D. WADE

Honeywell

CONTROL CARDS AND ABORT CODES POCKET GUIDE

SERIES 60 (LEVEL 66)

SERIES 6000

GCOS

SOFTWARE

Honeywell

CONTROL CARDS AND ABORT CODES POCKET GUIDE

SERIES 60 (LEVEL 66)
SERIES 6000 **GCOS**

SUBJECT:

Descriptions of Control Cards for Series 60 Level 66 and Series 6000 Programming.

SPECIAL INSTRUCTIONS:

This pocket guide replaces the Control Cards and Abort Codes Pocket Guide, Order No. BJ69, for Series 6000 users. Order No. BJ69 must be used by Series 600 users and by Series 6000 users who are on prior software releases.

DATE:

May 1974

ORDER NUMBER:

DD04, Rev. 0

PREFACE

This guide is an abbreviated pocket reference to the control cards and abort codes used in Series 60 Level 66 or Series 6000 systems. A complete description of the control cards and their uses can be found in the Control Cards Reference Manual, Order No. DD31.

Three appendices cover system file codes, the logical unit designator (LUD), and the device name.

Not included in this guide are Directive control cards and Bulk Media Conversion (BMC) cards, which normally are not used by application programmers. A complete description of these cards is in the Control Cards Reference Manual.

File No.: 1713, 1P13

CONTENTS

CONTENTS (cont.)

CONTENTS (cont)

CONTROL CARDS

BASIC CONTROL CARD RULES

Control cards are used for many different functions, but certain rules apply to all cards:

1. All control cards are identified by a symbol in column 1.
2. The control card name (i.e., SNUMB, IDENT, etc.) normally begins in column 8. ($ Label names can begin in columns 3-7.)
3. The variable field begins in column 16. It must be preceded by a blank in column 15 and cannot exceed column 72 (except where noted).
4. Variables must be separated by commas.
5. A blank terminates the variable field definition. No embedded blanks are permitted. (A dash can be used as a spacer.)

   Exceptions: $ COMMENT, $ IDENT, $ FORM, $ MSGn, and $ MSG3. See the specific description for each of these cards.
6. Variables longer than 12 characters are truncated, except on those cards noted in rule 5.
7. Options can be listed in any order in the variable field, unless otherwise indicated for the specific control card.
8. If options are allowed, but not specified, GCOS uses standard (default) options, which are underlined.
9. Except where noted, no more than six variables are allowed in the operand field.
10. A blank terminates the card scan.

$ ABORT

Variable Field:

16
Options

Purpose: Allows the user to perform file operations (such as file dumps, file copy, etc.) as a subactivity by means of the Utility routine. $ ABORT defines the subactivity initiated by an abnormal termination. Upon encountering $ ABORT, System Input sets the abort bit (bit 12 in the program switch word) to 1 (on).

Options:

JREST	NJREST
REST	NREST

Rules:

1. $ FFILE, $ FUTIL, $ QUTIL, or $ ETC must follow $ ABORT to define utility processing.

NOTE: See Terminate Messages/Abort and Delete Reason Codes Section.

$ ALGOL

Variable Field:

16
Options

Purpose: Calls the ALGOL compiler

Options:

LSTIN	NLSTIN
NLSTOU	LSTOU
DECK	NDECK
NCOMDK	COMDK
NDUMP	DUMP
NDEBUG	DEBUG
JREST	NJREST
REST	NREST
	SYMTAB

Rules:

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

$ ALTER

Variable Field:

16

M,N (N not used when inserting cards)

Where: M = alter 1
N = alter 2

Purpose: Used to make corrections to source card or COMDK input to the various language processors.

Rules:

1. All cards to be placed on the alter (A*) file must be preceded by $ UPDATE.
2. Alter numbers specified on $ ALTER must be in ascending order (i.e., $M \leq N$).
3. The alter file must be prepared in ascending alter number order.

$ BREAK

Variable Field:

16

Not used

Purpose: Enables continued processing of execution activities of a job even though a previous activity has aborted. Compilations and assemblies following an activity abort normally are processed under GCOS. Execution activities (including BMC, Program, and Utility) also can be processed if they are preceded by $ BREAK. Once $ BREAK is encountered, the $ BREAK abort condition is reset and the job continues normally.

Rules:

1. Must precede the $ CONVER, $ PROGRAM, and $ UTILITY card to which it refers.
2. Placement of $ BREAK between compile and execution activities causes an L3 abort.

$ CHANGE

Variable Field:

16

M, N (N not used when inserting cards)

Where: M = change 1
N = change 2

Purpose: Used with the *C Editor (SCED) record maintenance program to insert, replace, or delete records on/from a file in standard system format.

Rules:

1. Change numbers must be in ascending change number order (i.e., M ≤ N).
2. The change file must be prepared in ascending change number order.
3. $ CHANGE allows $ ALTER cards to be contained on the A* file if the user wishes to add $ ALTER cards to a file.

$ COBOL

Variable Field:

16

Options

Purpose: Calls the COBOL compiler.

Options:

LSTIN	NLSTIN
NLSTOU	LSTOU
DECK	NDECK
NCOMDK	COMDK
NDUMP	DUMP
NCOPY	COPY
NEISF	EISF
JREST	NJREST
REST	NREST
LIBCPY	SYMTAB
ON6	SEGMNT
	NOXREF

Rules:

1. 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. Source decks using the COPY clause or RENAMING file option must use the COPY option on $ COBOL. The LIBCPY option takes precedence over the COPY option.

$ COMMENT

Variable Field:

16 72

Comments

Purpose: Used by the programmer to communicate to the operator via the online console.

NOTE: $ MSGn, rather than $ COMMENT, should be used whenever possible.

Rules:

1. If the activity being affected is the first activity in the job, $ COMMENT can precede or follow the activity. $ COMMENT cards affecting any subsequent activities in the job must follow those activities.
2. Only one $ COMMENT card can be used with each activity.
3. The comment cannot exceed column 72.

$ CONVER

Variable Field:

16

Options

Purpose: Calls the Bulk Media Conversion (BMC) program.

Options:

SPIN	NSPIN
NDUMP	DUMP
JREST	NJREST
REST	NREST

Rules:

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

$ DAC

Variable Field:

16

File code,X (any character appended to SNUMB and used as inquiry name)

Purpose: Provides direct access capability, through File and Record Control, between a remote terminal and a slave program in execution.

Rules:

1. Must be preceded by $ USE . RTYP, which causes loading of the proper File and Record Control routine to allow accessing of the terminal through File and Record Control.
2. $ DAC and its associated $ USE card can be used only in execute activities.
3. Must follow the activity definer ($ EXECUTE, etc.), but not necessarily immediately.

$ DATA

Variable Field:

16

File code,Options

Purpose: Writes files onto temporary linked disk for input to a user activity. Files created in this manner cannot be written on by the user. After termination of the job for which this file was created, the file and all references to it are removed from the computer system.

Options:

CKSUM	NCKSUM
NCOPY	COPY
NSEQ	SEQ
ENDFC	TAKEc
IBMF	CKSEQ
IBMC	GE225
IBMEL	

Rules:

1. Must immediately precede data cards to be put onto the specified file.
2. If the COPY option is used, $ ENDCOPY must follow the last data card.

$ DKEND

Variable Field:

16	67	73
Normally not used	Assembly date or NO TTL	Program ID number

Purpose: Identifies the end of an object program or subprogram in the input sequence of a job. Produced by the assembler as the last card of all assembled or compiled programs or subprograms.

Options: CONTINUE
.ALODR

Rules:
1. Must follow an object program or subprogram.

$ DUMMY

Variable Field:

16
GENEW

Purpose: Spawns a job residing on a temporary file.

Rules:
1. The $ DUMMY card image is replaced by a valid $ SNUMB card image.
2. The $ SNUMB and activity number of the originating job appear on the $ SNUMB card of the spawned job (starting in column 36).
3. The execution report from a job spawned from a remote station is returned to that station.
4. $ DUMMY must follow $ DATA.

$ DUMP

Variable Field:

16
Name

Purpose:

Signals General Loader that the cards following should be processed for use with the Debug subroutine.

Rules:

1. Must precede all other cards of the program or overlay except $ LOWLOAD.
2. At least one Debug card must follow $ DUMP.

$ ENDCOPY

Variable Field:

16
File code

Purpose: The file code option is required only if the $ DATA card contains the ENDFC option. The file code specifies the file being built from data preceding $ ENDCOPY.

Rules:

1. $ ENDCOPY must be preceded by $ DATA specifying the COPY option.
2. $ ENDCOPY must follow last data card of file specified in the previous $ DATA card.
3. No $ SNUMB, $ DATA, or $ INCODE card (containing same file code as $ ENDCOPY) can be between $ DATA and its corresponding $ ENDCOPY card.

$ ENDJOB

Variable Field:

16
Not used

Purpose: Indicates the job being processed is a candidate for allocation and execution if no errors are detected. Errors are noted on the console and the printer. System Input completes processing, and the entire job is deleted without being allocated.

Rules: 1. Must be the last card of every job.

$ ENTRY

Variable Field:

16
Name

Purpose: Denotes program SYMDEF (primary or secondary) at which entry is made to execute the program or subprogram. In the overlay mode, this card refers to the current link (see $ LINK).

Rules:

1. Must precede $ EXECUTE.
2. If the job is linked, must be in the same link as the program to which it applies.

***EOF

Variable Field:

1	8	16
***EOF	Not used	Not used

Purpose: A hopper empty status initiates a search by the I/O package to determine if the last card read was ***EOF. If EOF status is set, the card reader is released and System Input is released from memory. If the last card before hopper empty was not ***EOF, the operator is alerted to load the hopper.

During system initialization, ***EOF is placed at the end of each section in the Startup deck and, when detected, Startup determines if all required modules were loaded.

Rules:

1. Should be the last card encountered before a hopper empty status occurs.
2. Used only with card reader input.

$ EQUATE

Variable Field:

16
SYMDEF names

Purpose: Defines SYMDEFs the same as or relative to those already defined. Or, equates labeled common relative to blank common.

Rules:

1. Must precede $ EXECUTE.
2. If used for labeled common, precedes programs containing labeled common region.

$ ETC

Variable Field:

16

Continuation of preceding card

Purpose: Continues variables of preceding card.

Rules:

1. Immediately follows card being continued.
2. A comma must follow last variable on preceding card; if preceding card is $ PRMFL, comma or slash (/) can follow variable.

$ EXECUTE

Variable Field:

16

Sense switches, and/or Options

Purpose: Requests loading/execution of object program.

Options:

NDUMP	DUMP
NPURGE	PURGE
NCLEAR	CLEAR
JREST	NJREST
REST	NREST
SEG	DEBUG

Rules:

1. Must appear after all subprograms to be executed, but before their data.
2. If variable field is null, switches are assumed off.

Sense Switches:

Turned on when Assembler/Compiler options are specified.

ON1 - Sense Switch 1 (bit 6 of program switch word)
ON2 - Sense Switch 2 (bit 7)
ON3 - Sense Switch 3 (bit 8)
ON4 - Sense Switch 4 (bit 9)
ON5 - Sense Switch 5 (bit 10)
ON6 - Sense Switch 6 (bit 11)

$ EXTEDIT

Variable Field:

16
Not used

Purpose: Directs GCOS to call the System Extension File Generator to produce an E* file, which is used to add patches or corrections to existing system routines or overlays at load time.

Rules:

1. Must precede the activity control stack which must include a $ TAPE, $ TAPE7, $ TAPE9, or $ FILE control card specifying E* with a save disposition code.
2. When assigned to nontape device, E* must be random linked.

$ EXTEND

Variable Field:

16
Options

Purpose: Enables checkout of octal patching of system programs without reediting the software library program. Upon encountering $ EXTEND, System Input sets the extend bit (bit 16 in the program switch word) to 1 (on).

Options:

<u>JREST</u>	NJREST
<u>REST</u>	NREST

Rules:

1. Must immediately follow the system call card of the software system to be extended or patched.
2. The contents of the E* file must previously have been created and assigned to E* via a $ TAPE, $ TAPE7, $ TAPE9, or $ FILE card. When assigned to a nontape device, E* must be random linked.

$ FFILE

Variable Field:

16

File code,Parameters,FILNAM/File name

Purpose: Used by General Loader to describe nonstandard file control blocks (FCBs). One FCB is created for each $ FFILE in the deck.

Parameters:

<u>STDLBL</u>
NBUFFS/n
LGU/(nn,mm,..)
MLTFIL
MLTRL
PTMODS
PTMODE
ASA9
NOSRLS (or NSER)
IGNORE
POSHED/SYMDEF
POSTRL/SYMDEF
MIXLNG/SYMDEF
NOSLEW
NDATE
PRTREC

NSTDLB (or NLABEL)
BUFSIZ/n
RETPER/n (or Rxxx)
MODBCD (or MBCD)
PTMODD
MODMIX
FIXLNG/n (or Fxxx)
LODENS (or LDENS)
PREHED/SYMDEF
PRETRL/SYMDEF
ERRXIT/SYMDEF
DSTCOD/(PRNTR,
MTAPE,...)
PHYREC
RANDOM/n

Rules:

1. Must follow (a) $ EXECUTE when used with a General Loader activity ($ OPTION must be included), (b) $ UTILITY when used with a Utility activity, and (c) $ ABORT when used with an abort subactivity.

2. $ OPTION specifies the FCB and is required when $ FFILE is used with a General Loader activity.

3. The PHYREC option cannot be used for mass storage device files.

4. The PHYREC and NDATE options can be used only with a Utility activity. The DSTCOD option can be used only with ALGOL.

NOTE: See Appendix A - File Codes.

$ FILE

Variable Field:

```
16
File code,LUD or Device name,Access,Type
```

Purpose: Sets up allocation of a mass storage file on a particular type of device. If no device type is specified or if the requested device is not configured, allocation is on the "fastest" device available.

Can be used to assign files on a reserved (dedicated) unit.

Permissible device types are:

DSS180
DSS181
DSS191

DSS167 (Series 6000 only)
DSS170 (Series 6000 only)
DSS190 (Series 6000 only)
DSS270 (Series 6000 only)
MSU0310 (Series 60 only)
MSU0400 (Series 60 only)

A new cataloged file can be created via:

```
16
File code,LUD,Size,NEW,Filename
```

A single-level file without passwords in the master catalog can be accessed via:

```
16
File code,LUD,Access,OLD,Filename
```

An option is available to allocate a NULL file, which provides an output sink and an input EOF.

Rules:

1. Must follow activity definer ($ COBOL, $ EXECUTE, etc.), but not necessarily immediately.
2. When no disposition code is indicated, R (release) is assumed.
3. If number of links (access field) is blank, one is implied.
4. If creating a new cataloged file, a catalog must previously have been established. $ USERID must precede activity in which $ FILE exists.

NOTE: See Appendix B - Logical Unit Designator, and Appendix C - Device Name.

$ FILEDIT

Variable Field:

16
Options

Purpose: Directs GCOS to call the Source and/or Object Library Editors.

Options:

NOSOURCE	SOURCE
OBJECT	NOBJECT
UPDATE	INITIALIZE
IGNORE	TRACE
OLD	NEW
NDUMP	DUMP
NEISF	EISF
JREST	NJREST
REST	JREST
6000F	355F
IOMF	NONE
SAVEB	EMOF
NEMOF	

Rules:

1. Must be the first control card of the activity.

$ FILSYS

Variable Field:

16
Options

Purpose: Calls the GCOS FILSYS activity.

Options:

NDUMP	DUMP
JREST	NJREST
REST	NREST

Rules:

1. Must precede the directives to be processed by FILSYS and must precede any other control card associated with that activity.

2. The DUMP option is ignored unless $ PRIVITY also is present for the activity.

$ FORM

Variable Field:

16

Form identification

Purpose: Used with $ CONVER when a specific form is required for printer or punch output. The form identification (up to 51 characters) from $ FORM is issued on the console.

Rules:

1. Must follow $ CONVER.
2. Only one $ FORM card can be used with each activity.

$ FORTY ($ FORTRAN)

Variable Field:

16

Options

Purpose: Calls the FORTRAN compiler.

Options:

LSTIN	NLSTIN
NLSTOU	LSTOU
NDECK	DECK
NCOMDK	COMDK
NDUMP	DUMP
NDEBUG	DEBUG
NXREF	XREF
FORM	NFORM
NLNO	LNO
BCD	ASCII
NOPTZ	OPTZ
NOMAP	MAP
JREST	NJREST
REST	NREST

Rules:

1. Must precede source cards of program(s) and/or subprogram(s) to be processed.
2. Must precede any other control card associated with that activity.
3. Must precede each program or subprogram to be compiled.

$ FUTIL

Variable Field:

```
16
File code 1,File code 2,Options
```

Purpose: A directive in the Utility activity to direct certain operations so that Utility produces a combination octal-BCD-ASCII dump, copies certain files, or compares certain files.

Options:

```
COPY/m/              COMP/m/
DUMP/m,n/            DDUMP/m,n/
SKIP/m,n/            MCOPY/m/
HOLD/FC1,FC2/        RWD (or REW)/FC1,
ADUMP/m,n/              FC2/
RSAVE/1F/            RPT/mP,nT/
RREST/1F/            AADUMP/m,n/
BKSP/m,n/            RCOPY/1F/
```

NOTE: See Appendix A - File Codes.

Rules:

1. $ FFILE must precede the corresponding $ FUTIL. Multiple $ FUTIL cards are permitted within an activity.
2. $ QUTIL must precede $ FUTIL if processing options $ QUTIL defines are in effect for functions specified on $ FUTIL.
3. Processing of a given random file (via RANDOM on $ FFILE) must be completed on one $ FUTIL card.
4. Nesting an RPT option within itself is not permitted.
5. $ FUTIL must not precede $ UTILITY.
6. The RPT option and its "to be repeated" parameters must be included on the same $ ETC card that is preceded by $ FUTIL.
7. $ FUTIL ...BKSP/m,n/ must be preceded by $ FFILE...FC, PHYREC.

$ GMAP

Variable Field:

16

Options

Purpose: Calls the Macro Assembler.

Options:

LSTOU	NLSTOU
DECK	NDECK
NCOMDK	COMDK
GMAC	NGMAC
NDUMP	DUMP
ON5	NXEC
JREST	NJREST
REST	NREST
NEMOF	EMOF
6000F	SYMTAB
IOMF	355F
EISF	NEISF

Rules:

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

$ GOTO

Variable Field:

16

Label

Purpose: Permits an unconditional transfer of control to the portion of the program specified by the label name; or, to end of job.

Rules:

1. The label cannot exceed six characters.
2. $ GOTO can skip around $ USERID and $ PRMFL. Before the job is begun, a test for user identification validity and permanent file presence is made. Non-existent user identification and file names cause job deletion.
3. If the named control label precedes the $ GOTO card referencing the label, the job is terminated when $ GOTO is encountered.

$ IDENT

Variable Field:

16

Account number,Identification

Purpose: Identifies the user of a job or activity and supplies the user's account number. Each activity can be preceded by $ IDENT or $ IDENT can be used to identify a series of activities. Identification also is used for printing and punching SYSOUT banners.

Rules:

1. At least one $ IDENT card must immediately follow $ SNUMB.
2. The account number must not exceed 12 characters.
3. Columns 16-72 are available during activity execution in words 66-77 (octal) of the slave program prefix.

$ IDS

Variable Field:

16

Options

Purpose: Calls the I-D-S translator.

Options:

LSTIN	NLSTIN
NLSTOU	LSTOU
DECK	NDECK
NCOMDK	COMDK
NDUMP	DUMP
NCOPY	COPY
ON6	SYMTAB
NEISF	EISF
JREST	NJREST
REST	NREST
LIBCPY	SEGMNT
	NOXREF

Rules:

1. 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. All source decks that use the COPY clause or the RENAMING file options must use the COPY option on $ IDS.

$ IF

Variable Field:

16
Operator,Label

Purpose: Permits conditional branching to a following activity 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. Allows the conditional execution of activities.

Operator Symbols: + - OR blank - YES

* - AND / - NOT

Rules:

1. Must follow all control cards, except $ IF cards, relating to an activity and must precede the $ Label card referenced.
2. Cannot be continued by $ ETC; successive $ IF cards can be used.

$ INCODE

Variable Field:

16
BCD card code

Purpose: Indicates that subsequent BCD input cards must undergo character transliteration from the character set specified on $ INCODE to the internal binary representation of the GCOS standard character set. All BCD cards, except $ control cards, following $ INCODE undergo transliteration.

Rules:

1. Must follow the $ control card denoting the compiler language and precede the data to be transliterated.
2. Should precede the data card images on the permanent file.
3. Does not perform transliteration on a $ DATA,,COPY...$ ENDCOPY set.
4. Cannot precede a COMDK.

Card Codes:

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

$ JOVIAL

Variable Field:

16
Options

Purpose: Calls the JOVIAL compiler.

Options:

LSTIN	NLSTIN
NLSTOU	LSTOU
DECK	NDECK
NCOMDK	COMDK
NDEBUG	DEBUG
NDUMP	DUMP
JREST	NJREST
REST	NREST
	SYMTAB

Rules:

1. 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. JOVIAL uses the standard $ OPTION, $ LIMITS, $ INCODE, $ EXECUTE, and $ TAPE control cards.

3. If the source deck is punched using the IBM character set, $ INCODE must be placed between $ JOVIAL and the first card of the source deck.

$ Label

Variable Field:

1	3-7
$	Label: or Label.

Purpose: Defines the entry to which control is transferred from a preceding $ GOTO, $ IF, or $ WHEN.

Rules:

1. Must follow $ GOTO, $ IF, or $ WHEN card referencing the label.

2. The label can begin in any of columns 3-7.

3. Termination must be with a colon or a period (followed by at least one blank).

$ LIBRARY

Variable Field:

16
File code

Purpose: Indicates to General Loader that user libraries are present and will be searched before the system subroutine library search.

Rules:

1. A file control card referencing the same file code must be included with the program and the file must be labeled.
2. Must precede the first $ LINK card in a linked overlay.
3. A maximum of 10 file codes (per job) is allowed on $ LIBRARY.

NOTE: See Appendix A - File Codes.

$ LINK

Variable Field:

16
Name,Origin,Option

Purpose: Specifies the beginning of an overlay or link and the origin of the overlay.

Option: NOPAC

Rules:

1. Must precede $ EXECUTE for the activity.
2. The name specified in the first field must be a unique alphanumeric identifier not longer than six characters and must not be a SYMDEF.
3. If present during loading of COBOL segmented programs, communication is possible only between the permanent portions of each link.

$ LIMITS

Variable Field: ✓ 1/100ths of an hour.
Eg 10 = 0.10Hours

16
Time,Storage-1,Storage-2, Print lines,I/O time

Purpose: Modifies standard activity limits. If omitted, allocation is made as follows:

Activity	Hours	Memory	Lines
ALGOL	0.08	28K	10,000
CONVER	0.20	7K	1,000
COBOL	0.15	32K	20,000
EXECUTE	0.05	16K	5,000
EXTEDIT/ FILEDIT	0.04	32K	10,000
FILSYS	0.03	32K	1,000
FORTY/FORTRAN	0.05	26K	12,000
GMAP	0.04	24K	10,000
IDS	0.15	32K	20,000
JOVIAL	0.08	28K	10,000
PROGRAM	0.05	16K	5,000
SYSEDIT	0.05	32K	10,000
UTILITY	0.03	10K	10,000
355MAP	0.04	32K	10,000
355SIM	0.08	25K	5,000

Compilers, assemblers, and executions with limits of more than 2500 lines of SYSOUT require 2K SSA; less than 2500 lines, 1K SSA.

Rules:

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

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

3. $ LIMITS preceding the first activity definition card ($ FORTRAN, $ EXECUTE, etc.) in the job deck is interpreted as a job limit definition. Processor run time, I/O time, and SYSOUT line limit fields limit the job as a whole. Actual resources used in each activity are subtracted from the job total. The job terminates when the remaining resource is zero.

$ LOWLOAD

Variable Field:

16

Integer or Blank,Option

Purpose: Initiates loading just above the slave program prefix (64 words) and the control words (two words) if a link job is being loaded.

Option: LSW

Rules:

1. Must precede all subprograms and control cards used by General Loader to define program and/or data space in memory.
2. Need not precede control cards that have nothing to do with assignment of space in memory (i.e., $ ENTRY, $ OPTION, etc.).

$ MSGn

Variable Field:

16

C,Message text

Where: C=1,2,3, or 4, corresponding to TY1, TY2, TY3, and TY4.

Purpose: Provides user at remote stations with capability of issuing messages to central site operators.

Rules:

1. MSG1 can appear anywhere in the job.
2. MSG2 follows the activity definer ($ COBOL, $ EXECUTE, etc.).

MSG3

ariable Field:

16

Hold until date/Hold until time

urpose: Allows user to hold a job in the System Scheduler until a specified date/time.

ules:

1. Hold until date format is mmddyy (mm = month; dd = day; yy = year).
2. Hold until time can take three formats: hh:mm (hh = hours, mm = minutes); and, hh.xx or hhxx (hh = hours, xx = hundredths of hours).
3. No check is made to verify date or time.
4. Can appear anywhere in job.

NLOAD

ariable Field:

16

SYMREFs (10 maximum)

urpose: Allows the user to have calls to subroutines that are to be brought in from a library (system or user library) but not in the link that contains the call. If the same subroutines are not to be loaded in a subsequent link, $ NLOAD must be repeated.

ules:

1. Can be placed anywhere in the link except in a binary deck.
2. If the same subroutines are not to be loaded in two or more links, $ NLOAD must be included within each link.
3. If $ NLOAD is present during loading of COBOL segmented programs, it must precede the first segmented program.

$ NOLIB

Variable Field:

16
File code or Blank

Purpose: Prevents a library search for the current program or link being loaded (see $ LINK description). If operand field is blank, no search is made of either the user or system libraries. If operand field contains a file code(s), only those libraries (files) are not searched.

Rules:

1. Must precede $ EXECUTE for the activity.
2. A separate $ NOLIB card must be used in each link overlay where no library search is desired.
3. If $ NOLIB is present during loading of COBOL segmented programs, it must precede the first segmented program.

$ NTAPE

Variable Field:

16
File code, Channel designator, Number of tapes

Purpose: Used by System Input to request a number of utility tapes to be assigned to the tape channel specified in the operand field.

Rules:

1. Must follow the activity definer ($ COBOL, $ EXECUTE, etc.), but not necessarily immediately.
2. If the number of tapes subfield is blank or zero, one tape is allocated. The other subfields in the operand field must be present.
3. If used for COBOL Sort collation tapes, the file code must be S1.

NOTE: See Appendix A – File Codes, Appendix B – Logical Unit Designator, and Appendix C – Device Name.

OBJECT

ariable Field:

16 60	61	67	73 80
Remarks and/or sequence option	Time of assembly or blank	Date of assembly or TTL date	Program identification number

Source Identification

urpose: Identifies the beginning of an object program or subprogram in the input sequence of a job. Is produced by the assembler or compiler as the first card of an object program or subprogram deck.

ptions: SEQ CKSEQ
NSEQ

ules: 1. Must be the first card of the object deck. The next $ control card after $ OBJECT must be $ DKEND.

OPTION

ariable Field:

16
Options

urpose: Alters the standard General Loader options during loading.

ptions:

MAP NOMAP
CONGO GO
NOGO SET/n/
NOSREF SYMREF
LOCOMN ERCNT/n/
NOSETU FCB
NOFCB COBOL
ALGOL JOVIAL
FORTRAN SAVOLD/Name
SAVE/Name RELMEM
NOMSUB

ules:

1. Must precede the object deck for which the option is to take place.

2. $ OPTION must precede $ USE if the LOCOMN option is used.

3. When loading COBOL segmented programs, $ OPTION...COBOL must precede the first segmented program.

$ PARAM

Variable Field:

16
Options

Purpose: Allows alteration of control information stored in the Select file.

Rules:

1. The $ COMMENT, $ IDENT, $ MSG1, $ MSG2, $ MSG3, $ SELECT, $ SNUMB, $ SELECTD, and $ USERID cards cannot be modified by $ PARAM.
2. Must precede the first replacement operator (#n) used, but multiple $ PARAM cards can be interspersed with $ control cards to provide more than nine parameters.
3. Stranger options are not permitted on $ PARAM.

$ PPTP ($ PPTR)

Variable Field:

16
File code, LUD or Device name

NOTE: See Appendix A — File Codes, Appendix B — Logical Unit Designator, and Appendix C — Device Name.

Purpose: Allocates the perforated tape punch (reader). Also can be used to allocate a named device.

Rules:

1. The file code indicated is the same as the file code that is given as subfield 2 in the file control block macro instruction.
2. Must follow (not necessarily immediately) the activity definer ($ COBOL, $ EXECUTE, etc.).

$ PRINT

Variable Field:

16

File code,LUD or Device name,Printer/160, Train #/VFC

NOTE: See Appendix A — File Codes, Appendix B — Logical Unit Designator, and Appendix C — Device Name.

Purpose: Allocates a high-speed printer as follows:

Field	Default Allocation
null	PRT201, PRT203/PRU1100,PRT300, PRT303,PRT401/PRU1200, or PRT402/PRU1600
PRT201	PRT201
PRT203	PRT201 or PRT203/PRU1100
PRT300	PRT300 or PRT303
PRT303	PRT303 or PRT300
PRT401	PRT401/PRU1200 or PRT402/ PRU1600
PRT402	PRT402/PRU1600 or PRT401/ PRU1200

Rules:

1. A file code must be present.
2. The train number can be the site BCD standard, octal train identification, or train name.
3. Must follow the activity definer ($ COBOL, $ EXECUTE, etc.), but not necessarily immediately.
4. The /VFC can be the site standard VFC, the VFC name, or a file code identifying the file containing a VFC image.

$ PRIVITY

Variable Field:

16
Not used

Purpose: Causes Allocator to query operator, via the control console, whether to permit job execution. If operator permits job execution, MME .EMM is legal for activity. If no $ PRIVITY is present when the attempt is made to execute MME .EMM, the user program is aborted.

Rules:

1. Must follow the activity definer ($ COBOL, $ EXECUTE, etc.), but not necessarily immediately.

$ PRMFL

Variable Field:

16
File code,Permit,Mode,File string
or
File code/LUD,Permit,Mode,File string

Purpose: Accesses a permanent data file previously created by a FILSYS activity.

Rules:

1. Must be preceded by $ USERID.
2. $ ETC continues the operand field.
3. The symbols dollar sign ($), slash (/), and comma (,) are delimiters and must not be embedded in a $ PRMFL field.
4. Must follow the activity definer ($ COBOL, $ EXECUTE, etc.), but not necessarily immediately.

NOTE: See Appendix B — Logical Unit Designator, and Appendix C — Device Name.

$ PROGRAM

Variable Field:

16
Name,Options

Purpose: Calls a program previously edited onto the software library for which no call capability exists. Also calls programs not known to system from user dynamic file. Standard GCOS limits are assumed. File control cards following $ PROGRAM are put on R*. Output files P* and T* also are included in the slave PAT.

Options: Can be any options currently available with $ EXECUTE, $ COBOL, $ FORTY ($ FORTRAN), $ ALGOL, and $ GMAP. No more than five options are recognized.

Rules:

1. Must precede the source card of each program or subprogram to be processed and must precede any other control card associated with the activity.
2. Any data file following $ PROGRAM is placed on I* unless preceded by $ DATA specifying a card code.

$ PUNCH (GCOS)

Variable Field:

16
File code,LUD or Device name,Punch

Purpose: Allocates card punch as follows:

Field	Allocation
null	CPZ100, CPZ200,CPZ201, CPZ300/PCU0120, or CPZ301
CPZ100	CPZ100,CPZ200, CPZ201
CPZ200	CPZ200,CPZ201,CPZ100
CPZ201	CPZ201,CPZ100, CPZ200
CPZ300	CPZ300/PCU0120, or CPZ301
CPZ301	CPZ301 or CPZ300/PCU0120

Rules:

1. Must follow the activity definer ($ COBOL, $ EXECUTE, etc.), but not necessarily immediately.

$ QUTIL

Variable Field:

16
Options

Purpose: Specifies Utility processing options.

Options:
EOF/ALL
NBYPSS
RESEQ
USER
ASIS
CMPERR/n
USE/n (or IGNORE/n)

Rules:

1. Must precede $ FUTIL if the processing options it defines are to be in effect for the functions specified on $ FUTIL.
2. The options specified on $ QUTIL remain in effect for the remainder of the activity, with three exceptions: USE, IGNORE, and CMPERR options can be reset in the same activity by use of another $ QUTIL card.

$ READ

Variable Field:

16
File code, LUD or Device name, Reader

Purpose: Allocates the card reader as follows:

Field	Allocation
null	CRZ200, CRZ201 or CRZ301/ CRU1050
CRZ200	CRZ200 or CRZ201
CRZ201	CRZ201 or CRZ200
CRZ301	CRZ301/CRU1050

Rules:

1. Must follow the activity definer ($ COBOL, $ EXECUTE, etc.), but not necessarily immediately.

$ RELCOM

Variable Field:

16
Integer

Purpose: Increments the blank common loading counter by the amount specified in the variable field.

Rules:

1. Must precede any object decks and compiler/assembler activities (subprograms).

$ REMOTE

Variable Field:

16
File code,Destination (2 characters)

Purpose: Assigns an output file to a remote station for use by Network Processing Supervisor (NPS) or General Remote Terminal Supervisor (GRTS).

Rules:

1. Use $ SYSOUT to generate output at the central station when submitting from a remote site.
2. A user program submitted at the central site can use $ REMOTE to send output to a remote terminal.
3. Must follow the activity definer ($ COBOL, $ EXECUTE, etc.), but not necessarily immediately.

$ REPORT

Variable Field:

16
Report code,Device code,Media data

Purpose: Used by SYSOUT to change the standard paper on the printer to a special form or to change the card stock on the card punch.

Device Codes:

PR or 3 - Printer
PU, 1, or 2 - Punch

Rules:

1. A separate $ REPORT card must be used to define each report when there is more than one report for an activity.
2. A combined total of six $ REPORT, $ REPTL, and $ REPTR cards can be used in a single activity.
3. Must follow the activity definer ($ COBOL, $ EXECUTE, etc.), but not necessarily immediately.

$ REPTL

Variable Field:

16
Report code,VFC, Paper

Purpose: Employs extended print line capabilities of PRT401/PRU1200 and PRT402/PRU1600 printers (with the 160-character print line extension option). $ REPTL is used by SYSOUT to change the standard paper on the printer to a special form, to specify VFC control, and to specify that 160-character print line option is required for the specified report.

Rules:

1. $ REPTL is effective for only one report per activity.
2. A combined total of six $ REPORT, $ REPTL, and $ REPTR cards can be used in a single activity.
3. Must follow the activity definer ($ COBOL, $ EXECUTE, etc.), but not necessarily immediately.

$ REPTR

Variable Field:

16

Report code,VFC,Paper

Purpose: Used by SYSOUT to change the standard paper on the printer to a special form and to specify the VFC or VFU-loop.

Rules:

1. $ REPTR is effective for only one report per activity.
2. A combined total of six $ REPORT, $ REPTL, and $ REPTR cards can be used in a single activity.
3. Must follow the activity definer ($ COBOL, $ EXECUTE, etc.), but not necessarily immediately.

$ SELECT

Variable Field:

16

Catalog-Name-1$Password-1/Filename-n $Password-n

Purpose: Instructs System Input to call the file system, which searches the permanent file catalog and locates the required catalog entry. System Input continues processing its input from the permanent file just as it did from the normal input device. If the catalog entry does not exist for the file card, the job is deleted.

Rules:

1. Before any $ SELECT request can be made for GCOS permanent files, the user must be identified to the file system by $ USERID.
2. $ SNUMB cannot be contained within a Select file.
3. If no file content has been established, $ SELECT cannot be used.
4. Control card nesting (use of $ SELECT within a Select activity) is permitted to a depth of 10.
5. $ SELECT can appear anyplace in the deck (following $ SNUMB and $ USERID).

$ SELECTD

Variable Field:

16

Catalog-Name-1$Password-1/Filename-n
$Password-n

Purpose: Allows System Input to build one data file comprised of card decks and control cards from several select files. System Input writes the data onto I*. $ SELECTD is written only to J*.

Rules:

1. $ SELECTD is ignored if encountered outside the $ DATA FC,,COPY... $ ENDCOPY sequence.
2. Control card nesting is permitted to a depth of 10.
3. If no file content has been established, $ SELECTD cannot be used.
4. $ SNUMB cannot be contained within a Select file.

$ SET

Variable Field:

16

Bit numbers to be set or reset in program switch word

Purpose: Can be used to set or reset bits in the program switch word external to activity execution. These bits then can be tested by a MME GESETS or a MME GERETS. They also can cause branching by utilizing $ IF or $ WHEN.

Rules:

1. If switch word bits are being set for a particular activity (i.e., GMAP), $ SET must follow the activity definer ($ GMAP).

$ SNUMB

Variable Field:

16
Job identifier, Urgency level

Purpose: Used by System Input to identify the job internally and to assign an urgency to the job for use in its allocation.

Rules:

1. Must be the first card of every job.
2. Must be followed immediately by at least one $ IDENT card.
3. The job identifier must not be blank or all zeros and must not contain a comma, dollar sign, slash, colon, or period. Detection of any of these causes job deletion.
4. A file invoked by $ SELECT must not contain $ SNUMB.

Urgency: 1-63. If none, 5 assumed. "Threshold" = 40.

$ SOURCE

Variable Field:

16
Not used

Purpose: Created by system on R* when $ system call card is encountered. Instructs General Loader to load this subprogram before continuing loading from R*. Used to reload user program saved on B*.

Rules:

1. Should precede $ EXECUTE and $ PRMFL B* when reloading saved program.

$ SYSEDIT

Variable Field:

16
Options

Purpose: Directs GCOS to call System Library Editor.

Options:

NDUMP	DUMP
JREST	NJREST
REST	NREST

Rules:

1. Precedes source cards of each program or subprogram to be processed.
2. Precedes any control card associated with that activity.

$ SYSOUT

Variable Field:

16
File code, Destination

Purpose: If no Destination field, used by System Input to assign output files to SYSOUT. Unless otherwise defined, P* is automatically assigned to SYSOUT. $ SYSOUT permits assignment of additional files. If destination field included, output is generated for specified destination.

Rules:

1. Used only for files recorded in standard system format with appropriate media and report codes included in logical record.
2. There is no restriction on the number of files assigned to SYSOUT. However, only nine report codes are generated.
3. Must follow the activity definer ($ COBOL, $ EXECUTE, etc.), but not necessarily immediately.
4. Neither file code nor station code can be 00.

TAPE, $ TAPE7, $ TAPE9

ariable Field:

16

File code,LUD or Device name,Multireel indicator,File serial no.,Reel sequence no.,File name,Class,Density

urpose: Used to assign 7-track and 9-track tape units as follows:

$ TAPE assigns either 7-track or 9-track tape unit. If the disposition is R (or null) and no file serial number or file number field is present, either a 7-track or 9-track unit is assigned (depending upon availability). When the disposition is other than R or when a file serial number or file name is present, track requirement is determined by installation option.

$ TAPE7 assigns a 7-track tape unit.

$ TAPE9 assigns a 9-track unit.

In addition, these cards can be used to assign a reserved (named) tape unit and a specified class of tape unit. A named unit takes precedence over a TAPE7/TAPE9 declaration.

As many as four classes of devices can be specified in a hierarchical manner:

3 - Most preferred
2 - Preferred
1 - Ordinary
0 - Subordinate (unclassified)

Density permits overriding a site standard density and is specified as DENx (MTS500, MTU0400, and MTU0500 subsystems only):

DEN2	- 200 bpi
DEN5	- 556 bpi
DEN8	- 800 bpi
DEN9	- As is set
DEN16	- 1600 bpi (9-track only)

ules:

1. One $ TAPE card must be present for each tape file used within the activity.
2. Must follow (not necessarily immediately) the activity definer ($ COBOL, $ EXECUTE, etc.), but not necessarily immediately.

$ TYPE

Variable Field:

16
File code

Purpose: Accesses the online control console (TY1).

NOTE: GCOS always allocates the console file codes T*, *T, T/, and /T. The user need not use $ TYPE for allocation of these files.

Rules:

1. Must follow the activity definer ($ COBOL, $ EXECUTE, etc.), but not necessarily immediately.

$ UPDATE

Variable Field:

16	73	80
Option	Identification (Optional)	

Purpose: Used when supplying alter input to a compiler or assembler.

Option: LIST

Rules:

1. Always precedes all $ ALTER cards for the activity being altered.
2. Must follow the source or COMDK to be altered.

$ USE

Variable Field:

16
Name or Name/Size/

Purpose: Instructs General Loader to enter the names specified in the variable field into its load table as SYMREFs or as labeled common regions. Also used to cause loading of a routine from the subroutine library.

Rules:

1. Must precede $ EXECUTE for the activity.

$ USERID

Variable Field:

16
System master catalog name$Log-on password

Purpose: Tests for the existence of the system master catalog name (known to Time Sharing System as USERID) and a correct log-on password. Prevents unauthorized users from accessing the system.

Rules:

1. Must be present for any job containing $ PRMFL, $ SELECTD, or $ SELECT and must precede $ PRMFL, $ SELECTD or $ SELECT.
2. Normally, $ USERID follows $ IDENT. If more than one $ USERID is present in a job, only the last $ USERID encountered is effective.

$ UTILITY

Variable Field:

16
Options

Purpose: Allows the user to perform file operations (such as file dumps, file compares, etc.) as separate activities via the Utility routine. When System Input encounters $ UTILITY, a U* file is opened. All subsequent Utility control cards are placed on U* up to the next $ control card indicating a new activity or forcing the opening of a new file. U* becomes input for Utility.

Options:

DUMP	NDUMP
NJREST	JREST
NREST	REST

Rules:

1. $ FFILE and/or $ FUTIL must not precede $ UTILITY.

2. Utility reads its control cards and acts on them one at a time. Therefore, $ FUTIL must follow $ FFILE and/or $ QUTIL if the FFILE/QUTIL options are to be in effect for the FUTIL functions.

$ WHEN

Variable Field:

16
Operator, Label

Purpose: Permits conditional branching to a following control label within an activity. Used to permit conditional $ file cards or other controls within an activity.

Operator Symbols:

+	- OR
*	- AND
/	- NOT
blank	- YES

Rules:

1. Must precede the $ Label card referenced.

$ 167PK, $ 170PK, $ 180PK, $ 181PK, $ 190PK,
$ 191PK, $ 310PK, $ 400PK

Variable Field:

16
File code,LUD or Device name,Access mode, Pack number,Pack name,Ownership, Beginning link number/Number of contiguous links

Purpose: Sets up allocation of files on a removable media disk storage subsystem.

Rules:

1. Must follow the activity definer ($ COBOL, $ EXECUTE, etc.), but not necessarily immediately.
2. Removable files do not contain any part of the operating system.
3. Use of pack number 99999 without a pack name causes deletion of the job.
4. Leading and embedded blanks are not permitted in the pack number; trailing blanks are permitted.
5. Requests for named devices are not honored on $ xxxPK cards.

$ 355MAP

Variable Field:

16
Options

Purpose: Calls the DATANET 355/6600 Macro Assembler.

Options:

LSTOU	NLSTOU
DECK	NDECK
NCOMDK	COMDK
GMAC	NGMAC
NDUMP	DUMP
NCOPY	COPY
JREST	NJREST
REST	NREST
	ON5

Rules:

1. 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 355/6600 Macro Assembly program can use the standard $ SNUMB, $ LIMITS, $ ALTER, $ UPDATE, $ ENDJOB, and ***EOF cards.

$ 355SIM

Variable Field:

16
Options

Purpose: Calls the DATANET 355/6600 Simulator.

NOTE: Only absolute programs are simulated. I/O is not simulated for remote communications devices.

Options:

NDUMP	DUMP
JREST	NJREST
REST	NREST

Rules:

1. 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 355/6600 Simulator uses the standard $ SNUMB, $ IDENT, $ ENDJOB, and ***EOF control cards.

TERMINATE MESSAGES/ABORT AND DELETE REASON CODES

CONSOLE OUTPUT ✓

Aborts:

- Aborts caused by hardware faults or GCOS-detected errors are identified by a message text in the form:

  *ABT sssss-aa tt.ttt (abort reason)

  The abort reason is a four-word message issued on the console. The octal code identifying the message is contained in bits 24-35 of the Q-register.

- Aborts caused by errors other than those above (that is, when not detected) are identified by a two-character code at cc in the message:

  *ABT sssss-aa tt.ttt *USER cc

Deletes:

- Deletes caused by an allocation fault that created an unrecoverable situation are identified by the message:

  *TILT, DLT S#sssss

  1. A TILT message usually indicates the system is going down.
  2. If a job is deleted during allocation for subsequent activities after having executed at least one activity, the following message appears:

     *END sssss tt.ttt ACTY DELETE

- Deletes caused by fatal control card errors are identified by the message:

  *DLT S# sssss-aa IMPROPER CONTROL CARDS

EXECUTION REPORT ✓

Aborts:

- Abort message texts and/or reason codes (see "Console Output" concerning codes and messages) appear instead of the NORMAL TERMINATION portion of the termination message following the J* (control card) file listing.

Deletes:

- Messages giving the reasons for System Input deletes are embedded in the J* file listing.
- All other GCOS delete messages appear following the J* listing.

DUMP ANALYZER ✓

AT Abnormal termination caused by STOP or KILL console verb.

BMC

B0 Partitioned record error.

B1 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 card are incorrectly used.

B3 Improper request for transliteration. Either the media combination is in error or codes (IBMF, IBMC, IBMEL, or GE225) are improperly interpreted. Transliteration applies only to card reader, card punch, and magnetic tape units (only card reader and punch for IBMEL); assigning it to any other device causes 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 incorrect; probable hardware failure.

COBOL SYSTEM ✓

C0 A requested input/output file is not contained in the file address table (an internal compiler problem). Notify Honeywell field representative.

C1 Address table (*3 file) is full. Normally, this problem can be corrected by increasing the size of the *3 file with a $ FILE card. The maximum assignable is 106 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 can be corrected by including a $ LIMITS card to extend memory limits, 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 that have been processed prior to this point.

C4 Substantive Stack build error in Report Writer (an internal compiler problem). Notify Honeywell field representative.

C5 Invalid internal list structures of the Report Writer (as internal compiler problem). Notify Honeywell field representative.

C6 Fixed and variable portion of the Report Table is full.

C7 Fixed and variable portion of the Report Table is full.

C8 Invalid internal list structures of the Report Writer Analyzers (an internal compiler problem). Notify Honeywell field representative.

C9 Report Writer Generator is unable to build its COBOL lists (internal language) using the Analyzer build routines (an internal compiler problem). Notify Honeywell field representative.

CA Invalid internal list structures of the Report Writer Generators (an internal compiler problem). Notify Honeywell field representative.

CB Invalid (not 17 or 23 octal) end-of-file mark has been encountered in a COBOL object program.

CC Compiler has developed an internal syntax processing error (an internal compiler problem). Notify Honeywell field representative.

CD Copy module needs more memory. If possible, adjust with $ LIMITS card.

CE Stack overflow in processing source syntax (an internal compiler problem). Notify Honeywell field representative.

CF Attempted RETURN statement without a SORT or MERGE in control.

CG A COBOL object program has attempted one of the following:

1. The value zero exponentiated by the value zero.
2. The value zero exponentiated by a negative value.
3. A negative value exponentiated by a non-integral value.

CH A denial return is received when a MME GESYOT is executed to engage the backdoor file facility.

CI An attempt to engage the backdoor file facility was made when the facility was not included in the system configuration.

CJ Misspelled or missing Procedure Division card.

CL Invalid LIST structures encountered by the internal language analyzers (an internal COBOL problem).

CR I/O incomplete for a random file in a COBOL object program.

CT Attempt to load test monitor dummy link was unsuccessful (an internal COBOL problem).

COBOL - TRANSACTION PROCESSING SYSTEM INTERFACE ✗

CX Content of the OUTPUT-SIZE field specifies a message larger than the message that can be contained in the sending field.

CZ No station currently attempting to access the message.

I-D-S ✗

D2 I-D-S has aborted the program because the accounting buffer is too small for the I-D-S record to be journalized.

DUMP ANALYZER ✗

DR Derail fault caused by dump analyzer.

DV Divide check fault.

EF Nonrecoverable error in data on dump tape.

FT Fault tag modifier.

FILE AND RECORD CONTROL ✓

G1 Blank tape rather than 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.)

GF File and Record Control has aborted program for the cause indicated by its abort message. GF errors usually indicate a program error; the program should not be rerun.

GR File and Record Control has aborted the program for the cause indicated by its abort message. It may be possible to rerun the program successfully.

GF and GR Abort Messages

```
ABORTED BY FILE AND RECORD
          CONTROL ROUTINE __ CODE-FILECODE--
                                        L(FCB) FOR | IS IN XR2
```

BSREC	1	Illegal request for this routine
	2	Status not tape on, load point, or ok
BSTFM	1	Illegal request for this routine
	2	Reached load point, but more files to skip
CLOSE	1	EOF status on output file
	2	Unrecoverable I/O error, no user routine
	3	File to be closed not in chain
	4	Illegal status for mass storage device
	5	Unrecoverable I/O writing EOF mark on tape
FORCE	1	Illegal request for this routine
FSREC	1	File not present
	2	EOF on device from prior command
	3	Impossible return from system routine
	4	Illegal file definition in FCB
	5	Unrecoverable I/O error; no user routine
	6	Illegal request for this routine
	7	I/O status other than blank tape on read
FSTFM	1	Illegal request for this routine
GET	1	File designated as output file
	2	Illegal file definition in FCB
	3	Unrecoverable I/O error; no user routine
	4	Block serial number or block length error
	5	Fixed or mixed record size for a file is zero, or variable record size is zero for tape file
GF200	1	EOF status on output file
	2	Unrecoverable I/O error; no user routine
GF275	1	Unrecoverable I/O writing EOF mark on tape
GR980	1	Tried to create an illegal I/O request
.GXLIT	1	Inconsistent file description
	2	Illegal file code
	3	Invalid card option
	4	Too many transliterated files
	5	I/O error; no recovery attempted
	6	Illegal transliterated device
	7	Block size too large
	8	User buffer too small
	9	Requested parameter not implemented
OPEN	1	Illegal device code for File and Record Control
	2	File is locked
	3	Device is printer or punch, not output file
	4	Illegal mass storage device format
	5	Linked file but not variable record type

	6	Illegal format for SYSOUT file
	7	File designated as required is not present
	8	Two file designators pointing to same file
PUT	1	EOF status on output file
	2	Unrecoverable I/O error; no user routine
	3	Illegal file definition in FCB
	4	Current logical record larger than buffer
PUTSZ	1	Illegal request for this routine
	2	New size larger than old record size
RDREC	1	Binary card not a COMDK card
	2	COMDK cards out of sequence
	3	Data error in decompressing COMDK card
READ	1	Illegal file definition in FCB
REMOT	1	Remote terminal output file has no buffer
REWND	1	Illegal request for this routine
USERR	1	Routine calling error repeated
WAIT	1	Unrecoverable I/O error; no user routine
WEF	1	EOF on output file
	2	Unrecoverable I/O error; no user routine
	3	Illegal request for this routine
	4	Illegal status for tape or mass storage device
WRITE	1	File not present
	2	EOF status on output file
	3	Illegal file definition in FCB

HEALS LOGGING PROGRAM

H0 Abnormal ECFR termination.

H1 No hardcore CPU error record buffers exist.

H2 MME GENEWS request to spawn ECFR was denied because HEALS has spawned eight programs, the SNUMB table is full, the GPOP queue is full, or the requested SNUMB is already in .CRSNB table. (To recover, reexecute HEAL.)

H3 ECF access was denied for reason other than NAME NOT IN MASTER CATALOG after a MME GEMORE to request ECF access. An undetected hardware error or a system software failure occurred. The reason code is in the A-register.

H4 ECF access denied after a MME GEMORE to request ECF access following initial creation of the ECF by HEALS. An undetected hardware error or a system software failure occurred. The reason code is in the A-register.

H6 ECF or ESF is released or is a system device. An undetected hardware error or a system software failure occurred.

H7 ECF physical block size is 40 words. HEAL requires that ECF reside on a device with 64-word physical blocks. HEAL creates the ECF file on ST1, which must reside on a device with a 64-word physical block.

H8 ESF access was denied after a MME GEMORE to request ESF access. An undetected hardware error or a system software failure occurred. The reason code is in the A-register.

H9 ASF access was denied after a MME GEMORE to request ASF access. An undetected hardware error or a system software failure occurred.

HB MPC statistics sampling was requested after the MPC statistics section was released from the HEALS logging program core space.

HD HEALS internal channel descriptor block was full. MPC configuration was too large for the HEALS logging program.

HE Too little core space was reserved for MPC device statistics. MPC configuration was too large for HEALS logging program.

HF Configuration change caused device statistics space requirements to overflow allocated core space.

HG Error occurred during MPC Read/Write memory scan.

HH Error occurred during scan to print MPC device statistics.

HI Illogical condition was detected internally. An undetected hardware error or a system software failure occurred.

HJ A cache memory error occurred after cache memory disable was attempted.

HK HEALS request (MME GEFSYE) to create HEALS master catalog was denied. The reason code is in the A-register.

HL HEALS request (MME GEFSYE) to create ECF was denied. The reason code is in the A-register.

HM HEALS request (MME GEFSYE) to create ESF was denied. The reason code is in the A-register.

HN HEALS request (MME GEFSYE) to create ASF was denied. The reason code is in the A-register.

HEALS REPORTER PROGRAM

H9 ECFR has waited at least three minutes for HEAL to assign ECFR a partial ECF segment.

I-D-S SYSTEM

ID I-D-S has aborted program for cause indicated by its abort message.

JOURNAL PROGRAM (XJRNAL)

J1 No parameter cards found in $ DATA file ".X"

J2 "BEFORE" and "AFTER" parameters both found in $ DATA file ".X"

J3 Parameter card in $ DATA file not an ISP parameter card

J4	Too many ISP data files (maximum is 20)
J5	ISP parameter card not recognized as either "BEFORE" or "AFTER"
J6	Journal tape "T1" format not correct, or tape read error
J7	Request for page buffer memory space not granted User should increase memory size via $ LIMITS card and rerun job
J8	Journal tape read error not corrected with reread
J9	Too many ISP data files, or too many volumes in multivolume files (maximum is 20)
JA	Journal tape format (or content) error
JB	Request for bit table memory space not granted – user should increase memory size via $ LIMITS card and rerun job
JC	Could not find correct file code for a volume in a multivolume file

FORTRAN SYSTEM X

LK	No $ ENTRY card for this link

GENERAL LOADER PROGRAM ✓

L1	Missing subroutine required
L2	NOGO option exercised, or program and Loader overlap
L3	Fatal error encountered during loading
L4	General Loader I/O error

DUMP ANALYZER X

ME	Memory fault
NT	Tape with requested density not available

FILE AND RECORD CONTROL ✓

NT	Attempt to access teleprinter through File and Record Control without having loaded proper routine; .RTYP option entered as SYMREF on the $ USE card

DUMP ANALYZER X

OK	User requested program abort of .MMDMP via $ SET 30 card
OP	Illegal operation code or command fault in .MMDMP code
OV	Overflow fault in .MMDMP code

GMAP COMPILER ✓

P0 Maximum of MACRO expansion (63 levels) exceeded

FORTRAN SYSTEM ✗

Q1 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

Q5 FORTRAN Execution Error Monitor (FXEM) told to take an alternate return but an alternate return name was not supplied

Q6 Termination of object program execution via FXEM

ALGOL SYSTEM ✗

Q3 Logical Unit 6 not present

QA Recursive call to the error processor (i.e., an error occurred while trying to output an error message)

SORT ✓

SC SORT abort; reason indicated on execution report

SYSTEM EDITOR ✗

SE System Editor encountered an irrecoverable error

SO FILEDIT encountered an irrecoverable error

COBOL SYSTEM ✗

SM COBOL Linkage not in stack

UTILITY ✓

U1 Control deck error

U2 Comparison error

U3 Hardware error

JOVIAL ✗

V0 Compiler error abort: fatal error encountered when processing direct code.

V1 Compiler error abort: alternate indirect statement construction probably will allow it to compile.

V2 Object program abort: JOVIAL compiler generates a MME GEBORT when replacing an erroneous JOVIAL statement.

V3 Compiler memory exhausted abort. Compiler needs more storage, but requests for additional memory have been denied. Additional space is not available to continue compilation.

V4 Compile activity completed and preparation for loader activity (binding the object H*) aborted because not enough core is available to satisfy CORE specification on RUN command. H* file will not be bound; but if on RUN command a request was made to save C* file, the C file for the object JOVIAL program will be created. Object program will not be placed in correction.

OPERATOR-INITIATED ABORTS ✓

X1 Operator deleted job from control stack.

X2 Operator aborted job in execution.

FORTRAN ✗

The abort code Y1 is always displayed as the reason code for any abort. The panel gives the specific reason code in the upper 18 bits of the Q-register.

Y1 (X1) Compiler space management module has unsuccessfully attempted to allocate contiguous core block for internal table. Rerun with DUMP option and $ SYSOUT card for file code *F. Return dump to Honeywell Field Support - PCO.

Y1 (X2) Compiler has attempted to execute request for additional core space more than 10 consecutive times (initial core space plus maximum of 30K). Increase allocation via $ LIMITS card or via "CORE=" option on TS RUN.

Y1 (X3) GCOS has denied compiler request for additional core space for internal tables. Increase allocation via $ LIMITS card or via "CORE=" option on TS RUN.

Y1 (03) Expression being handled has tree structure depth greater than 64. Expression must be divided.

Y1 (04) Rerun with DUMP option and $ SYSOUT card for file code *F. Return dump to Honeywell Field Support - PCO.

Y1 (P4) Unrecoverable error occurred in code generator; error message will print following source statement causing abort. Rerun with DUMP option and $ SYSOUT card for file code *F. Return dump to Honeywell Field Support - PCO.

REMOTE TERMINAL SUPERVISOR (GRTS)

41	Invalid I/O status pointer in MME GEROUT
43	Invalid DCW pointer or illegal character count
72	Invalid MME GEROUT operation code
73	PAT area exhausted
74	Output record too large for terminal type
75	Undefined remote terminal station identification or duplicate remote inquiry name
76	Invalid MME GEROUT program identification or illegal program number

SYSTEM ABORT MESSAGE TEXTS

00	No reason specified
01	Cannot load SSA module
02	SSA module checksum error
03	Cannot find SSA module
04	Too many pop-ups of SSA
05	I8-run time exhausted
06	I2-MME GEROAD in CC (courtesy call)
07	I6-GEENDC but not in CC
10	Primary mailbox error
11	Connect or TI error
15	MEMORY PARITY DURING I/O
20	Memory time-out in I/O
22	F0-memory address fault
23	I1-improper MME address
24	F1-fault tag fault
25	F2-command fault
26	F3-derail fault
27	F4-lockup fault
30	F6-memory parity fault
31	F7-undefined OP fault
32	F9-overflow/underflow
33	I0-divide check fault
34	F8-OP not complete fault
35	I/O out of memory bounds
36	I3-file code not defined
37	I5-no GEINOS filepointer
40	I7-access beyond file
41	K2-bad status return pointer
42	K3-invalid file pointer
43	K4-invalid DCW pointer

44	K5-bad courtesy call PTR
45	K6-bad I/O command-file
46	K7-two successive TDCWS
47	K1-invalid I/O on device
50	X2-operator term request
51	X2-operator kill request
52	M4-N4-I/O lim. Call/Save/RSTR
53	M6/N7-I/O err. Call/Save/RSTR
54	M5-no pat for Call/Save/RSTR
55	M1/N5-bad dev. Call/Save/RSTR
56	Non-random GECALL/GESAVE/GERSTR file
57	N8-GESAVE file is full
60	M6-Call/Save/RSTR checksum
61	Low GESAVE/GERSTR origin used
62	N4-zero GESAVE word count
63	M2/M3-Call/RSTR name missing
64	N8-Call-out of file span
65	Improper GECOS call
66	GEMORE - zero filecode
67	M0-bad GEMORE parameter
70	No room in PAT - GEMORE
71	M7-not disk/drum GEMORE
72	R1-bad GEROUT operation code
73	R4-GEROUT PAT exhausted
74	RMT terminal record size
75	Undefined remote station ID
76	Invalid GEROUT program ID
77	0 - output limit exceeded
100	SYSOUT record size error
101	SYSOUT seek error
102	SYSOUT allocation error
103	01-bad SYOT status pointer
104	01-bad SYOT buffer pointer
105	01-SYOT buffer limits
106	EP-irrecoverable I/O error
107	K4-invalid seek DCW I/O
110	Bad GESYOT media code
111	N3-rollback not possible
112	TSS requested term
113	GEMORE duplicate file
114	Invalid I/O SCT pointer
115	Inadmissible perm write
116	Inadmissible perm read
117	D2-IDS record too long
120	SYSOUT storage exhausted

121	Invalid MME parameter
122	Lost extra SSA contents
123	J3-ALLOCATOR deleted job
124	Bad buffer address-GENEWS
125	Guessing on PRMFL GEMORE
126	No USERID-PRMFL GEMORE
127	X2 operator stop request
130	Cannot move job in core
131	IOM CHAN unexpected PCW
132	IOM CHAN improper instruction
133	IOM CHAN improper DCW
134	IOM CENT IPW tally RNOUT
135	IOM CENT two TDCWS
136	IOM CENT boundary error
140	IOM CENT IDCW res mode
141	IOM CENT char POS/SZ dis
142	Backdoor F.C. unknown
143	Invalid file to backdoor
144	IDS address out of bound
145	SECUR-SCC invalid
146	File allocation error
147	SECUR-SCC inval for user
150	SECUR-SCC inval for line
151	SECUR-DUPLICATE report
152	SECUR-INVALID DCW string
153	SECUR-INVALID funct code
154	No links for swap file
155	Data access system abort
156	Data management system protection failure
157	BSS (Bulk Store Subsystem) main memory boundary error
160	BSS no system response
161	BSS connect halt
162	ISP system termination
163	Illegal EIS data

APPENDIX A

FILE CODES

DESCRIPTION

File codes are two-character alphanumeric codes assigned by the programmer. A slash (/) or comma (,) should not be used. An asterisk (*) should be used only when referring to a system file.

STANDARD SYSTEM FILE CODES

A*	-	Alter file
B*	-	Object program file
*B	-	DATANET 355/6600 simulator input file
C*	-	Binary deck file
*C	-	Editor input file
D*	-	Stranger option file (directives file)
E*	-	Extension editor file
*F	-	FORTRAN dump file
F*	-	Intermediate file, editor
G*	-	GMAP source file
H*	-	Program link file
I*	-	Data storage file
*J	-	Job subfiles
J*	-	Intermediate file, source library editor
K*	-	Compressed deck file, new source library file
*K	-	Source library editor intermediate file
L*	-	System subroutine library file
*L	-	Secondary subroutine library file
M*	-	Old source library editor
O*	-	Current system library file
P*	-	System output file
*P	-	Production library tape file
Q*	-	System-loadable file
*Q	-	Master input file to system library editor
R*	-	Loader input file (new object library)
*R	-	Old object master file
S*	-	Compiler source file
*S	-	I-D-S source file
T*	-	Console message file (for magnetic tape)
*T	-	Console message file (for unit record device)
U*	-	Utility file
X*	-	SYSEDIT file
*Z	-	Intermediate file, source library editor
*1	-	Intermediate file, GMAP, FORTRAN, I-D-S, JOVIAL, and COBOL
*2	-	Intermediate file, ALGOL, JOVIAL, COBOL
*3	-	Intermediate file, I-D-S, COBOL
*4	-	Intermediate file, object library editor
*5	-	Intermediate file, source and object editor
*6	-	COBOL intermediate file
*7	-	Editor intermediate file (punch card file)
**	-	Dynamic system-loadable file

APPENDIX B

LOGICAL UNIT DESIGNATOR

DESCRIPTION

A logical unit designator (LUD) is a two- or three-character symbol (followed by a disposition code) identifying a file. The LUD is required when a file is to be saved for use in a subsequent activity in a job or when a specific peripheral device has been named and is required for the current activity.

```
1      8       16
$      FORTY
$      TAPE    P*,A1S
        .
        .
        .
$      COBOL
$      TAPE    CR,A1R
```

The LUD also is used to indicate those files that are to be purged (overwritten) for security purposes or to indicate removable media files that are to be dismounted and removed from the computer system.

```
1      8       16
$      TAPE    CR,A1P
```

The first character of the identifier can be alphabetic or numeric. The second character must be numeric. These characters are followed by one or two single-character codes indicating the desired disposition of the file at end of activity or file release (whichever is first).

DISPOSITION CODES

For all types of storage media:

R - Release file (implied if null)
S - Save file for subsequent activity
P - Overwrite file contents and release file

For magnetic tape and disk pack files only:

D - Dismount media from system
C - Write inhibit media and save for subsequent activity.

For example:

A2DR - Release file A2 normally and dismount media from system; if activity aborts, release file.

X1S - Save file X1 normally if activity aborts.

Files are purged only if they have Write permission. In the case of cataloged files, the user must be the creator of the file. Files with no I/O activity during the job are not purged.

[illegible]

The first character of the identifier can [illegible] the second character must be [illegible]

[illegible]

[illegible] tape and [illegible]

[illegible] from system [illegible]

[illegible] is the same [illegible]

[illegible]

Reference [illegible]

[illegible]

APPENDIX C

DEVICE NAME

DESCRIPTION

A device name is a three-character symbol identifying a specific peripheral device required for use. Any device referenced in this manner should be configured at startup time. (If the device is not configured, the operator can enter a device name at the console.)

Device names on $ file control cards must be consistent with the type of device named (i.e., the name of a mass storage device cannot be used on a $ TAPE card.)

The device name can be followed by a disposition code.

NAMING CONVENTIONS

Naming of a specific peripheral device must conform to the following conventions:

- Each name comprises three characters: the first character must not be zero; the second character must be alphabetic; and the third character can be numeric or alphabetic.
- Each name must be unique.
- A device can have more than one name, but one name cannot be assigned to more than one device.

For example:

MTAS - MTA is the device name; save file for subsequent activity.
MT2D - MT2 is the device name; dismount storage media from system.

3.395180 mhz

3395.18 khz

Honeywell

294

He

SER
SER

SU

SP

DATE:

May 1974

ORDER NUMBER: