Honeywell

# CONTROL CARDS AND ABORT CODES POCKET GUIDE

SERIES 600/6000

GCOS

SOFTWARE

**Honeywell**

# CONTROL CARDS AND ABORT CODES POCKET GUIDE

**SERIES 600/6000**

**GCOS**

SUBJECT:

Descriptions of Control Cards for Series 600 and Series 6000 Programming.

SPECIAL INSTRUCTIONS:

This pocket guide, Order Number BJ69, Rev. 2, supersedes BJ69, Rev. 1. The control card descriptions are in alphabetical order by card name.

DATE:

January 1973

ORDER NUMBER:

BJ69, Rev. 2

User's Name ______________________________

# PREFACE

This book is an abbreviated pocket reference to the control cards used in Honeywell Series 600/6000 Systems. A complete description of the control cards and their uses may be found in the following reference manuals:

Control Cards Reference Manual - BS19
(formerly CPB-1688)
General Comprehensive Operating Supervisor (GCOS) - BR43 (formerly CPB-1518)

The control cards appear in alphabetical order within this guide.

Three appendices also are provided:

Appendix A - Terminate Messages/Abort and Delete Reason Codes
Appendix B - File Codes
Appendix C - Logical Unit Designator (LUD)

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

Other related Series 600/6000 manuals referred to in this guide include:

File System, BR38 (formerly CPB-1513)
COBOL User's Guide, BS09 (formerly CPB-1653)
Termination and System Output SMD, BR25 (formerly CPB-1496)
Integrated Data Store, BR69 (formerly CPB-1565)
General Loader, BN90 (formerly CPB-1008)

File No.: 1613, 1713

## CONTENTS

CONTENTS (cont)

## CONTROL CARDS

### Basic Control Card Rules

Control cards are used for many different functions. However, certain basic rules are applicable to all cards:

1. All control cards are identified by a symbol in column 1.

2. The control card name normally (i.e., SNUMB, IDENT, etc.) begins in column 8. ($ Label may begin in columns 3-7.)

3. The variable field begins in column 16; must be preceded by a blank in column 15; and, may not exceed column 72(except where noted).

4. Variables must be separated by commas.

5. A blank terminates the variable field definition. Therefore, no imbeded blanks are permitted. (A dash may be used as a spacer.)

   Exceptions: $ COMMENT, $ IDENT, $ FORM, $ MSGn and $ MSG3. See specific description pertinent to each of these cards.

6. Variables exceeding 12 characters are truncated except on those cards noted in Rule 5.

7. Unless otherwise indicated for the specific control card, options may be listed in the variable field in any order.

8. If options are allowed, but not specified, GCOS uses standard options (underlined).

9. No more than six variables are allowed in the operand field, except where noted.

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 Appendix A - Abort Codes.

---

## $ 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. $ ALTER cards not following this convention are printed on the execution report as alter file errors.

---

$ 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 may 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.

## $ 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 |
| | SYMTAB |
| | SEGMNT |
| | NOXREF |
| | LIBCPY |
| | ON6 |

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. However, the LIBCPY option takes precedence over the COPY option.

## $ COMMENT

Variable Field:

16 72

Any Comments

Purpose: Used by the programmer to communicate to the operator via the on-line console.

NOTE: It is suggested that $ MSGn, rather than $ COMMENT be used whenever possible.

Rules:

1. If the activity being affected is the first activity in the job, $ COMMENT may precede or follow the activity. $ COMMENT cards affecting any subsequent activities in the job must follow those activities.

2. Only one $ COMMENT card may be used with each activity.

3. The comment may not 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

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 (by General Loader) 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 (not necessarily immediately) the activity definer ($ COBOL, $ EXECUTE, etc.).

NOTE: X = any character appended to SNUMB and used as inquiry name.

## $ DATA

Variable Field:

| 16 |
|---|
| File Code,Options |

Purpose: Writes files onto temporary linked disk or drum for input to a user activity. Files created in this manner may not 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 |
| | TAKEc |
| | CKSEQ |
| | ENDFC |

Rules:

1. Must immediately precede data cards to be put onto the specified file.
2. Must follow (not necessarily immediately) the activity definition card for the activity in which the file is to be read.
3. Absence of a file code results in a warning message on the job execution report and the default assignment of the file to I*.
4. If the COPY option is used, $ ENDCOPY must follow the last data card.

## $ DKEND

Variable Field:

| 16 | 67 | 73 |
|---|---|---|
| Normally Not Used | TTL Date (or NO TTL if no TTL card) | 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 new job 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 will 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: File code specifies the file being built from data preceding $ ENDCOPY. Ends the copy function of $ DATA.

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, $ INCODE or $ ENDCOPY card 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 were detected. Errors are noted on the console typewriter 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 the Startup deck and, when detected, the module name table is examined to determine 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. Is the last card in the Startup deck.

## $ EQUATE

Variable Field:

16

SYMDEF Names

Purpose: Defines new SYMDEF location 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 (/) may 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 null, switches assumed off; NDUMP implied.

Sense Switches: Turned on when Assembler/Compiler options are specified.

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

$ 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 saved 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 re-editing the software library program. Upon encountering $ EXTEND, System Input sets the extend bit (bit 16 in the program switch word) to 1 (ON).

Options: JREST NJREST
REST 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

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

Parameters:

STDLBL
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 $ EXECUTE when used with a General Loader activity ($ OPTION must be included); $ UTILITY when used with a Utility activity; and, $ 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 disk and drum files.

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

NOTE: See Appendix B - 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 and replaces $ DSPK, $ DISC, $ MASS or $ DRUM cards. If no device type is specified or if the requested device is not configured, allocation is on the "fastest" device available.

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

Permissible device types are:

DSS167
DSS170
DSS180
DSS204
DSS270
MSS800
MDS200
DSS181 (Series 6000 only)
DSS190 (Series 6000 only)

A new cataloged file may be created via:

16

File Code, LUD, Size, NEW, Filename

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

Rules:

1. Must follow (not necessarily immediately) activity definer ($ COBOL, $ EXECUTE, etc. ).
2. When no disposition code is indicated, R (release) is assumed.
3. If number of links 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 C - LUD/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 |
| 635R | 655F |
| 6000F | IOCF |
| IOMF | 355F |
| SAVEB | |

NOTE: If no object computer options are specified, GCOS assumes the object computer is the same as the computer doing the assembly.

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.

$ 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 21 characters) from $ FORM is typed on the console typewriter.

Rules:

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

---

## $ FORTA

Variable Field:

16

Options

Purpose: Calls the FORTRAN IV compiler.

Options:

| | |
|---|---|
| LSTIN | NLSTIN |
| NLSTOU | LSTOU |
| DECK | NDECK |
| NCOMDK | COMDK |
| NSTAB | STAB |
| 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.

## $ FORTY ($ FORTRAN)

Variable Field:

16

Options

Purpose: Calls the Series 6000 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; must precede any other control card associated with that activity. May 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, FC2/ |
| ADUMP/M, N/ | RPT /mP, nT/ |
| RSAVE/1F/ | AADUMP/M, N/ |
| RREST/1F/ | RCOPY/1F/ |

NOTE: See Appendix B - 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 of a 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 which is preceded by $ FUTIL.

$ 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 |
| | SYMTAB |
| 635F | 655F |
| 6000F | IOCF |
| IOMF | 355F |
| EISF | NEISF |

NOTE: If no object computer options are specified, GCOS assumes the object computer is the same as the computer doing the assembly.

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: Causes 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 may not exceed six characters.

2. $ GOTO may skip around $ USERID and $ PRMFL but before the job is begun a test for user ID validity and PRMFL file presence is made. Nonexistent user ID 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 complete.

$ IDENT

Variable Field:

16

Account Number,Identification

Purpose: Identifies the user of a job or activity and supplies accounting information. Each activity may be preceded by $ IDENT; or, $ IDENT may 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 and can be alphabetic, numeric or alphanumeric.
3. Columns 16-72 are available during activity execution in cells 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 |
| | SEGMNT |
| | NOXREF |
| | LIBCPY |

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 which 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. May not be continued by $ ETC; successive $ IF cards may be used.

---

## $ INCODE

Variable Field:

16

BCD Card Code

Purpose: Indicates the following BCD input cards must undergo character transliteration from the character set specified on $ INCODE to the internal binary representation of the Series 600 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 PERM file.
3. Will not perform transliteration on a $ DATA, , COPY. . . $ ENDCOPY set.
4. May not precede a COMDK.

Card Codes:

| | |
|---|---|
| IBMF | - IBM FORTRAN set |
| IBMC | - IBM COBOL set |
| GE225 | - GE225 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:<br>or<br>Label. |

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

Rules:

1. Should follow $ GOTO, $ IF or $ WHEN that reference the label.
2. The label may 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 prior to 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 files codes is allowed on $ LIBRARY.

NOTE: See Appendix B - 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 exceeding six characters and must not be a SYMDEF.
3. May not be used with the COBOL Segmentation Loader ( CLOAD).

$ LIMITS

Variable Field:

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 | 0.08 | 6k | 1,000 |
| COBOL | 0.15 | 32k | 20,000 |
| EXECUTE | 0.05 | 16k | 5,000 |
| EXTEDIT/ FILEDIT | 0.04 | 32k | 10,000 |
| FORTA | 0.05 | 25k | 12,000 |
| FORTY/FORTRAN | 0.05 | 24k | 12,000 |
| FILSYS | 0.03 | 32k | 1,000 |
| GMAP | 0.04 | 24k | 10,000 |
| IDS | 0.15 | 32k | 20,000 |
| JOVIAL | 0.08 | 28k | 10,000 |
| PROGRAM | 0.05 | 16k | 5,000 |
| PRODUCT | 0.50 | 20k | 30,000 |
| SYSEDIT | 0.05 | 32k | 10,000 |
| UTILITY | 0.03 | 9k | 10,000 |
| 355MAP | 0.04 | 32k | 10,000 |
| 1401SIM | 0.08 | 12k | 5,000 |
| 225SIM | 0.08 | 24k | 5,000 |
| 44SIM/94SIM | 0.08 | 48k | 5,000 |
| 355SIM | 0.08 | 24k | 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.

   Need not precede control cards which have nothing to do with assignment of space in memory (i. e., $ ENTRY, $OPTION, etc.).

---

$ MSGn

Variable Field:

16

C, Message Text

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

C = Number (1, 2, 3 or 4) corresponding to TY1, TY2, TY3 and TY4. The message text follows.

Rules:

1. MSG1 may appear anywhere in the job.
2. MSG2 should follow the activity definer ($ COBOL, $ EXECUTE, etc.).

---

$ MSG3

Variable Field:

16

Hold Until Date/Hold Until Time

Purpose: Allows user to hold a job in the system scheduler until a specified date/time.

Rules:

1. Hold Until Date format is mmddyy (mm = month; dd = day; yy = year).
2. Hold Until Time may 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. May appear anywhere in job.

$ NLOAD

Variable Field:

16

SYMREFs (Maximum 10)

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

Rules:

1. Can be placed anywhere in the link except in a binary deck.
2. The standard number of SYMREFs in the operand is 10.
3. If the same subroutines are not to be loaded in two or more links, $ NLOAD must be included within each link.

---

$ 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.

## $ NTAPE

Variable Field:

| 16 |
|---|
| File Code, Channel Designator, No. 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 (not necessarily immediately) the activity definer ($ COBOL, $ EXECUTE, etc.).
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 Appendices B and C.

---

## $ OBJECT

Variable Field:

| 16 | 60 | 61 | 67 | 73 80 |
|---|---|---|---|---|
| Remarks and/or Sequence Option | ↑ | Time of Assembly | Date of Assembly | Program Ident. Number |

Source Identification

Purpose: 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. (If assembler encounters Model 655 op code in source deck, $ OBJECT5 is punched.)

Options: SEQ CKSEQ<br>NSEQ

Rules:

1. Must be the first card of the object deck. If the next $ control card after $ OBJECT is not $ DKEND, $ DKEND is presumed missing.
2. When $ OBJECT is used to define the beginning of the DSS180 controller deck in Startup, columns 35-37 contain the identification CON (control deck) or DAT (data deck).

## $ OPTION

Variable Field:

16

Options

Purpose: Alters the standard loader options during loading. Options must be separated by commas.

Options:

MAP
CONGO
NOGO
NOSREF
LOCOMN
NOSETU
NOFCB
ALGOL
FORTRAN
SAVE /Name
NOMSUB

NOMAP
GO
SET/n/
SYMREF
ERCNT/n/
FCB
COBOL
JOVIAL
SAVOLD/Name
RELMEM

Rules:

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.

## $ PARAM

Variable Field:

16

Options

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

Rules:

1. Use of $ PARAM is not restricted to $ SELECT files and may be used in any control deck setup when desired.

2. The $ COMMENT, $ IDENT, $ MSG1, $ MSG2, $ MSG3, $ SELECT, $ SNUMB and $ USERID cards cannot be modified by $ PARAM.

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

4. Must precede the first replacement operator (#n) used, but multiple $ PARAM cards may be interspersed with $ control cards to provide more than nine parameters.

$ PPTP ($ PPTR) or Device Name

Variable Field:

16

File Code, LUD or Device Name

NOTE: See Appendices B and C.

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

Rules:

1. The file code indicated is the same as the file code which 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, Train #

NOTE: See Appendices B and C.

Purpose: Allocates a high-speed printer as follows:

| FIELD | ALLOCATION |
|---|---|
| null | PRT201, PRT203, PRT300 or PRT303 |
| PRT201 | PRT201 |
| PRT203 | PRT203 or PRT201 |
| PRT300 | PRT300 or PRT303 |
| PRT303 | PRT303 or PRT300 |

Rules:

1. A file code must be present.
2. The train number must be numeric.
3. The second character of the device name must be nonnumeric.
4. Must follow (not necessarily immediately) the activity definer ($ COBOL, $ EXECUTE, etc.).
5. The Train # field is used only in conjunction with PRT300 and PRT303. The number must be numeric; if not, the job is terminated.
6. If the user does not designate a standard site train number on the $ INFO card at Startup, the default train number 2 is assumed.

## $ 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 (not necessarily immediately) the activity definer ($ COBOL, $ EXECUTE, etc.).

---

## $ PRMFL

Variable Field:

16

File Code, Permit, Mode, File String
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 $, / and, (comma) are delimiters and must not be imbedded in a $ PRMFL field.
4. Must follow (not necessarily immediately) activity definer ($ COBOL, $ EXECUTE, etc.).

NOTE: See Appendix C - LUD/Device Name

## $ PRODUCT

Variable Field:

16

Name, FCB Option, Assembler/Compiler Options

Purpose: Retrieves and initiates execution of programs from a production library.

Options:

| | |
|---|---|
| LSTIN | NLSTIN |
| NLSTOU | LSTOU |
| DECK | NDECK |
| NCOMDK | COMDK |
| NDUMP | DUMP |
| NCOPY | COPY |
| JREST | NJREST |
| REST | NREST |
| | SYMTAB |

Rules:

1. $ PRODUCT replaces $ OBJECT, $ DKEND and $ EXECUTE cards in basic input deck.
2. No imbedded blanks are allowed.

---

## $ PROGRAM

Variable Field:

16

Name, Options

Purpose: Calls a program previously edited onto the Software Library (such as GMAP, FORTRAN and General Loader) and for which no call capability exists. Standard GCOS limits (16k) are assumed. File control cards following $ PROGRAM are put on R*. Output files P* and T* also are included in the slave PAT Table.

Options: Can be any options currently available with $ EXECUTE, $ FORTA, $ COBOL, $ 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. Also may allocate a named device. Allocation is as follows:

| FIELD | ALLOCATION |
|---|---|
| null | CPZ100, CPZ200, CPZ201, CPZ300 or CPZ301 |
| CPZ100 | CPZ100, CPZ200, CPZ201 |
| CPZ200 | CPZ200, CPZ201, CPZ100 |
| CPZ201 | CPZ201, CPZ100, CPZ200 |
| CPZ300 | CPZ300 or CPZ301 |
| CPZ301 | CPZ301 or CPZ300 |

Rules:

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

---

$ QUTIL

Variable Field:

16

Options

Purpose: Specifies Utility processing options.

Options:

| | |
|---|---|
| EOF/ALL | ASIS |
| TERM | CMPERR/n |
| NBYPSS | USE/n (or IGNORE/n) |
| RESEQ | USER |

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,Name,Reader

Purpose: Allocates the card reader and a named device (if desired) as follows:

| FIELD | ALLOCATION |
|---|---|
| null | CRZ200, CRZ201 or CRZ301 |
| CRZ200 | CRZ200 or CRZ201 |
| CRZ201 | CRZ201 or CRZ200 |
| CRZ301 | CRZ301 |

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

---

$ 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 up to nine output files to a remote station for use by Network Processing Supervisor (NPS), formerly 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 may use $ REMOTE to send output to a remote terminal.
3. Must follow (not necessarily immediately) the activity definer ($ COBOL, $EXECUTE, etc. ).

$ REPORT

Variable Field:

16

Report Code, Device Code, Paper Form #, Ply #

Purpose: Used by SYSOUT to change the standard paper on the printer to a special form; or, to redirect output from the device normally used to a different device (i.e., from standard to optional printer).

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 maximum of six $ REPORT cards may be used in a single activity.
3. Must follow (not necessarily immediately) the activity definer ($ COBOL, $ EXECUTE, etc.).

---

$ 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 have identified himself to the file system by $ USERID.
2. $ SNUMB cannot be contained with a Select file.
3. If no file content has been established, $ SELECT may not be used.
4. Control card "nesting" (use of $ SELECT within a Select activity) is permitted to a depth of 10.
5. $ SELECT may appear anyplace in the deck (following $ SNUMB).

$ SET

Variable Field:

16 ______________________________________________

Bit Numbers to be Set or Reset in Program Switch Word

Purpose: May be used to set or reset bits in the program switch word external to activity execution. These bits then may be tested by a MME GESETS or a MME GERETS. They also may 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 zeroes and must not contain a comma, $, /, : or a period. Detection of any of these causes the job to be ignored. The job identifier can be alphabetic, numeric or alphanumeric.
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 a subprogram was generated on B* and 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; precedes any control card associated with that activity.

---

## $ SYSOUT

Variable Field:

16

File Code

Purpose: 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.

Rules:

1. Used only for files recorded in standard system format with appropriate media and report codes included in logical record.
2. Used for files written through File and Record Control.
3. There is no restriction on the number of files assigned to SYSOUT. However, only nine report codes are output.
4. Must follow (not necessarily immediately) activity definer ($ COBOL, $ EXECUTE, etc.).

$ TAPE, $ TAPE7, $ TAPE9

Variable Field:

16

File Code, LUD or Device Name, Multireel Indicator, File Serial No., Reel Sequence No., File Name, Class, Density

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

$ TAPE - Assigns a tape unit, which may be either 7-track or 9-track. 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 may 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 of a site standard density and is specified as DENx:

| | |
|---|---|
| DEN2 | - 200 BPI |
| DEN5 | - 556 BPI |
| DEN8 | - 800 BPI |
| DEN9 | - As is set |
| DEN16 | - 1600 BPI |

Rules:

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.).

## $ TYPE

Variable Field:

| 16 |
|---|
| File Code |

Purpose: Accesses the on-line console typewriter TY1.

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

Rules:

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

## $ UPDATE

Variable Field:

| 16 | 73 | 80 |
|---|---|---|
| Option | Identification<br>(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 as USERID) and a correct log-on-password. Prevents unauthorized users from accessing the system.

Rules:

1. Must be present for any job containing a $ PRMFL or $ SELECT and must precede the $ PRMFL 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 a separate activity by means of a 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 $ FILE 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

Variable Field:

16

File Code, LUD or Device Name, Access Mode, Pack Number, Pack Name, Ownership, Beginning Link Number/Number of Contiguous Links

Purpose: Set up allocation of files on a removable media DSS167, DSS170, DSS180, DSS181 or DSS190. Also may be used to assign files on a reserved (or dedicated) unit.

Rules:

1. Must follow (not necessarily immediately) the activity definer ($ COBOL, $ EXECUTE, etc.).
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 imbedded blanks are not permitted in the pack number; trailing blanks are permitted.

$ 355MAP

Variable Field:

16

Options

Purpose: Calls the DATANET 355 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 Macro Assembly program may use the standard GCOS $ SNUMB, $ LIMITS, $ ALTER, $ UPDATE, $ ENDJOB and ***EOF cards.

---

$ 94OPTN

Variable Field:

16

S, D, T

Purpose: Used by the IBM 7094 Simulator to provide information concerning startup and debug dumps from the simulator.

Rules:

1. Must follow (but not immediately) $ 94SIM.

2. Start (S) specifications: Blank or LDTP, MANUAL, ZERO and LDCD; I/O debug (D) specifications: Blank and IODBG/xx, . . . /; and trace (T) specifications: Blank, TRACE/xxxxxx/ and TRACE/xxxxxx/yyyyyy.

$ 94PRT

Variable Field:

16

C, SD

Purpose: Used by the IBM 7094 Simulator to describe the disposition of an on-line printer.

Rules:

1. Must follow (but not immediately) $ 94SIM.
2. Channel (C) numbers are A-H; simulation device (SD) codes are blank or TYPE for console typewriter and SYSOUT for SYSOUT file.

---

$ 44SIM

Variable Field:

16

Options

Purpose: Calls the Series 600 IBM 7044 Simulator and causes the 44SIM program to be loaded into memory from the system file.

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. If loading the simulation activity from an input device rather than from the system file, the $ 9SA card must be used following $ EXECUTE.

$ 94SIM

Variable Field:

16

Options

Purpose: Calls the Series 600 IBM 7094 Simulator.

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. If loading the simulation activity from an input device rather than the system file, the $ 9SA control card must follow $ EXECUTE.

---

$ 225SIM

Variable Field:

16

Options:

Purpose: Calls the G-225 Simulator.

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.

---

$ 1401SIM

Variable Field:

16

Options

Purpose: Calls the Series 600 IBM 1401 Simulator.

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.

$ 355SIM

Variable Field:

16

Options

Purpose: Calls the DATANET 355 Simulator.

NOTE: Only absolute programs are simulated. I/O is not simulated for any device.

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 Simulator uses the standard GCOS $ SNUMB, $ IDENT, $ ENDJOB and ***EOF control cards.

---

$ 94SYSO

Variable Field:

16

F, M, S, P, Ct

F - File code.<br>
M - Maximum record size.<br>
S - Simulation mode (blank or R, regular mode; F, fast mode).<br>
P - Print chain (blank or F, FORTRAN; C, standard Series 600/6000).<br>
Ct - DCW count.

Purpose: Used by the IBM 7094 Simulator to obtain printed output directly from SYSOUT without a separate tape-to-printer activity.

Rules: 1. Must follow (but not immediately) $ 94SIM.

# APPENDIX A

## TERMINATE MESSAGES/ABORT AND DELETE REASON CODES

### CONSOLE TYPEWRITER OUTPUT

ABORTS:

- When caused by hardware faults or detected errors, aborts are identified by a message text in the form:

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

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

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

  *ABT sssss-aa  tt.ttt  *USER cc

DELETES:

- When because of an allocation fault that has caused an unrecoverable situation, deletes are identified by the message:

  *TILT, DLT S#sssss

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

     *END sssss  tt.ttt

- When because of fatal control card errors, deletes are identified by the message:

  *DLT sssss-aa  IMPROPER CONTROL CARDS

### EXECUTION REPORT

ABORTS:

- Abort message texts and/or reason codes (see discussion under "Console Typewriter" concerning codes and messages) appear in place 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 imbedded in the J* file listing.
- All other GCOS delete messages appear following the J* listing.

BMC

B0 Partition 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 erroneous; probable hardware failure.

COBOL SYSTEM

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

C1 Drum address table (*3 File) is full. Normally, this problem may be corrected by increasing the size of the *3 File with a $ FILE card. In addition, this problem can be corrected by assigning files *1 and G* to tape. 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 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 Substantive Stack build error in Report Writer, which is an internal compiler problem. (Notify Honeywell field representative.)

C5 Invalid internal list structures of the Report Writer, which is an internal Compiler problem. (Notify Honeywell field representative.)

C6 Fixed portion of the Report Table is full. See COBOL User's Guide 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 COBOL User's Guide 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. (Notify Honeywell field representative.)

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. (Notify Honeywell field representative.)

CA Invalid internal list structures of the Report Writer Generators, which is 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 via $ LIMITS card.

CE Stack overflow in processing source syntax. An internal compiler problem. (See 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 being exponentiated by the value zero.
2. The value zero being exponentiated by a negative value.
3. A negative value being 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 (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 (internal COBOL problem).

## COBOL - TRANSACTION PROCESSING SYSTEM INTERFACE

CW Transaction Processing Executive not in core storage.

CX NO DATA return received on first attempt to access INTERCOM file.

CY Unsuccessful attempt to switch line from Direct Program Access (DAC) mode to Transaction Processing mode.

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.

## PRODUCT

EF Input tape in improper format for PRODUCT.

## MACHINE FAULTS

See Termination and SYSOUT reference manual.

## 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 and the program should not be rerun.

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

## GF and GR Abort Messages

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

| | | |
|---|---|---|
| BSREC | 1 | Illegal request for this routine. |
| | 2 | Status not tape on load point or ok. |
| BSTFM | 1 | Illegal request for this routine. |
| | 2 | Reached load point but more files to skip. |
| CLOSE | 1 | EOF status on output file. |
| | 2 | Unrecoverable I/O error, no user routine. |
| | 3 | File to be closed is not in chain. |
| | 4 | Illegal status for disk or drum. |
| FORCE | 1 | Illegal request for this routine. |
| FSREC | 1 | File is not present. |
| | 2 | EOF on device from prior command. |
| | 3 | Impossible return from system routine. |
| | 4 | Illegal file definition in FCB. |
| | 5 | Unrecoverable I/O error; no user routine. |
| | 6 | Illegal request for this routine. |
| | 7 | I/O status other than blank tape on read. |
| FSTFM | 1 | Illegal request for this routine. |
| GET | 1 | File designated as output file. |
| | 2 | Illegal file definition in FCB. |
| | 3 | Unrecoverable I/O error; no user routine. |
| | 4 | Block serial 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. |
| GF980 | 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 disk or drum format. |
| | 5 | Linked file but not variable record type. |
| | 6 | Illegal format for SYSOUT file. |
| | 7 | File designated as required is not present. |
| | 8 | Two file designators pointing to same file. |
| PUT | 1 | EOF status on output file. |
| | 2 | Unrecoverable I/O error, no user routine. |
| | 3 | Illegal file definition in FCB. |
| | 4 | Current logical record larger than buffer. |
| PUTSZ | 1 | Illegal request for this routine. |
| | 2 | New size larger than old record size. |

|  |  |  |
|---|---|---|
| RDREC | 1 | Binary card is not a COMDK card. |
|  | 2 | The COMDK cards are 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 was 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 disk, drum or tape. |
| WRITE | 1 | File is not present. |
|  | 2 | EOF status on output file. |
|  | 3 | Illegal file definition in FCB. |

## HEALS SYSTEM

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, because the requested SNUMB already is in CRSNB table. (To recover, re-execute HEAL.)

H3 ECF access denied for reason other than "NAME NOT IN MASTER CATALOG" after a MME GEMORE to request ECF access. An undetected hardware error occurred; or, a system software failure occurred.

NOTE: The reason code is in the A-register.

H4 ECF access denied for reason other than "NAME NOT IN MASTER CATALOG" after a MME GENEWS to spawn CRECF. An undetected hardware error occurred; or, a system software failure occurred.

NOTE: The reason code is in the A-register.

H5 MME GENEWS request to spawn CRECF (a FILSYS activity to create the ECF file) was denied. An undetected hardware error occurred; or, a system software failure occurred.

NOTE: The reason code is in the A-register.

H6 ECF file is released or is a system device. An undetected hardware error occurred; 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.

H9 ECFR has waited at least three minutes for HEAL to assign a partial ECF segment. HEAL was terminated while ECFR was in execution.

## I-D-S SYSTEM

ID I-D-S has aborted program for cause, as indicated by its abort message. (See the Integrated Data Store reference manual.)

## 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 could not be corrected via 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

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 (fatal loading errors are listed in the <u>General Loader</u> reference manual).

## FILE AND RECORD CONTROL

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

## GMAP COMPILER

PO Maximum of 63 levels of MACRO expansion exceeded.

## FORTRAN SYSTEM

Q1 Logical Unit Table overflow.

Q2 Missing Logical Unit Table. [illegible]

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.

L4 2/0 ABORT [illegible]

## ALGOL SYSTEM

Q3 Logical Unit 6 not present.

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

## SORT

SC SORT abort; reason indicated on execution report.

## SYSTEM EDITOR

SE System Editor has 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 is encountered when processing direct code.

V1 Compiler error abort. An alternate construction of the indirect statement probably will allow it to compile.

V2 Object program abort. The JOVIAL Compiler generates a MME GEBORT when replacing an erroneous JOVIAL statement.

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

V4 Compile activity has completed and preparation for loader activity (binding the object H*) has 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.

## INDEXED SEQUENTIAL PROCESSOR

XA File opened previously or an access was made to a file that is not open.

XB File not properly opened. Attempting to create a file without first opening with NOPEN.

XC Physical Read or Write error.

XD Insufficient buffer storage; cannot create the required minimum of three buffers. Increase the amount of core allocated via $ LIMITS card or the amount of labeled common storage.

XE Insufficient device storage for index file. Increase the index file size and rerun job.

XF Insufficient device storage for data file. Could be caused at file initialization time; or, by adding overflow records to the file. The file is filled to capacity and user action is required.

XG Record and/or key described by IOPEN do not agree with record and/or key description provided by NOPEN when the file was created.

XH File accessed is not the data file.

XN No coarse or no fine index page found; file has probably been destroyed.

XO Record size greater than 255 words.

XP Page size specified is larger than allowed. The file using the largest page size must be opened first.

XQ Buffer size described by IOPEN does not agree with buffer size provided when the file was created.

SERIES 6000 FORTRAN

NOTE: The abort code Y1 is always displayed as the reason code for any abort. The panel reveals the specific reason code (see codes in parenthesis of following descriptions) 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.

# APPENDIX B

# FILE CODES

Two-character alphanumeric codes assigned by 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 are:

| | |
|---|---|
| A* | Alter File |
| B* | Object Program File |
| *B | DATANET 355 Simulator Input File |
| C* | Binary Deck File |
| *C | Editor Input File |
| D* | Stranger Option File |
| E* | Extension Editor File |
| *F | Series 6000 FORTRAN Dump File |
| F* | Intermediate File, Editor |
| G* | GMAP Source File |
| H* | Program Link File |
| I* | Data Storage File |
| J* | Intermediate File, Source Library Editor |
| K* | Compressed Deck File |
| *K | Source Library Editor Intermediate File for special punch functions |
| L* | System Subroutine Library File |
| *L | Secondary Subroutine Library File |
| M* | Source Master File, 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 |
| *R | Old Object Master File, Object Library Editor |
| S* | Compiler Source File |
| *S | I-D-S Source File |
| T* | Console Typewriter Message File (re Magnetic Tape) |
| *T | Console Typewriter Message File (re Unit Record Device) |
| U* | Utility 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, Change Letter History |
| *6 | COBOL Intermediate File |
| *7 | Punch Card File |
| ** | Dynamic System - loadable file |

# APPENDIX C

## LOGICAL UNIT DESIGNATOR/DEVICE NAME

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 which are to be purged (overwritten) for security purposes or to indicate removable media files which are to be dismounted and removed from the computer system.

```
1      8      16
$      TAPE   A1P
```

The first character of the identifier may be alphabetic or numeric. The second character may be numeric only. 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

Allowable disposition codes for all types of storage media are:

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

and, for magnetic tape and disk pack files only:

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

For example:

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

X1S - Save file X1 normally; abort activity (implied by blank following S).

NOTE: Files are purged only if they have write permission and, 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. When possible, hardware write-protected files should be dismounted rather than purged.

Device Name

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

Use of device names on $ File control cards must be consistent with the type of device so named (i.e., use of the name of a mass-storage device on a $ TAPE card is not allowed).

The device name may be followed by a disposition code.

Naming of a specific peripheral device must conform to the following conventions (see the <u>System Startup and Operation</u> reference manual):

- Each name comprises three characters: The first character must not be zero; the second character must be alphabetic.
- Each name must be unique.
- A device may have more than one name. However, a 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. |

The Other Computer Company:

Honeywell

HONEYWELL INFORMATION SYSTEMS

In the U.S.A.: 200 Smith Street, MS 061, Waltham, Massachusetts 02154
In Canada: 2025 Sheppard Avenue East, Willowdale, Ontario

6710
5273
Printed in U.S.A.