, will be 3.

Major Status: Command Reject (0101) Substatus: Invalid Operation/Device Code (000001)

Meaning:

The received command is being rejected because either the operation code or the device code (or both) is of an unacceptable configuration.

Action taken by GEPR:

On Initiation: This error may be in the user program. There is no recovery action that GEPR can take. The program is aborted.

Major Status: Command Reject (0101) Substatus: Parity Alert on Operation Code (000011)

Meaning:

The received command is being rejected because a parity error has been detected on an otherwise acceptable operation code.

Action taken by GEPR:

On Initiation: The command is reissued N times. N, easily changed by an EQU card will be 3. If recovery is not successful, a write alarm (audio alarm) will be issued to inform the operator of a malfunction somewhere between the IOC and the console.

APPENDIX D
GENERATION AND USE OF A PRODUCTION LIBRARY TAPE FILE

A new function of the GE-625/635 software system has been implemented to provide for the generation and use of a library of production programs which can be easily accessed by the user. The utility program, PRODUCT, will generate a library of user programs on magnetic tape. At STARTUP of the system, this tape file may be mounted on a dedicated tape handler of the system and accessed by the $ PRODUCT control card (see Chapter 5) of the job input deck. The user assumes the responsibility for maintaining the Production Library.

GENERATION OF THE PRODUCTION LIBRARY

The utility program PRODUCT is used to generate a tape library consisting of object programs and overlays required by each program, as shown in Figure 28. Each program and overlay must be uniquely identified. The source of programs must be tapes generated by the System Editor. (See the Appendix of GE-625/635 System Editor, CPB-1138, for a description of the required "System-Loadable Format.")

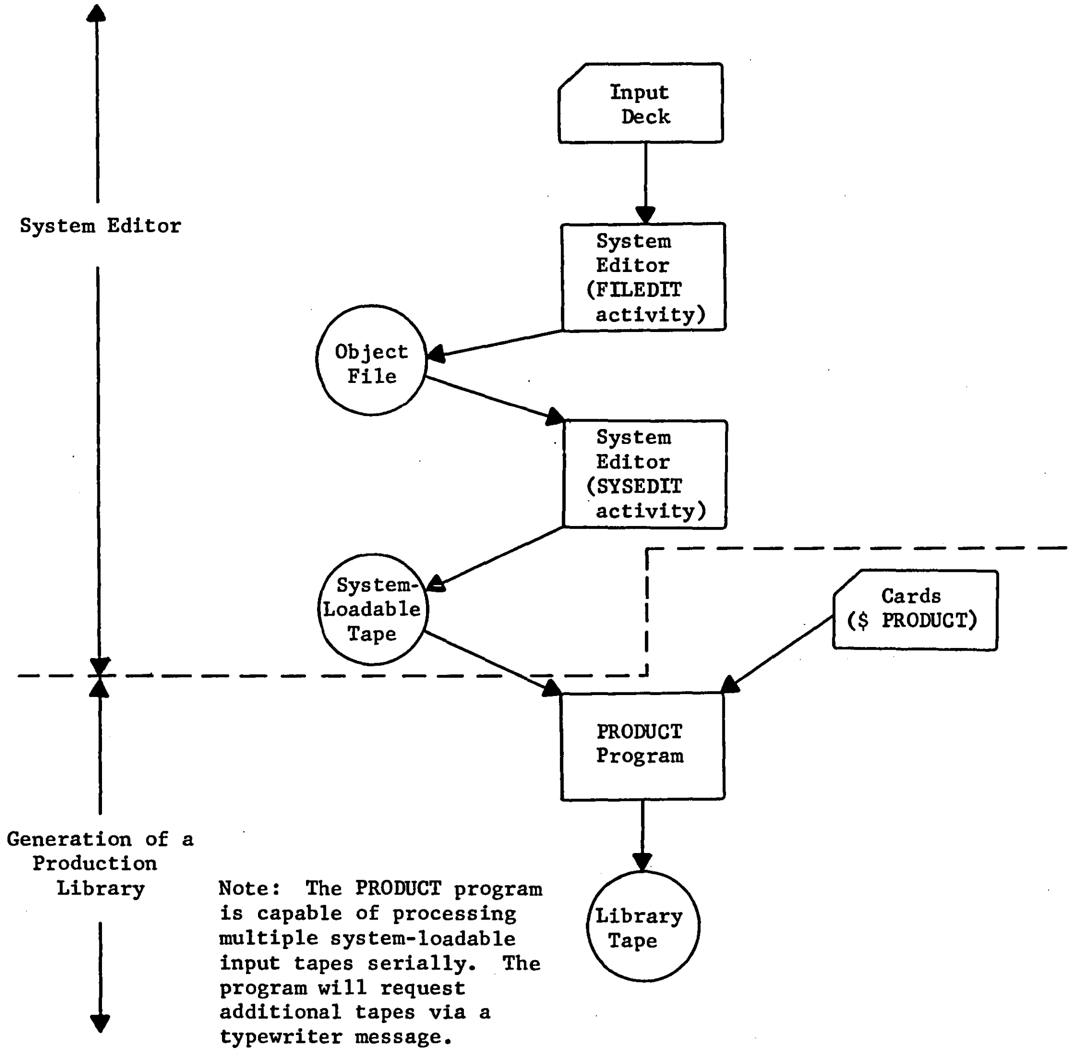
Creating System-Loadable Tapes

The tapes used as the source of programs which will ultimately reside on the Production Tape Library are created in two parts by the System Editor. The first part consists of a FILEDIT activity; the second, a SYSEDIT activity. The deck setup for these activities would be as follows:

```
1        8           16
-------------------------------------------------------------
$        IDENT
$        FILEDIT     INITIALIZE
$        COMMENT     LOAD PART 2 DECK
$        READ        *C, A1R
$        TAPE        R*, X1S, , , , PAY-RSTAR
$        SYSEDIT     INITIALIZE
$        TAPE        R*, X1D
$        TAPE        *1, X2R
$        TAPE        H*, X3D, , , , PAY-SYSTAPE
$        LIMITS      40, 42000
$        ENDJOB
***EOF
```

Note that the tapes allocated above are "saved" by means of the "S" and "D" Disposition codes.

There is a limit on the size of each program that is edited through the System Editor. To edit a program of size n, a $ LIMITS card specifying 13k + n should be supplied. For example, to edit a program requiring 32000 core locations, the card deck for the SYSEDIT activity would contain $ LIMITS ,45000.


System Editor
Input Deck
System Editor (FILEDIT activity)
Object File
System Editor (SYSEDIT activity)
System-Loadable Tape
Cards ($ PRODUCT)
PRODUCT Program
Generation of a Production Library
Library Tape

Note: The PRODUCT program is capable of processing multiple system-loadable input tapes serially. The program will request additional tapes via a typewriter message.

Figure 28. Schematic Diagram of the Generation of a Production Library

The following deck setup supplies the input which will be edited (variable fields contain options which are at the discretion of the user and have not been noted in the example below):

```
1       8          16
_________________________________________________
$       INDEX      ,X
$       SYSLD      CATALOG=PAY1
$       OPTION
$       LOWLOAD
$       OBJECT
          .
          .
$       DKEND
$       LINK       PAY2
$       OBJECT
          .
          .
$       DKEND
$       LINK       PAY3, PAY2
$       OBJECT
          .
          .
$       DKEND
$       EXECUTE
$       ENDLD
$       SYSLD      CATALOG=SAV1
$       OPTION
$       LOWLOAD
$       OBJECT
          .
          .
$       DKEND
$       EXECUTE
$       ENDLD
```

Note: All programs edited through the System Editor must contain a $ LOWLOAD card.

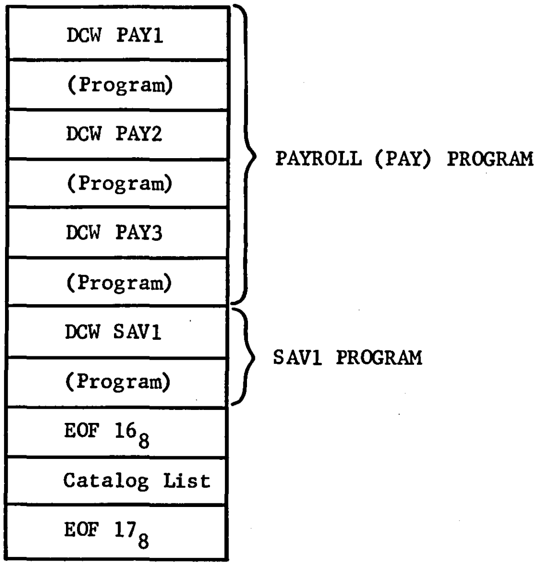
The name specified in the $ SYSLD card will be the name given to the program to identify it uniquely for the PRODUCT program.

The System Editor considers as "fatal" certain GELOAD-recognized "non-fatal" loading errors:

ILLEGAL BINARY CARD
INCONSISTENT FIELD-PREFACE CARD
ILLEGAL CHECKSUM
LOADER SKIPPING TO NEXT BCD CARD
ILLEGAL CLAUSE
LABELED COMMON SIZE INCONSISTENT
ENTRY NOT IN TABLE

This means that user programs that are to be edited for purposes of creating a Production Library must not contain any such "non-fatal" errors.

Running the two activities (described above) will result in system-loadable tapes, which can then be used by the PRODUCT program to create the Production Library. An example of the format of these tapes is shown in Figure 29.



Note: This tape must have been saved in the System Editor run.

Figure 29. System-Loadable Tapes Created by the System Editor

Generating the Production Library

The Production Library is generated by using the system-loadable tape (created by the System Editor) and the Program PRODUCT. (See Figure 28.) The following deck setup will create the Production Library:

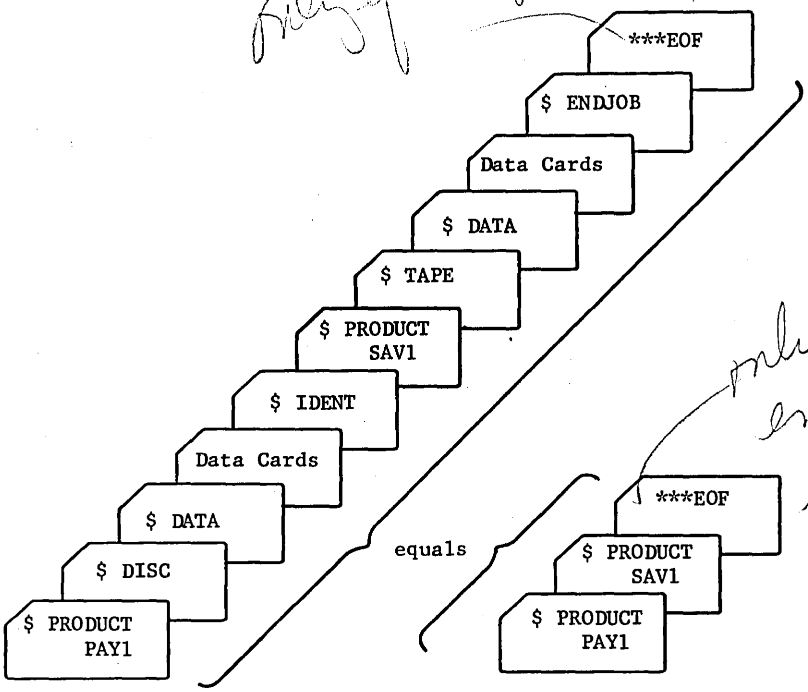
```
1      8       16
$      SNUMB
$      IDENT
$      PROGRAM PROD
$      LIMITS  25,21000
$      TAPE    TP,X1D,,0353,,PAY-HSTAR
$      TAPE    OT,X3D,,,,PROD-LIB
$      DATA    IN,,COPY
$      PRODUCT PAY1
$      ENDCOPY
$      ENDJOB
***EOF
```

Note: All file code designations (TP,OT, and IN) on the $ TAPE cards and the $ DATA cards must be TP, OT, and IN, respectively.

The $ PROGRAM card calls the PRODUCT program from the Software Library. The first $ TAPE card must specify the reel number of the tape created by the System Editor. The PRODUCT program will generate a message requesting that the tape, created in the System Editor run, be mounted. The second $ TAPE card allocates a tape for the Production Library that is being created. The $ DATA IN,,COPY places the $ control cards that are following it on a GEIN - created file. Note: If the input tape used is not in system-loadable format, an EF abort will occur.

A table is constructed from the $ PRODUCT control cards in the input deck. The input deck may consist entirely of $ PRODUCT cards or may contain all of the GECOS control and data cards required to execute each program name appearing on the $ PRODUCT cards. (See Figure 30.) Regardless of whether the input deck consists of GECOS control cards or $ PRODUCT cards, it must immediately follow the $ DATA IN,,COPY card. A $ ENDCOPY card must be used to terminate the copying by GEIN. (Note; however, the $ ENDCOPY card does not close the IN file, this file is closed when it encounters another activity delimiter card.) The table of program requests is augmented by the names of the overlays required by each program. The overlay names are obtained from the complete catalog list contained on the input tape.

filing of empty hopper / (see p 8/33)



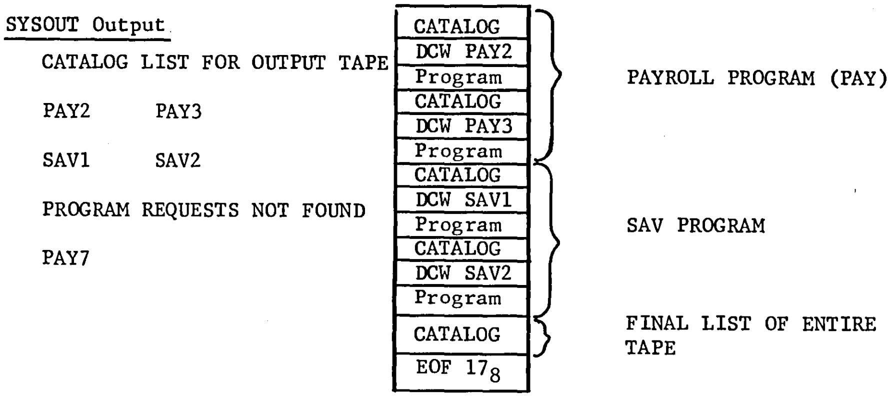
only if empty hopper

Note: This deck must immediately follow the $ DATA IN,,COPY card.

Figure 30. Input Card Deck

The total number of programs, together with the overlays required, must not exceed 1984_{10}. If exceeded, the run is terminated.

As mentioned previously, a catalog list of this tape is printed, including any routines which could not be found (See Figure 31.)



```
SYSOUT Output
    CATALOG LIST FOR OUTPUT TAPE

    PAY2        PAY3

    SAV1        SAV2

    PROGRAM REQUESTS NOT FOUND

    PAY7
```

Figure 31. Production Library and SYSOUT Output Created by PRODUCT

USE OF THE PRODUCTION LIBRARY BY GECOS

The $ PRODUCT control card is used to retrieve and initiate execution of programs from the Product Library, as described in Chapter 5.

$ PRODUCT Card

```
1      8        16
$      PRODUCT  Name, FCB Option, Compiler/Assembler Options
```

The name is a 6-character (or less) name of the program to be called. This name must agree with the name specified in the $ SYSLD card ($ SYSLD CATALOG=Name) noted in the preceding section, "Generation of the Production Library."

If the program designated on the $ PRODUCT card has a GELOAD-generated file control block, and is a linked job, the FCB option should follow the program name and comma. No blank spaces should occur.

Example:

Control card without option

```
1      8        16
$      PRODUCT  Name
```

Control card with option

```
1      8        16
$      PRODUCT  Name, FCB
```

Note: In the basic input deck, the $ PRODUCT control card replaces the $ OBJECT, $ DKEND and $ EXECUTE control cards.

GECOS, upon recognition of the $ PRODUCT control card, will set up the job control stack, allocate memory and peripherals, and at execution time bring into memory the program designated.

If the option FCB is found, the program and the file control block link will be brought into memory at execution time.

The basic input deck consists of:

```
1      8          16
$      |SNUMB     |
$      |IDENT     |
$      |PRODUCT   |PAY1  (replaces $ OBJECT, $ DKEND,
       |          |       $ EXECUTE)
$      |LIMITS    |
$      |TAPE      |AA, X1R, , , , PAY-RECORD
$      |TAPE      |BB, X2R
$      |DATA      |CC
$      |ENDJOB    |
***EOF |          |
```

The $ LIMITS must follow the $ PRODUCT card.

After allocation of the required memory and peripherals, the program is read into core and execution is initiated.

ALTERNATE USE OF A PRODUCTION LIBRARY

The PRODUCT program may also be used to provide an alternate library of programs for a particular customer.

A $ TAPE H* file card may be used to specify that an alternate Production Library is to be used for this job instead of the dedicated Production System Tape or if no dedicated production System Tape is present:

```
1      8          16
$      |TAPE      |H*, X3D, , 6524, PROD-LIB(see note below)
```

Note: The reel number (in the example above, 6524) must agree with the reel number of the H*-formatted tape created in the first phase of the PRODUCT program.

After allocation of memory and peripherals, the requested program will then be searched for in the H* file. Any overlays or FCB will also come from this alternate file.

The designated H* is to contain the program identified in the $ PRODUCT control card and all related FCB's and overlays required by the program.

ADDITIONAL EXAMPLES

The following are additional examples illustrating the generation and use of a Production Library. Example 1 is an example of a job run as a standard GELOAD job.

EXAMPLE 1

```
1      8       16
$      SNUMB   12345
$      IDENT
$      LOWLOAD 1455
$      OPTION  FORTRAN
$      USE     DUMP, FLGEOF,·FRWD·,·FEXIT·,·FVFI·, LBL/6/
$      EQUATE  ·CMN·/TIME/
$      RELCOM  3
$      OBJECT
          .
          .
          .
$      DKEND
$      OBJECT
          .
          .
          .
$      DKEND
$      ENTRY   ......
$      LINK    ALPHA
$      OBJECT
          .
          .
          .
$      DKEND
$      ENTRY   ALPHA0
$      LINK    ASUB2
$      OBJECT
          .
          .
          .
$      DKEND
$      OBJECT
          .
          .
          .
$      DKEND
$      ENTRY   A2
$      LINK    ASUB1, ASUB2, NOPAC
$      OBJECT
          .
          .
          .
$      DKEND
$      ENTRY   A1
$      NOLIB
$      LINK    BETA, ALPHA
$      OBJECT
          .
          .
          .
$      DKEND
```

EXAMPLE 1

```
1      8       16
$      ENTRY   BETA0
$      EXECUTE
$      SYSOUT  07
        .
        .  (Data)
        .
$      ENDJOB
***EOF
```

Example 2 demonstrates how to take this job and create a Production Library using the System Editor and Production Library routine and execute from this Production Library.

EXAMPLE 2 (PART 1)

```
1      8       16
$      SNUMB   67890
$      IDENT
$      FILEDIT INITIALIZE
$      COMMENT *** LOAD PART 2 ***
$      READ    *C, X1R
$      DISC    B*, X1R, 30L
$      TAPE    R*, X1S
$      LIMITS  50, 48000,, 10000
$      SYSEDIT INITIALIZE
$      TAPE    R*, X1R
$      TAPE    H*, X2S
$      TAPE    *1, X3R
$      LIMITS  50, 48000,, 10000
$      PROGRAM PROD
$      TAPE    TP, X2R
$      TAPE    OT, X1S,,,, MY-PROD-LIB
$      DATA    IN,, COPY
$      PRODUCT Name1
$      ENDCOPY
$      PRODUCT TEST, FCB
$      LIMITS  25, 16000, 0, 10000

$      TAPE    H*, X1D
$      SYSOUT  07
$      DATA    I*
        .
        .  (Data)
        .
$      ENDJOB
***EOF
```

Note: This $ DATA card not required in example 1

EXAMPLE 2 (PART 2)

```
1      8              16
       |              |
$      |INDEX         |,TEST
$      |SYSLD         |CATLOG = TEST
$      |LOWLOAD       |1455
$      |OPTION        |FORTRAN
       |              |
       |ESSENTIALLY, SAME DECK SETUP AS EXAMPLE 1 (FROM
       |  $ USE THROUGH $ ENTRY, INCLUSIVE)
       |              |
$      |EXECUTE       |
$      |FFILE         |07
       |              |
       |              |
       |              |
       |              |
       |              |
       |              |
$      |ENDLD         |
***EOF |              |
       |              |
```

This $ FFILE card is necessary to have the Loader generatte a FCB for SYSOUT 07 when test is executed. A $ FFILE card is required for any file whose FCB's are generated by GELOAD, with one exception. The $ FFILE card should not be used for implied FORTRAN files: 05 (I*), 06 (P*), 41 (card reader), 42 (printer), 43 (punch).

After the Production Library has been created, it can be used as shown in Example 3.

EXAMPLE 3

```
1      8              16
       |              |
$      |SNUMB         |94956
$      |IDENT         |
$      |PRODUCT       |TEST, FCB
$      |LIMITS        |25, 16000, 0, 10000
$      |TAPE          |H*, X1D, , XXXX, , MY-PROD-LIB
$      |SYSOUT        |07
$      |DATA          |I*
       |  :     (Data)|
       |  .           |
$      |ENDJOB        |
***EOF |              |
```

APPENDIX E
NAMED DEVICE CAPABILITY AND USE

The capability to allow selection of a particular device peripheral for a special purpose is provided by GECOS. However, disc,. drum, or remote peripherals cannot be selected for special purposes. From an Operations viewpoint, the "Named Device" capability is a valuable feature since repeated forms changing and tape switching can be avoided. For example, all 9SA jobs may be directed to a specific tape handler on which the IBSYS system tape is mounted; BMC jobs requiring special forms may be directed to a printer on which these forms have been previously mounted.

In addition, the flexibility of this implementation allows the designated peripheral to be used in the normal manner when the special functions are not required.

NAMING THE DEVICE

The selection of a specific peripheral device begins with "naming" of the device via the System Description Cards read by Startup. The definition of the Name is specified in the variable field of a $ IOC control card (or one of its associated $ ETC cards). The $ IOC card is fully described in GE-625/635 GECOS-III Startup, CPB-1489. It is repeated here for reference purposes.

1	8	16
$	IOC-#	pub-N, device type, optional fields

where # = IOC no. (0-3)
N = channel (0-15)

Name Considerations and Limitations

Certain factors must be considered in assigning a Name to a device. These are discussed here.

A device can have more than one Name; however, a Name may be assigned to only one device. It is not necessary to name all devices. The Name must be 3 characters.

Only the last 2 characters of the Name are scanned during an attempt at allocation. Therefore, the last 2 characters of the Name must provide the required uniqueness. The first 2 characters must be non-numeric. This distinguishes the Name from the Logical Unit Designator normally used in the second subfield of a $ File control card.

The following is a summary of the restrictions which must be considered, and the possible error messages that may be encountered.

1. The Name must be 3 characters.
2. If the Name is zero, it is invalid. If null, it is assumed to be no name.
3. Character 2 must be non-numeric.
4. Names must be unique when compared with all other Names.
5. The Names:

   TY1 TY2 TY3 TY4 PR1 PR2 {CC1; CP1; TL1;

   may not be used, as these are system functions. So are {ST1; ST2; LBP; MD1 →

   DS1 →
6. Punch devices may have a Name assigned.
7. At least one tape unit must be named. (Startup needs a dedicated tape unit.)

Assigning a name to a peripheral device does not, in itself, exclude that device from normal allocation procedures of GECOS. The named device is treated in a special manner by the Allocator only after that device is "reserved" for special use.

RESERVING A PERIPHERAL

A peripheral may be reserved by the operator via a console typein. (See CPB-1469 for a description of operator-initiated requests.)

Operator - Initiated Request

The operator reserves a peripheral by typing in the following message:

```
DEDCAT nnn
```

where nnn is the Name which will be used to designate the device to be reserved.

If the request to reserve the peripheral is not to be honored, the following messages will be issued by the system:

```
nnn NAME ILLEGAL
nnn NAME NOT FOUND
nnn MULT. NAMED DEU CAN'T DEDICATE
```

This message is in addition to any possible *ASN--- or *MNT---, etc., message.

The specially formatted $ file card which may be used in reserving the peripheral via a system-initiated request is as follows.

This card is also used to allocate an already reserved device.

1	8	16
$	PRINT	File Code, Name and Disposition Code

or:

1	8	16
$	TAPE (TAPE9)	FC, Name & Disp. Code, , FileSer. No., , Filename

As shown above, instead of the normally used Logical Unit Designator, the second field contains the Name of this Named Device and the Disposition Code.

The following is an example of the use of these $ File cards.

1	8	16
$	PRINT	AB, PRAR

The job containing this control card will be allocated to a printer designated as PRA; the file will be released (R) when the activity is completed. The reserved device will not be unreserved, however; this MUST be done by the operator when he so desires.

APPENDIX F
CONVERSION CHART FOR IBM EXTENDED LANGUAGE CODE TO GE-625/635 CODE

IBM Graphic	IBMEL Card Punch	GE-625/635 Code	GE Graphic
0	0	00	0
1	1	01	1
2	2	02	2
3	3	03	3
4	4	04	4
5	5	05	5
6	6	06	6
7	7	07	7
8	8	10	8
9	9	11	9
:	2-8	15	:
#	3-8	13	#
@	4-8	14	@
'	5-8	57	'
=	6-8	75	=
"	7-8	76	"
blank	blank	20	blank
A	12-1	21	A
B	12-2	22	B
C	12-3	23	C
D	12-4	24	D
E	12-5	25	E
F	12-6	26	F
G	12-7	27	G
H	12-8	30	H
I	12-9	31	I
&	12	32	&
.	12-8-3	33	.
<	12-8-4	36	<
(	12-8-5	35	(
+	12-8-6	60	+
logical OR 1	12-8-7	37	\
none	11-0	40	↑
J	11-1	41	J
K	11-2	42	K
L	11-3	43	L
M	11-4	44	M

IBM Graphic	IBMEL Card Punch	GE-625/635 Code	GE Graphic
N	11-5	45	N
O	11-6	46	O
P	11-7	47	P
Q	11-8	50	Q
R	11-9	51	R
-	11	52	-
$	11-8-3	53	$
*	11-8-4	54	*
)	11-8-5	55	)
;	11-8-6	56	;
logical NOT ¬	11-8-7	20	blank
none	12-0	60	+
/	0-1	61	/
S	0-2	62	S
T	0-3	63	T
U	0-4	64	U
V	0-5	65	V
W	0-6	66	W
X	0-7	67	X
Y	0-8	70	Y
Z	0-9	71	Z
none	0-2-8	20	blank
,	0-3-8	73	,
%	0-4-8	74	%
underscore _	0-5-8	20	blank
>	0-6-8	16	>
?	0-7-8	17	?

8/1/68

DOCUMENT REVIEW SHEET

TITLE: GE-625/635 Comprehensive Operating Supervisor (GECOS-III)

CPB #: 1518

From:
Name: BETTY J. COLALUCA
Position: SYSTEMS ANALYST
Address: GE ISS&S METROPOLITAN DISTRICT
250 SHEFFIELD ST.
MOUNTAINSIDE, N.J. 07092

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: A CURSORY REVIEW OF THE DOCUMENT REVEALED THE FOLLOWING ERRORS:

PG 9: The $ TAPE CARD IN THE EXAMPLE NEEDS ANOTHER COMMA AND SHOULD READ: $ TAPE 6*,XID,,123,,ASSEMBLER

PG 17: THE TABLE OF SYSOUT limits OMITS JOVIAL -10,000 (unless the ambiguous ENTRY ILANG implies it) AND the entry for EXECUTE should be in CAPS as are all the rest. Since all the items are given in the form they have on $ CONTROL CARDS, the BMC entry should be changed to CONVERT (BMC).

PG 26: CONGRATULATIONS! That @*!#? diagram of the $ OBJECT card is ALMOST correct! HOWEVER, on the line pointing to the program IDENT field (column 73) you forgot the DOT in front of .LP30000

PG 31: This diagram is partly duplicated on page 17 and the comments made for that page apply here also. In addition, entries for PRODUCT and PROGRAM should be included.

(Continued on attached page.)

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

Please cut along this line

CTB # 1518 Document Review Sheet (cont'd)

Betty J. Colalucca

Pg 38: Rule #4 is now not illegal; it would be interpreted as a single file with two file codes. The disposition code and size & access of file is ignored on the second card, so that one file of 5L is assigned with R disposition; the file may be accessed using either the file code AB or AC.

Pg 43: The definition of File String at the bottom of the page should have the example changed to

```
    ---- CATALOG1$PASSWORD/CATALOG2$PASSWORD/FILENAME$PASSWORD
```

Pg 46: The vertical dotted lines have been omitted from the example diagram.

Pg 47: The (c/b) mark next to Report code 74 should be removed.

Pg 24: After reviewing the $ SELECT card, I feel this card must be included with the $ PERM and $ PRMFL card references in the paragraph on The Identification Subfield on pg 24.

Pg 51: In the description of the S- Save and C- Continue disposition codes, the use of the term "channel-unit designator" is not defined and is not clear in its meaning. "Logical Unit Designator" has always been used; if this new term is being used to refer to only the channel-character-and-unit-number part of the Logical Unit Designator, please say so.

Pg 53: Vertical dotted line at col. 16 missing in example also the reel sequence number 0001 and 001 in the explanation does not match (anyway, leading zeroes are never significant).

Pg 99: The diagram at top of page is partially incorrect in that a card terminates at column 80. Therefore, the diagram should be

```
                 73            80
              | IDENTIFICATION |
              |  (OPTIONAL)    |
```

Pg 103: The last example on the page is grossly incorrect, per the statement. It should be

```
   16
   |/21+25*/26,IN
```

[illegible] NOT 21 OR 25 AND NOT 26

CPB 1518 Document Review Sheet (Cont'd.)

Betty J. Colaluca

Pg 106: In the last example, the $ in column 1 has been omitted!

Pg 107 to 114 incl.: I find that Section 6 on System Files has changed very little from GECOS II documentation, when in reality, these files are handled most differently in GECOS III. Also, the discussion of the primary system job files J* and *J has been completely omitted! An example of contradicting information may be found by checking Figure 3 for the assigned size of *1 file, against the text on page 114. One place says 5L, the other says 2 or 4L, and nowhere does it say that *1 can be assigned to a tape if the size of the source file exceeds 20 blocks.

You will find that the system output files (P*, C*, K*, 06, etc.) are assigned space on the "SYSOUT" job file, J*. Also, the "input" files, such as R*, S*, G*, A*, U*, etc., are assigned space on the job "data" file, *J.

Pg 147, 148: Those Master Catalog operations should be marked in some way to indicate that they can not be executed by regular slaves; they may only be performed by PRIVILEGED slaves, or else every one and his brother will be violating the security of the file system.

Pg 168 to 173 inclusive: I fail to see the purpose of record formats in this user (programmer) reference document. In a maintenance document, yes, but here, no. If you showed the card format, that would be OK.

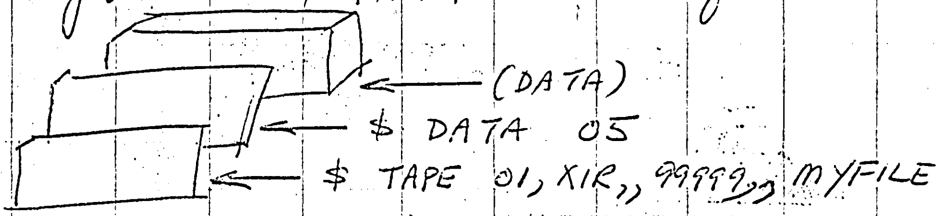
Pg 189: The J* file finally appears! Also, reference is made to a $ BREAK card - where is it documented?

CPB 1518 Document Review Sheet (Cont'd)

Betty J. Colalucca.

Pg 192: In the last paragraph, we have a couple of bad references given. The last sentence should read: "The time-sharing user, at a remote terminal, communicates by means of the ACCESS subsystem which is documented in Appendix B of the _GE-625/635 Time Sharing System TEXT EDITOR_, ~~doc~~ CPB-1515 and by means of the BASIC subsystem which is documented in _GE-625/635 Time Sharing System BASIC Language_, CPB-1510.

Pg 193: Near the back of the example, the $TAPE card needs another comma and should be followed by another $ DATA card as follows:



The omission of the $ DATA card will still permit the jobs to run, but it is, at best, _operating by default_. We should not set examples for our users that works OK as a result of a fluke in the system, but will lead to gross problems if the user doesn't know why this fluke lets him omit it other times.

Pg 194: Figure 21 needs a number with the $ SNUMB card and definitely needs a $ DATA BB card following $ DISC AL, XIR, 3L

Pg 195: Figure 23 should have a $ DATA 05 card
Figure 24 has the $ USE card preceding $ OPTION which is _NOT_ legal !!??

Pg 196: Figure 26 should have $ DATA I* preceding the two references to ← (Data, file code I*).

INFORMATION SYSTEMS

GENERAL ELECTRIC

LITHO U.S.A.