3300
3500

COMPUTER SYSTEMS MASTER

OPERATOR'S GUIDE

CONTROL DATA
CORPORATION

Additional copies of this manual may be obtained
from the nearest Control Data Corporation Sales office.

April, 1967
Pub. No. 60178600

CONTROL DATA CORPORATION
Documentation Department
3145 PORTER DRIVE
PALO ALTO, CALIFORNIA

CONTENTS

CHAPTER 1 INTRODUCTION

CHAPTER 2 OPERATING PROCEDURES

CHAPTER 3 JOB DECKS

INTRODUCTION 1

MASTER is a highly versatile multiprogramming executive operating system for the CONTROL DATA® 3300 and 3500 computers. Its task orientation is particularly adaptable to multi-access and multi-processing applications. The basic design permits expansion to multiple on-line remote stations and to multiple central processing units (CPU's).

MASTER optimizes the use of compute modules and data channels. By simultaneously considering more than one job, MASTER finds activities for these processors whenever they become idle.

MASTER consists of an operating system and a system executive. The operating system, in program state, accepts jobs and translates them into executable entities called tasks, which are manipulated by the system executive. The system executive administers requests made by the tasks for the execution of other tasks and assigns tasks to processors as they become idle. The system executive, which operates in monitor state, is entered through interrupts.

Monitor state and program state comprise the executive mode, selected from the console by the operator when MASTER is initialized. Monitor state allows execution of all instructions. Program state prohibits execution of certain instructions such as I/O, reserved for the monitor. It is entered only through execution of a boundary jump by the system executive (EXEC) operating in the monitoring state.

1.1 CONFIGURATION

MASTER operates with the following minimum configuration:

32K Core Memory

One 3304 or 3504 Central Processor

One 3311 or 3511 Multiprogramming Option

One 405 Card Reader and buffered controller

One 501 or 505 Line Printer and buffered controller

One 415 Card Punch and buffered controller

Two 3306 or 3307 (3506 or 3507) Communications (Data) Channels

2.5 million words of mass storage. This may be obtained by:

Five 852 disks	One 813 disk
Three 853 disks	One 814 disk
Two 854 disks	Three 863 Drums

or any combination of the above that totals 2.5 million words.

A recommended configuration includes four to eight magnetic tape units (none required), additional core memory, and an additional printer.

1.2 FILE STRUCTURE

In MASTER's file-oriented input/output, each unit record device, such as a printer, punch, or magnetic tape unit, and each file definition for mass storage devices has associated with it a unique file identifier called its data set identifier (dsi).

1.2.1 SYSTEM FILES

When MASTER is autoloaded (2.1), a file environment is established consisting of nine mass storage files. The nine system files exist throughout the life of MASTER; they are maintained on permanently on-line (Class A) mass storage. The table that follows outlines each of these nine system files.

System files involved with management of the mass storage system, *MSD, *LAB, and *IDF, may be accessed by the operating system only. Library files *LIB and *DIR are accessed by any job and task in the system as well as by the system itself. Files of the permanently allocated job file pool are accessed by both system and user for jobs running in the system.

1.2.2 FILE LABELS

File labels are tabulated entries in system files that identify and describe space on mass storage. A mass storage file exists in the system when the user defines a label. The user must uniquely identify and describe a file each time he makes a definition.

The important prerequisite to using MASTER's mass storage is that space for files must be labeled and reserved. The user makes calls to the MASTER operating system (*DEF task) to create a file label. These calls provide file identification, security codes, block size, block count, etc.

SYSTEM FILES

COMMON NAME	MASS STORAGE DIRECTORY FILE	FILE LABEL DIRECTORY FILE		MASTER LIBRARY FILE	MASTER LIBRARY DIRECTORY FILE	JOB FILE POOL			
		FILE LABEL FILE	FILE IDENTIFIER FILE			INPUT	OUTPUT	PUNCH	SCRATCH
FILE IDENTIFICATION OWNER FILENAME	MSIO MSDFILE	MSIO LABELFILE	MSIO IDFILE	MASTER LIBRARY	MASTER LIBRARY DIRECTORY	MASTER SYSTEM INPUT	MASTER SYSTEM OUTPUT	MASTER SYSTEM PUNCH	MASTER SYSTEM SCRATCH
EDITION NO.	00	00	00	00-99	00-99	00	00	00	00
SECURITY CODES ACCESS MODIFICATION	←			master security codes as specified by installation					→
BLOCKSIZE	←		variable		→	—fixed length segments			→
DATA SET IDENTIFIER	*MSD	*LAB	*IDF	*LIB	*DIR	INP, OUT, PUN (See Summary)			
Contents	copies of all device labels for all mass storage devices in the system; therefore, a complete mapping of all mass storage	labels of all files in the mass storage system	the file identification and security codes from all file labels in the mass storage system	complete library for MASTER	a directory of the MASTER library	a pool of permanently allocated contiguous mass storage that will be managed by EXEC for each job in the system			
Status after Initialize	opened and assigned data set identifier *MSD	opened and assigned data set identifier *LAB	opened and assigned data set identifier *IDF	opened and assigned data set identifier *LIB	opened by and assigned data set identifier *DIR	allocated by operating system task and segment maps read into EXEC tables. Managed by system OCARE on a job basis through user and operating system calls.			

1.3 JOB FLOW

The`following discussion on job flow outlines the stages through which a job progresses in the MASTER system. Most of these stages involve calls for program tasks, some of which are permanently allocated and others which are loaded as part of the job. In all cases, the tasks executed in the progress of a job are multiprogrammed with all other tasks currently active in the system. These tasks are executed on a priority basis. The operator begins a MASTER run by autoloading a version of MASTER from the library.

1.3.1 JOB INPUT

All user job decks, consisting of control cards and possibly program and data decks, are presented serially to MASTER through the input card reader. The input backgrounder writes card images of a user's deck on a job INP file on mass storage. This transfer is bypassed on a job basis when CR is specified on a DIRECT card. Jobs can be transferred continuously to job INP files by the input backgrounder until there are no more jobs on the input card reader. Reading of jobs is temporarily suspended when no mass storage is available in the INP file pool, or when the schedule table is full.

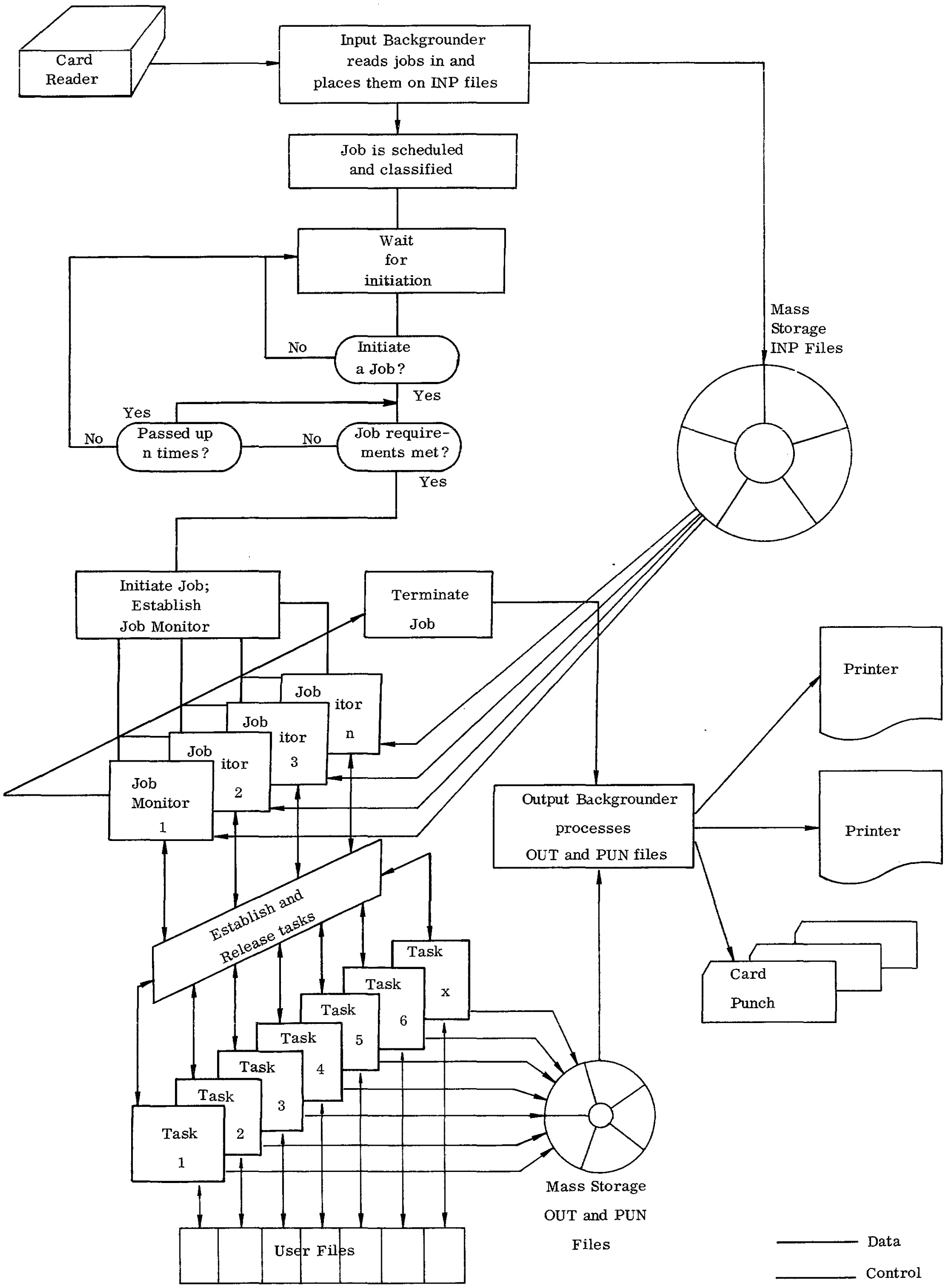
1.3.2 PREPROCESSING

After the input backgrounder transfers a job to its INP file, it calls the job scheduler, an operating system task, and passes to it the information obtained from the DIRECT, JOB, and SCHED cards. The scheduler checks and assigns job classes as outlined in the following diagram. Classifying the job and listing it as a candidate for initiation constitute scheduling.

1.3.3 JOB INITIATION

Whenever possible, MASTER seeks a new job. It considers such variables as job class, equipment and core requirements, and wait time. All required core and I/O devices, such as tapes, card readers, and printers must be available before a job is initiated.

MASTER first looks for emergency jobs waiting for initiation; and if equipment requirements can be met for one, MASTER`initiates it. When MASTER initiates a job, it types B i on the console typewriter where i is the job identifier taken from the JOB card. The operator, through manual interrupt procedures, can determine if a job has been initiated or which jobs have been initiated.


Card Reader
Input Backgrounder reads jobs in and places them on INP files
Job is scheduled and classified
Wait for initiation
Initiate a Job?
No
Yes
Job requirements met?
No
Passed up n times?
Yes
No
Yes
Mass Storage INP Files
Initiate Job; Establish Job Monitor
Terminate Job
Job Monitor 1
Job itor 2
Job itor 3
Job itor n
Establish and Release tasks
Task 1
Task 2
Task 3
Task 4
Task 5
Task 6
Task x
User Files
Output Backgrounder processes OUT and PUN files
Printer
Printer
Card Punch
Mass Storage OUT and PUN Files
Data
Control

Except when emergency jobs are active or waiting, MASTER attempts to keep active at least one job from each of the four regular job classes (background, special, input/output, and compute). Another job from the same class is initiated only if no scheduled jobs from other classes are capable of being initiated.

Within a class, jobs are initiated on a first in, first out basis. However, the first job in the list might not always be the first initiated if core and I/O requirements cannot be satisfied by the available equipment and storage; a job submitted later may be initiated first. On the other hand, no job can be refused initiation because of lack of equipment more than a certain number of times as determined by an installation parameter. When this limit is reached the job's class is changed to emergency and no non-emergency job is initiated until the equipment required by the waiting job is released by terminating jobs.

When its requirements can be met, the waiting job is initiated and normal job initiating resumes. The job initiator loads a copy of the blocking and deblocking routines into memory for the job and calls the job monitor, an operating system task. The job monitor is then loaded and established.

1.3.4 PROCESSING

After initiation, the job monitor processes control statements from the job's INP file. These control statements result directly or indirectly in the loading (when necessary) and execution of program tasks.

A task is a direct part of a job when loading and execution is directly called for by a Task Name control card or by a task of the job currently in execution. Tasks resulting indirectly from a job are those required by the operating system in processing tasks resulting directly from the job.

Once the job is initiated, its priority is set according to its class. Any task having inherited priority inherits this job priority. The priorities of the job classes are: compute = 10_8, I/O = 20_8, special = 30_8, background = 40_8, and emergency = 50_8.

Tasks which require loading, including the job monitor and relocatable loader, occupy core scheduled for the job. The loader, as do most tasks, releases core upon completion of the operation.

When a task and all of the tasks it called are completed, it returns to its caller. If the task is called by a control card, the caller is the job monitor. Processing of a job ends when its job monitor, seeking more work in the job's INP file, detects an end-of-file condition.

Processing can also be terminated by the operator or when a returning task notifies the job monitor of an abnormal condition. Upon abnormal termination if the user requested ABORT on the SCHED card, a recovery dump is written on the job's OUT file. Otherwise, the user receives only a dump of the console registers, and locations 40_8 through 77_8 of the register file, if used.

At job end, any open files are closed. All scratchfiles, the INP file, core, and any scheduled devices are released. The output backgrounder is requested to process the OUT and PUN files when no DIRECT processing takes place. For a direct job, the direct unit file is closed and the device returned to the output backgrounder.

When MASTER closes a job it types T i, where i is the identifier taken from the JOB card.

1.3.5 POSTPROCESSING

The output backgrounder drives all available printers and punches at full speed as long as there are OUT and PUN files to be processed. All printers and punches of specified device types are controlled by the output backgrounder which may relinquish control of an idle printer or punch upon receiving a request from a job. This request results directly from processing a user's OPENU request for a printer or punch of the type in the pool and indirectly from processing of a DIRECT card specifying PR or PU.

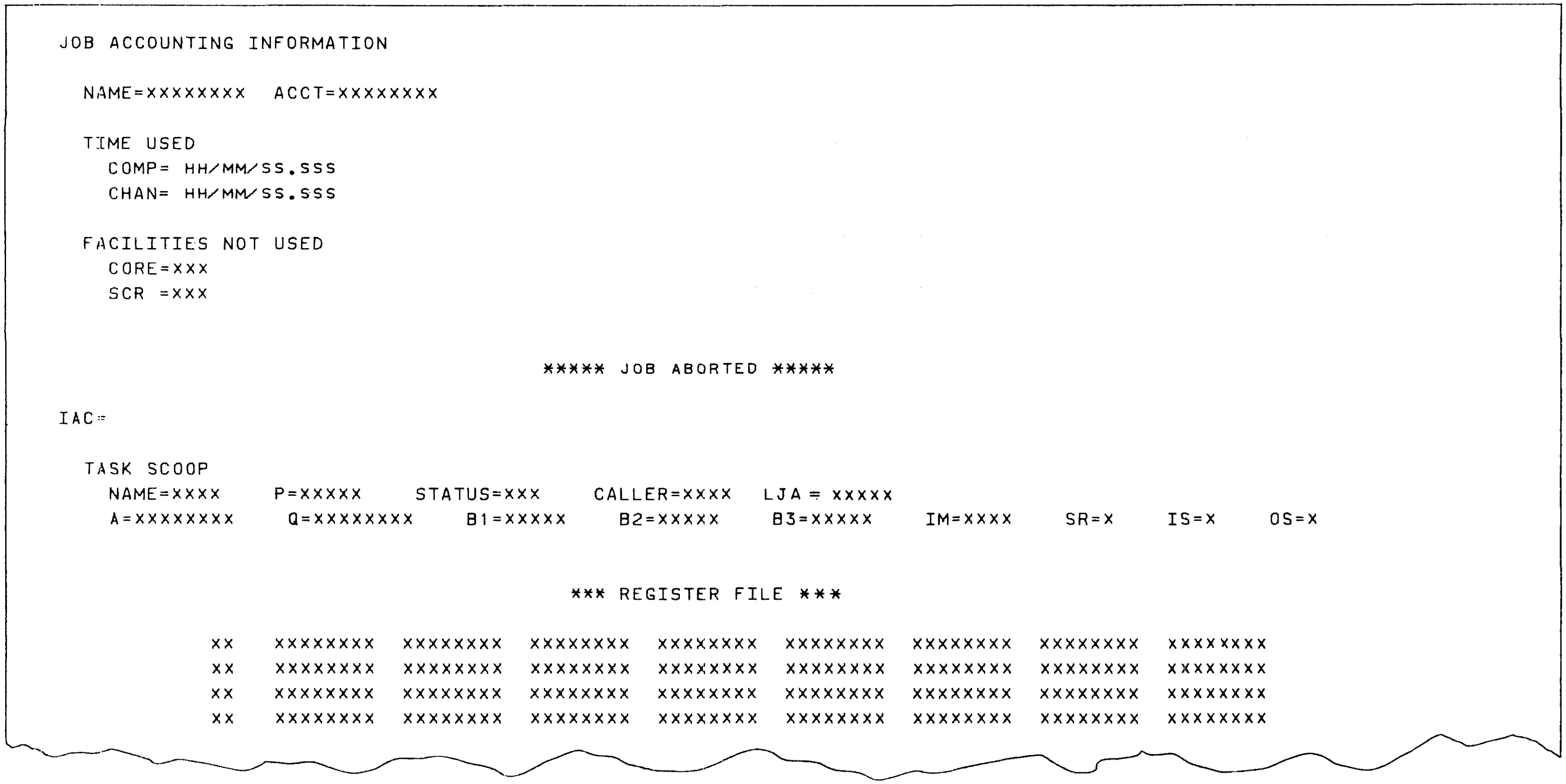
The job containing the OPENU or DIRECT request will not proceed until a unit of the required type becomes available.

OUT FILES

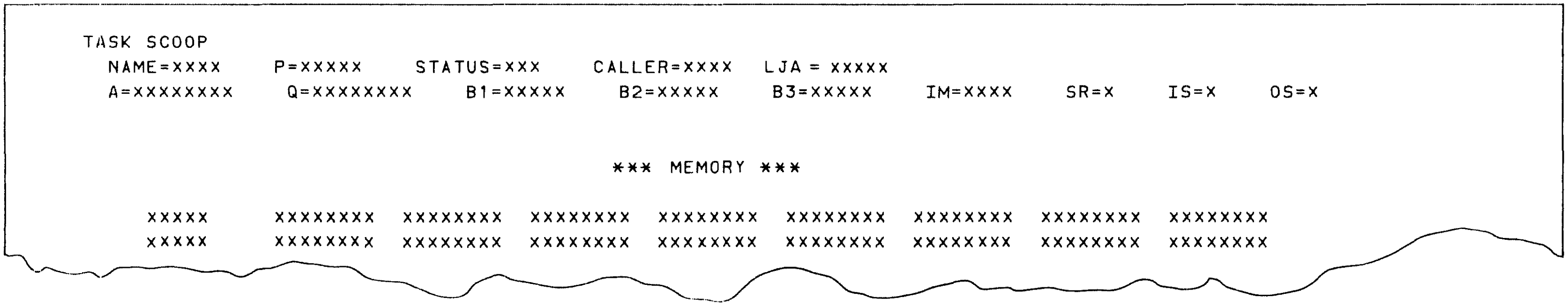
OUT files are placed in a file disposition list which the output backgrounder processes on a first in, first out basis. OUT files are printed on the standard form for the installation. (See XFER, 4.3.)

An OUT file begins with a heading — or for a direct PR job, ends with a trailer — of the form shown. If the job ends normally, only the accounting information is printed. If the job terminates abnormally the information under JOB ABORTED is printed. The register file and MEMORY are printed according to SCHED card options.

The backgrounder then prints information placed on OUT by the user and the job monitor.



```
JOB ACCOUNTING INFORMATION

  NAME=XXXXXXXX  ACCT=XXXXXXXX

  TIME USED
    COMP= HH/MM/SS.SSS
    CHAN= HH/MM/SS.SSS

  FACILITIES NOT USED
    CORE=XXX
    SCR =XXX


                                        ***** JOB ABORTED *****

IAC=

  TASK SCOOP
    NAME=XXXX     P=XXXXX     STATUS=XXX     CALLER=XXXX   LJA = XXXXX
    A=XXXXXXXX     Q=XXXXXXXX     B1=XXXXX     B2=XXXXX     B3=XXXXX     IM=XXXX     SR=X     IS=X     OS=X


                                          *** REGISTER FILE ***

              XX    XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX
              XX    XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX
              XX    XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX
              XX    XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX
```



```
  TASK SCOOP
    NAME=XXXX     P=XXXXX     STATUS=XXX     CALLER=XXXX   LJA = XXXXX
    A=XXXXXXXX     Q=XXXXXXXX     B1=XXXXX     B2=XXXXX     B3=XXXXX     IM=XXXX     SR=X     IS=X     OS=X


                                             *** MEMORY ***

         XXXXX      XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX
         XXXXX      XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX
```

JOB ACCOUNTING INFORMATION

NAME = Identifier taken from JOB card

COMP = Central processor time used by job in hours, minutes, seconds and milliseconds.

CHAN = Sum of time consumed on each I/O channel used by job in hours, minutes, seconds and milliseconds.

CORE = Number of quarter pages reserved in excess of those used; the difference between the core estimated on SCHED card and the maximum used at any one time.

SCR = Number of scratch area segments reserved in excess of those used; the difference between scratch-file estimate on SCHED card and maximum number of segments used at any one time.

***** JOB ABORTED *****

IAC = abort message.
or If abort was at task's request (voluntary)
VAC IAC is replaced by VAC. IAC abort message inserted by EXEC (Appendix D)

*** REGISTER FILE ***

Contents of locations 40_8-77_8 of register file.

*** MEMORY ***

***** OUT FILE OVERFLOW *****

Abort dump exceeds scheduled space.

TASK SCOOP

NAME = Task's name.

STATUS = Task status when terminated.

CALLER = Caller of terminated task.

P
A
Q
B1 } Register contents at termination.
B2
B3
LJA

IM Interrupt mask of internal faults selected by the task.

SR The subcondition register indicates whether operand addresses were routed through operand or instruction state register.

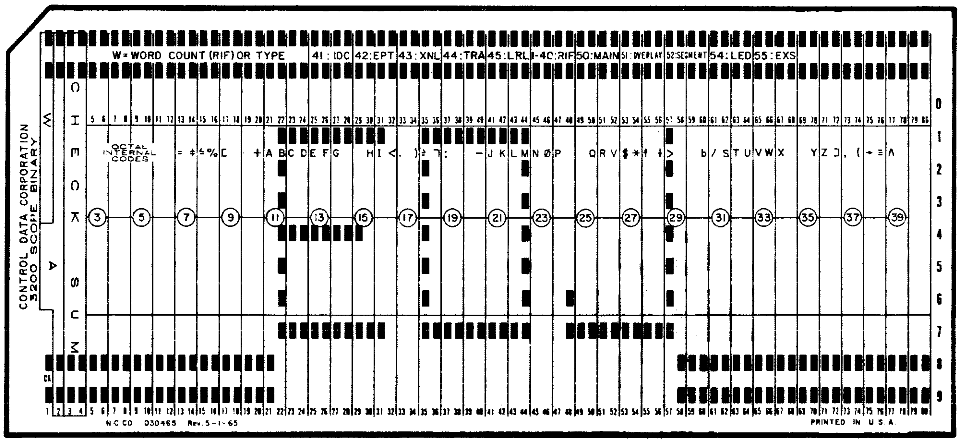
IS Instruction state last assigned to task.

OS Operand state last assigned to task.

PUN FILES

PUN files are placed in a file disposition list which the output backgrounder processes on a first in, first out basis. PUN files are punched on standard cards for the installation. The first card on each punched deck is an exact duplicate of the JOB card for the job.

When the backgrounder detects an end-of-file condition, it punches and offsets the end-of-job card. On a DIRECT job, the end-of-job card is punched and offset before the backgrounder processes the next PUN file.



End-of-Job Card

1.3.6 JOB CLASSES

Each job submitted to MASTER is assigned a job class by the user or the job scheduler. Class determines when a job is initiated and the priorities of its tasks. The job initiator considers job classes when seeking work. Job classes, from highest to lowest are:

Emergency

Background

Special

Input/Output

Compute

Emergency

Emergency jobs are submitted in the same way as all other jobs. A job is classified as an emergency job if (1) the user has declared it as class E on the SCHED card or (2) the job deck is preceded by a DIRECT card, and MASTER has reclassified the job as emergency.

Background

A job primarily intended to drive slow-speed peripheral equipment can be declared as class B on the SCHED card. Background jobs generally use little compute time and when ready, require attention quickly to drive their equipment at full speed.

Special Job

Upon receiving a job declared by the user to be an I/O or compute job or one in which the class declaration is omitted, MASTER determines if the job qualifies as special. For a job to be reclassified as special, the user must supply, on the SCHED card, a time estimate (te) that lies within a range determined by installation parameters Tmin and Tmax, where

$$\text{Tmin} \leq \text{te} < \text{Tmax}.$$

The special class provides faster turnaround time for jobs having time estimates within this range. With this scheme, several jobs are likely to pass through the system during the processing of an I/O or compute job. The class is eliminated if the upper parameter for the range is zero.

Input/Output

A job is classified by MASTER as an input/output job if it does not qualify as special and the user declared it as I on the SCHED card or the installation parameter (set to I) was used in lieu of a declaration.

Compute

A job is classified by MASTER as a compute job if it does not qualify as special and the user declared it as C on the SCHED card or the installation parameter (set to C) was used in lieu of a declaration.

1.4 OPERATING MASTER

For normal MASTER operation, a computer center may find it advantageous to employ two or more operators on a single shift. The amount of service (mounting of disk packs; reloading, unloading of unit devices, etc.) can be prodigious.

Operation is smoothest when the responsibility for the typewriter console resides with a single operator. The console operator directs other operators to service peripheral equipment, and enters operator-to-system messages.

The console operator may interrupt the system and enter a statement to:

Remove a malfunctioning channel, equipment, or unit from the configuration

Return a repaired channel, equipment, or unit to the configuration

Obtain a list of all active jobs currently in the system

Determine if a specific job has been initiated

Update the date

Update the time of day

Terminate a job

MASTER and jobs operating under it communicate with the operator through the console typewriter. A message prefixed with an I is informative and requires no operator action. A message prefixed with an A is an action request; some phase of operation cannot continue until the operator has taken the proper action, perhaps readying of a unit. A message prefixed by an R may require action of the operator, but it always requires that a response message be entered by the operator.

OPERATING PROCEDURES 2

2.1 SYSTEM INITIALIZE

To autoload MASTER:

1. Place the autoload binary deck on the card reader or ready the autoload tape.
2. Release EXECUTIVE MODE key; set SELECT STOP and MC (Master Clear).†
3. Press AUTO LOAD to read the first card or record. The computer will halt.
4. Release SELECT STOP; press EXECUTIVE MODE, MC, and GO.

The system then types R SINT 015 (EDITN)

5. Type the two-character edition code of the library to be loaded.
6. Press FINISH.

The computer reads the card deck into core and searches the File Label Directory for the label of the library having this edition code. An initialize routine is then read from the library. This routine types DATE, TIME, and SET requests on the console typewriter.

The first request is R SINT 009 (SET)

7. The operator may modify any words in resident by typing SET responses or by inserting SET control cards at the card reader.

   SET responses that change table entries pertaining to peripheral equipment must be typed.

SET Responses at Typewriter

a. Type: SET, task, ept±xxxx, m_1n_1/v_1, m_2n_2/v_2, . . .

   task Name of task containing word to be modified. Task may be omitted, but its trailing comma must appear.

   ept System entry point to task; it must be a global entry point for the library being autoloaded.

† In installation equipped with standard option toggle switch memory protection, all switches must be set to the one's position.

xxxx 1 to 4 decimal digits specifying the number of locations to be moved forward (+xxxx) or backward (-xxxx) from ept to find the word to be modified.

mn/v Value v is to be set into n bits according to mode m as follows:

m = 0 Octal; $1 \leq n \leq 24$, $1 \leq v \leq 77777777_8$. Value v is entered into the field of n bits right justified, with leading zeros.

H Hollerith; n is a multiple of 6 and ≤ 24. Value v is stored as 6-bit internal BCD characters, right justified with leading spaces.

N No change; $1 \leq n \leq 24$. n bits remain unmodified. v should be omitted; if present, it is ignored. The slash must be present.

If v exceeds the length of the field set by n, it is truncated; the rightmost digits are used.

A SET response can modify only 24 bits; modification begins at bit 23.

b. Press FINISH.

Initialize types I SINT 011 (aaaaa = cccccccc)

aaaaa	octal address
cccccccc	octal contents of aaaaa entered by a SET response

c. Repeat steps a and b for each word to be modified.

d. When no more words are to be modified, press FINISH.

SET Responses from the Card Reader

SET control cards on the card reader may replace SET responses at the typewriter. Parameters on the control card correspond to those in the typed SET response. Each card modifies one word in memory.

$\text{\$SET, task,ept} \pm \text{xxxx,} m_1 n_1 / v_1, m_2 n_2 / v_2, \ldots$

An end-of-file card must follow the last SET card on the card reader.

To have the initialize routine read SET control cards, type CR and press FINISH after initialize makes the request R SINT 009 (SET). Initialize makes further requests before reading SET cards from the card reader.

Combined Typewriter and Card Reader Responses

When using a combination of typed and card SET responses, make all typed responses first. When initialize types R SINT 011 (aaaaa = cccccccc) after the last typed SET response, type CR and press FINISH. Initialize then makes the DATE, TIME, and DIRCT requests before reading the SET cards, which must be followed on the card reader by an end-of-file.

SET Errors

If the operator has made a format error in a typed SET response, after he presses FINISH, initialize types

R SINT 005 (REPEAT)

The operator must repeat the SET response correctly.

If a SET control card contains a format error, initialize types the card image and the informative message I SINT 017 (FMT ERR). Job processing takes place without modification of the word indicated on this SET card.

Initialize next types R SINT 007 (DATE)

8. Type the 6-character numeric reply in the form mmddyy (month, day, year).
9. Press FINISH.

Initialize types R SINT 008 (TIME)

10. Type 4- or 6-character reply of the form hhmmss (hours, minutes, seconds elapsed since midnight; ss is optional).
11. Press FINISH.

The last request typed by initialize is R SINT 010 (DIRCT)

12. Press FINISH to set up standard files as currently defined in the Label File.

or

Type YES and press FINISH if the initialize routine is not to set up the standard file pool on mass storage.

The first job on the card reader (following any required SET cards) is then a DIRECT job which allocates these files on the desired mass storage device. After completion of this DIRECT job, repeat the autoload procedure, steps 1 through 12, to use the files set up by the DIRECT job.

At the end of the autoload procedure (and before reading the SET cards or DIRECT job) initialize informs the operator of the number of quarter pages of available core with the message

```
I SINT 012 (AV. CORE =nnnn QP).
```

Following is an example of typeout at the end of initialization:

```
R SINT 015(EDITN) QA
R SINT 009 (SET)
R SINT 007 (DATE) 020367
R SINT 008 (TIME) 0815
R SINT 010 (DIRCT)
I SINT 012(AV. CORE = 0059 QP)
```

2.2 UNIT LOGGING

Both OPEN and OPENU functions of the *DEF task, as well as the *FMU task, log hardware assignments on the console typewriter.

*DEF Format

```
Rr JOB i *DEF LOGGING
dsi=hhhh,yyyyyy,CcEeUuuu, WR
          yyyyyy,CcEeUuuu
          .
          .
          .
          etc. (one line for each device currently needed for the file
                and not on line)
          .
          .
END LOG i
```

*FMU Format

```
Rr *FMU LOGGING
hhhh, yyyyyy,CcEeUuuu
          .
          .
          .
END LOG *FMU
```

r	message number (0-9) assigned by the system
i	job identifier taken from JOB card
dsi	data set identifier
hhhh	hardware type
yyyyyy	device number or device identifier. This field is blank for scratch tapes and unit record devices.
EcEeUuuu	channel, equipment, and unit
WR	Insert write ring on tape to be used for output; otherwise, WR is omitted from message.

The operator must either ready the logged devices and confirm the request, or reject the logging request.

When the request to ready the unit has been honored:

1. Press MANUAL INTERRUPT
2. Type Rr, OK
3. Press FINISH

To reject the request:

1. Press MANUAL INTERRUPT
2. Type Rr, NO
3. Press FINISH

Tape

After the operator has mounted a tape requested by a logging statement, MASTER:

rewinds it.

confirms that the tape was mounted on the specified unit.

confirms write ring status. If tape has a write ring and usage is I (input only), or if the tape has no write ring and usage is O (input and/or output), MASTER repeats the logging request.

verifies the label if it is present and sets tape unit to the proper density. An unlabeled tape is left at the density set by the operator when he mounts the tape.

Disk

If the operator has mounted a requested disk, MASTER confirms that the device label is correct. A disk may not be ready for several seconds after the operator presses the START button.

If an error occurs during the above procedures, MASTER repeats logging until:

Condition is corrected.

Operator rejects the request.

Request has been repeated five times. In this case, MASTER assumes the request has been rejected.

After an open unit request has been processed, the physical unit assignment cannot be changed by dialing in another unit.

2.3 MANUAL INTERRUPT PROCEDURES

The operator may request an EXEC function, using a command code, as follows:

1. Press MANUAL INTERRUPT
2. When TYPE LOAD lights, type

   cmcd, message

cmcd	1-4 character command code. It may not be R0-R9 and may not contain a carriage return.
message	Message to be transmitted; maximum 80 characters, including command code, comma, and blanks.

3. Press FINISH

If an error is made in typing input, press REPEAT before pressing FINISH, and repeat steps 1-3.

If the operator's message is unintelligible to MASTER, the system types

ILL. PARAM

The operator must repeat steps 1-3.

2.3.1 OPERATOR TO EXEC COMMANDS

Executive command codes and corresponding messages are listed below. Each command code consists of the characters EC, followed by a two-digit code number.

EC01,CcEeUuuu,DN

EC01,CcEeUuuu,UP

These messages declare the hardware indicated by channel c, equipment e, and unit uuu as down or up.

The operator cannot declare a real-time unit or channel to be down.

To refer to an equipment only, omit Uuuu; then all units on equipment e are affected. However, if any of the units on equipment e are also on some other equipment, they are still unaffected on this other equipment.

By omitting both Ee and Uuuu parameters, the operator declares that all equipments and units on channel c are affected. If these same equipments and units are also on another channel, they remain unaffected on that channel.

EC02,hr/mn/sc/mil

The operator uses this message to modify the time allowed for any one task to execute without interruption.

hr	hours, 2 digits
mn	minutes, 2 digits
sc	seconds, 2 digits
mil	milliseconds, 3 digits

The message must include all parameters and slashes.

EC03,hr/mn/sc

The operator uses this message to update the time of day. The parameters are the same as the corresponding parameters in EC02; all must be specified, with slashes.

EC04, mo/dy/yr

Executive Command Four updates the date.

mo	month, 2 digits
dy	day, 2 digits
yr	year, 2 digits

All parameters and slashes must be present.

EC05, JT

In response to this request, the system types the identifiers of all jobs initiated and in some stage of execution.

i INIT job i is in execution; the message is repeated for each job in the Job Table. i is the job identifier from the JOB card.

NO JOBS INIT No jobs are currently in execution; all initiated jobs have terminated.

EC05, SC, x

This message requests a list of all jobs of class x that have been scheduled, but not initiated.

x =	E	(Emergency)
	B	(Background)
	S	(Special)
	I	(I/O)
	C	(Compute)

MASTER replies:

i IN x SC Job i is scheduled under job class x; the message is repeated for each job in class x.

or

NO JOBS IN x SC No jobs are currently scheduled in class x.

EC05, ST, i

In response to this request, the system types the status of job i:

i UNDEF Job i has not been scheduled or has already terminated.

or

i IN x SC Job i is scheduled in class x, but has not yet been initiated.

or

i INIT Job i has been initiated.

EC05, OT, i, N

The operator uses this request to terminate Job i. Parameter N is optional; if N is present, Job i is terminated without an abort dump, regardless of the ABORT parameter specified on the SCHED card. MASTER types:

D JOB i y E05	The operator aborted Job i during execution of task y.
or	
i UNDEF	Job i is unknown to the system.
or	
RPT OT i	The system temporarily cannot honor the request. Repeat the request immediately.

2.4 FORMS CONTROL

When MASTER encounters an XFER Task Name control card containing the f parameter (section 4.3.2), the system requests the operator to mount form f:

```
Rr XFER 001  (MOUNT FORM f ON ht CcEeUuuu)
```

The operator may comply or refuse the request.

To comply:

1. Mount requested form
2. Press MANUAL INTERRUPT
3. Type Rr, OK
4. Press FINISH

To refuse the request:

1. Press MANUAL INTERRUPT
2. Type Rr, NO
3. Press FINISH

If the operator refuses the request, files requiring form f are not processed.

When the transfer is complete, MASTER directs the operator to return to the installation's standard form by typing:

```
Rr XFER 002 (REMOVE FORM f FROM ht CcEeUuuu)
```

The operator must comply to allow the XFER task to resume execution. Operator is required to:

1. Remove form f
2. Press MANUAL INTERRUPT
3. Type, Rr, OK
4. Press FINISH

2.5 SHUT DOWN PROCEDURES

When the card reader and all other peripheral equipment become inactive, the operator may make certain MASTER has completed job processing by using the manual interrupt procedures described in section 2. 3 with the request EC05, JT. If all jobs submitted to MASTER have terminated, the system types NO JOBS INIT. Since MASTER may still be processing real-time tasks which are not jobs, the operator should not press STOP or MC until he is certain that all processing is finished.

JOB DECKS 3

A user submits a job to MASTER as a set of control cards which may be accompanied by source language decks, binary object decks, and data. The control cards used most often by the operator are described below.

DIRECT
JOB
SCHED
FILE
Task Name
End-of-file

Other control cards are described in the MASTER Reference Manual and the MASTER Installation Manual. Source language decks are described in the compiler and assembler reference manuals; binary object decks (generated by compilers and assemblers for execution under MASTER) are described in the MASTER Reference Manual.

3.1 CONTROL CARDS

From control cards, MASTER receives the information it needs to allocate storage and equipment, schedule a job and initiate its tasks, assign priorities, and perform other job-monitoring functions. A MASTER control card, with the exception of an end-of-file, is identified by a $ in column 1. When a control card is transferred to a job's INP file, trailing spaces are removed.

When the job monitor interprets a control card, it copies it onto the job's standard output file. If a control card is out of sequence or contains an error, the job is terminated and a message is written on the job's OUT file.

3.1.1 DIRECT

The DIRECT control card is optional; it permits a job to use the card reader for input, and a printer or a punch for standard output without first transferring the data to files on mass storage. When used, the DIRECT card precedes the JOB card as the first card in the job deck.

```
$DIRECT,CR,PR,PU
```

CR When the job is initiated, MASTER does not transfer the job deck to an INP file on disk, but makes the input card reader the job's INP file. No other job can be transferred to mass storage while the card reader is so occupied. When CR is omitted, normal job input and transfer to mass storage take place.

PR Standard printer output for the job goes directly to a printer without first being transferred to the OUT file. The job will not be initiated until a printer is available. When PR is omitted, normal transfer of standard output to the OUT file takes place. If PR is specified, MASTER ignores the line limit, set by either an installation parameter or the JOB card parameter.

PU Standard punch output for the job goes directly to a card punch without first being transferred to the PUN file. The job will not be initiated until a punch is available. When PU is omitted, normal transfer of standard punch output to the PUN file takes place. If PU is specified, MASTER ignores the punch limit set by either an installation parameter or JOB card parameter.

The parameters are free field and may appear in any order. A parameter and its comma may be omitted. The job monitor ignores extra spaces. Equipment reserved for a job by a DIRECT card is released when the job terminates.

If a parameter on a DIRECT card is not in correct form, the input backgrounder skips the erroneous parameter and types the following message on the console typewriter:

```
I   JOB i *BKI 01
```

i identifies the last job read in. Processing continues with the job following the DIRECT card, ignoring the bad parameter.

DIRECT makes it possible to run a job with input or output that exceeds available mass storage. A printer or a punch acquired with a DIRECT card must be scheduled on a SCHED card (3.1.3).

Avoid using a DIRECT card whenever possible; its indiscriminate use would contravene system design.

3.1.2 JOB

A JOB card must appear in a job deck as the first card or, if DIRECT is used, as the second card. It can be followed by a SCHED, Task Name, FILE or RLDR control card. The system ignores any additional JOB cards detected before an end-of-file card. The job monitor ignores extra spaces on the card.

```
$JOB, c,i,tl,l,p
```

c 1 to 8 BCD characters indicating account to be charged; may not be omitted.

i 1 to 8 BCD characters identifying originator of the job; may not be omitted.

tl Limit ≤1440 minutes, for job execution beyond which MASTER terminates the job. The limit does not include background processing. If tl is omitted but one of the other limits is specified, its comma must appear.

l Line printer limit ≤99999 for the OUT file, beyond which MASTER terminates the job.[†] If line limit is omitted but punch limit is specified, its comma must appear.

p A punched card limit for the PUN file (0 to 99999) beyond which MASTER terminates the job.[†] Comments may follow the p field when it is terminated with a comma.

All limits are optional. They do not apply to DIRECT jobs. When they are not specified and the job is not DIRECT, MASTER uses installation parameters for time and line limits. The installation parameter used for the punch limit is normally set to zero, in which case no punch file is allocated.

Example:

```
$JOB,421,BT2,15,150,100,COMMENTS APPEAR HERE
```

[†] The line and punch limits are not set to the actual number of lines or cards as specified in these fields. They are merely used to calculate the number of segments of mass storage scratch area to be assigned. Since trailing blanks are deleted on the mass storage files, many more lines or cards may be allowed than specified.

3.1.3 SCHED

SCHED cards, of which there may be two per job, immediately follow the JOB card in the job deck. Correctly used, the SCHED cards promote efficient MASTER operation. SCHED cards are followed by a Task Name, XFER, FILE, TASK, or an RLDR card.

```
$SCHED,TIME=te,CLASS=cl,CORE=qp,SCR=seg,ABORT=dl,

                                  RF=flg,peq1=u1,...,peqn=un
```

All fields are optional; they may appear in any order on either card. If a field other than peq is inadvertently repeated, (two TIME fields for instance) the second entry has precedence. When a peq field is repeated (two entries for 501 = u), the first entry has precedence. A field cannot be continued from one card to the other. The job monitor ignores any extra spaces on the card.

DO NOT SCHEDULE MORE THAN A JOB NEEDS. A job that requests unneeded core, mass storage, or peripheral equipment may needlessly wait for facilities which must be available before the job can be initiated. Also, when the job is processed, the idle facilities are withheld from other jobs requesting them.

TIME=te Estimated running time (0 to 99999 minutes). MASTER uses it to determine if the job is special. When this field is omitted, the job cannot qualify as a special job. te is not a limit; for time limit, see JOB card.

CLASS=cl

cl	Class
E	Emergency
B	Background
I	Input/Output
C	Compute

When both TIME and CLASS fields are specified and te lies in the special class range set by installation parameters, the job acquires the higher class as determined by both fields. If the job is specified as E or B which are higher than special, it will maintain its specified class, but if it is specified as I or C, which are lower than special, it is reclassified as special.

When the CLASS field is omitted and the job does not qualify as special, it becomes I or C depending upon an installation parameter.

CORE=qp

Estimate of the maximum amount of core in quarter pages used by tasks residing in core simultaneously. An estimate includes requirements of library tasks such as compilers and assemblers. Estimates for object decks must allow for expansions of pseudo instructions, macros, library routines, etc.

A job cannot exceed its estimated core. When the loader determines that a requested task exceeds the estimate, it terminates the job and writes a message on the OUT file.

When the CORE field is omitted, qp is set by an installation parameter.

SCR=seg

Number of segments of mass storage scratch area required.

If the sum of the mass storage requirements as indicated by the line and punch limits (JOB card) and the SCR and ABORT requests (SCHED card) exceed the storage reserved for these files, the job is not initiated. A diagnostic is typed on the console typewriter and the INP file containing the job is released.

When the SCR field is omitted, seg is set by an installation parameter.

ABORT=dl

Requests a recovery dump of task. dl specifies number of lines to be dumped if the job is abnormally terminated. MASTER reserves mass storage for dl lines of dump. When dl is 0 or the field is omitted, the user obtains only the dump described in section 1.3.5 and not a dump of his task.

RF=flg

Non-zero flg indicates that the job will use the register file. When any task of the job is interrupted, the contents of file registers 40-77 will be saved. When a task of the job is again placed in execution, the contents will be restored.

When flg is zero, or the field is omitted, the register file contents are not saved.

$peq_i=u_i$

All peripheral equipment other than Class A mass storage devices required by a job must be scheduled. peq identifies the hardware type; u designates the number of units or drives required.

peq	Hardware type
405	card reader
415	punch
501	printer

peq	Hardware type
607	607 tape unit
606	606 tape unit
604	604 tape unit
603	603 tape unit
852	852 disk drive
853	853 disk drive
854	854 disk drive

The following card reserves one 501 printer, three 607 tape units, and two 852 disk drives.

```
$SCHED,501=1,607=3,852=2
```

Mass storage drives need be scheduled only for files on removable Class B devices.

A printer or punch required by a DIRECT job must be scheduled.

If peq requirements exceed system capacity, the job cannot be scheduled. The message D JOB i *SCH 07 is typed, the job is abnormally terminated, and its INP file is released.

A job can be initiated when all peq requirements are satisfied.

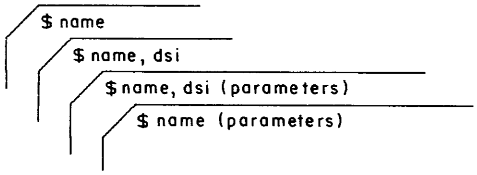
Any attempt by a job to use a device not scheduled causes job termination with a message on the job's OUT file.

To summarize, when no SCHED card is used, the job cannot be classified as special, installation parameters are used for class, core, and scratch, no recovery dump will be given on abnormal termination, and the job cannot use the register file.

3.1.4 TASK CALLS

A Task Name control card directs MASTER to call and load the named program task from the specified file (or from the library if no file is specified), to pass any parameters, and to begin execution of the task. Usually at least one Task Name card will follow the SCHED cards for a job.

A Task Name card has one of the following forms:



```
$ name
$ name, dsi
$name, dsi (parameters)
$ name (parameters)
```

name — 1-4 alphanumeric characters identifying the task to be called; name is the only required parameter.

dsi — The dsi of an opened file from which the named task is to be loaded. If dsi is 0 or omitted, MASTER looks for the task on the system library.

parameter string — Parameters used by the called task, for example the *DEF parameters specified in section 4.1. MASTER removes any spaces in the parameter string before passing the string, with parentheses, to the called task.

When dsi is INP, the binary object deck for the task immediately follows the Task Name card or TASK card. When dsi is other than INP or *LIB, the loader relocates to block 1 for mass storage or rewinds the tape.

A source or data deck required by the task may follow the Task Name card or binary object deck on the INP file.

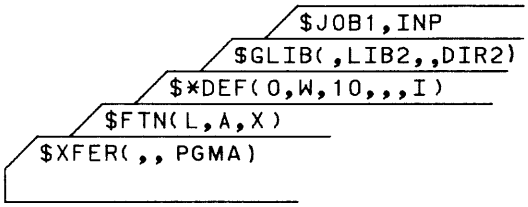
COMPASS, FORTRAN, COSY, and GLIB are examples of standard library tasks. Corresponding task names are CMP, FTN, COSY, and GLIB. Object time routines are tasks arbitrarily named by users. Thus, the quantity and variety of Task Name cards is unlimited. Operators commonly use three library tasks described in Chapter 4:

XFER

*FMU

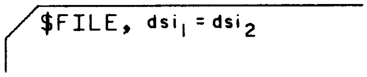
*DEF

Examples of Task Name cards:



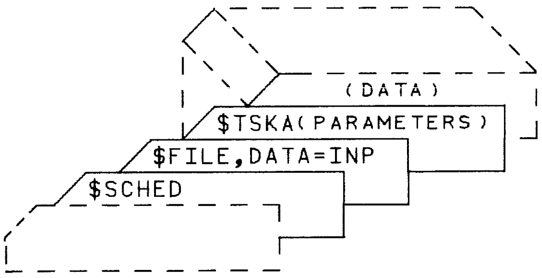
```
$JOB1,INP
$GLIB(,LIB2,,DIR2)
$*DEF(0,W,10,,,I)
$FTN(L,A,X)
$XFER(,, PGMA)
```

3.1.5 FILE



```
$FILE, dsi1 =dsi2
```

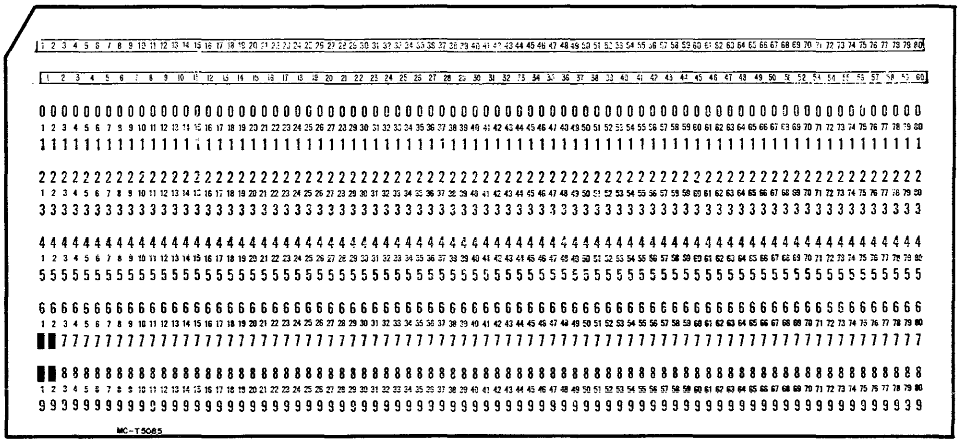
A FILE control card indicates to MASTER that dsi_2, defined earlier in the job, can also be referred to as dsi_1. They are the same physical file; thus, two tasks of the same job can refer to the same file by different names. When dsi_2 is INP, OUT, or PUN, FILE can immediately follow SCHED. Otherwise, it must come after the *DEF control card, FILE card, or task that defines dsi_2. For example, TSKA, below, refers to file DATA for its input. The data can be entered on the INP file by using:



```
(DATA)
$TSKA(PARAMETERS)
$FILE,DATA=INP
$SCHED
```

3.1.6 END-OF-FILE

A job is terminated with an end-of-file card characterized by 7, 8 punches in columns one and two. This card has the same format as an end-of-file card that terminates PUN output. Columns 3-80 may contain comments.

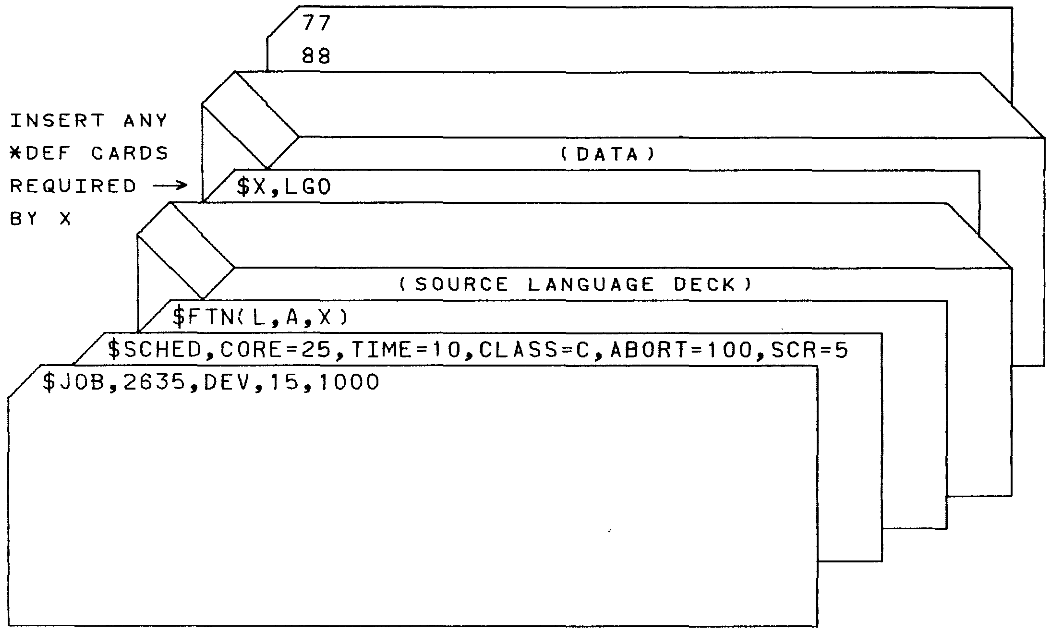


3.2 DECK PREPARATION

Decks portrayed in this section represent general job applications of MASTER. The operator submits decks in sequence on the input card reader.

3.2.1 COMPILATION AND EXECUTION

The user with a source language program, such as a set of FORTRAN statements, to be compiled and executed prepares his deck as shown below. In the example, LGO is a scratch file allocated and opened by the first compiler or assembler of a job using System OCER. It is not closed until the end of the job.

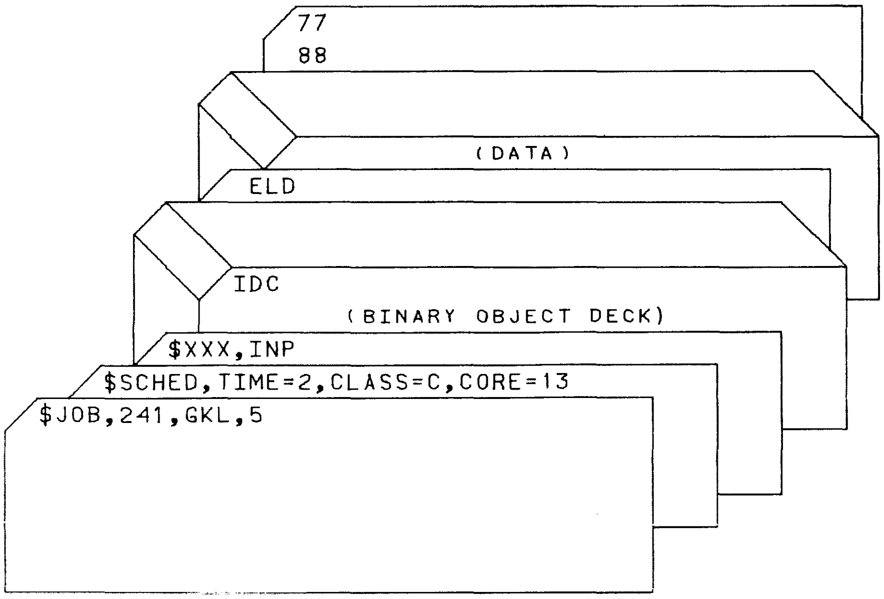


INSERT ANY *DEF CARDS REQUIRED → BY X

```
77
88
(DATA)
$X,LGO
(SOURCE LANGUAGE DECK)
$FTN(L,A,X)
$SCHED,CORE=25,TIME=10,CLASS=C,ABORT=100,SCR=5
$JOB,2635,DEV,15,1000
```

3.2.2 OBJECT DECK EXECUTION

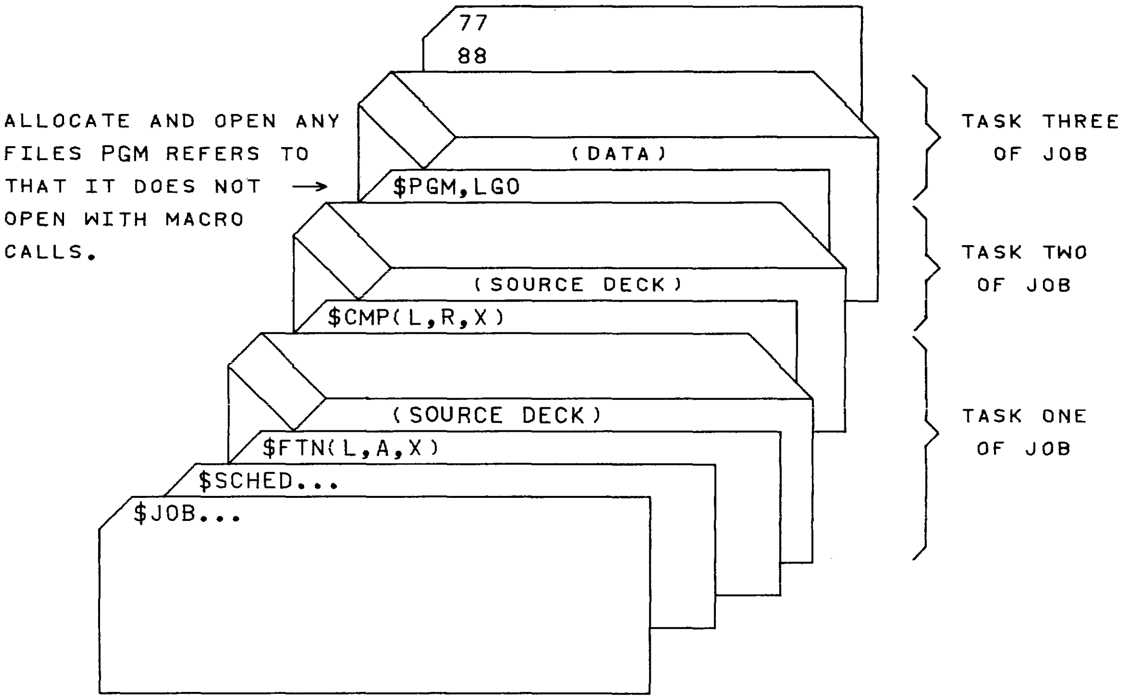
To load and run a binary object deck generated by a MASTER compiler or assembler, organize the deck similar to the example below. Schedule all peripheral equipment required and, if the program itself does not do so, insert control cards to allocate and open files used by the program.



```
77
88
(DATA)
ELD
IDC
(BINARY OBJECT DECK)
$XXX,INP
$SCHED,TIME=2,CLASS=C,CORE=13
$JOB,241,GKL,5
```

3.2.3 MULTIPROGRAMMING A JOB

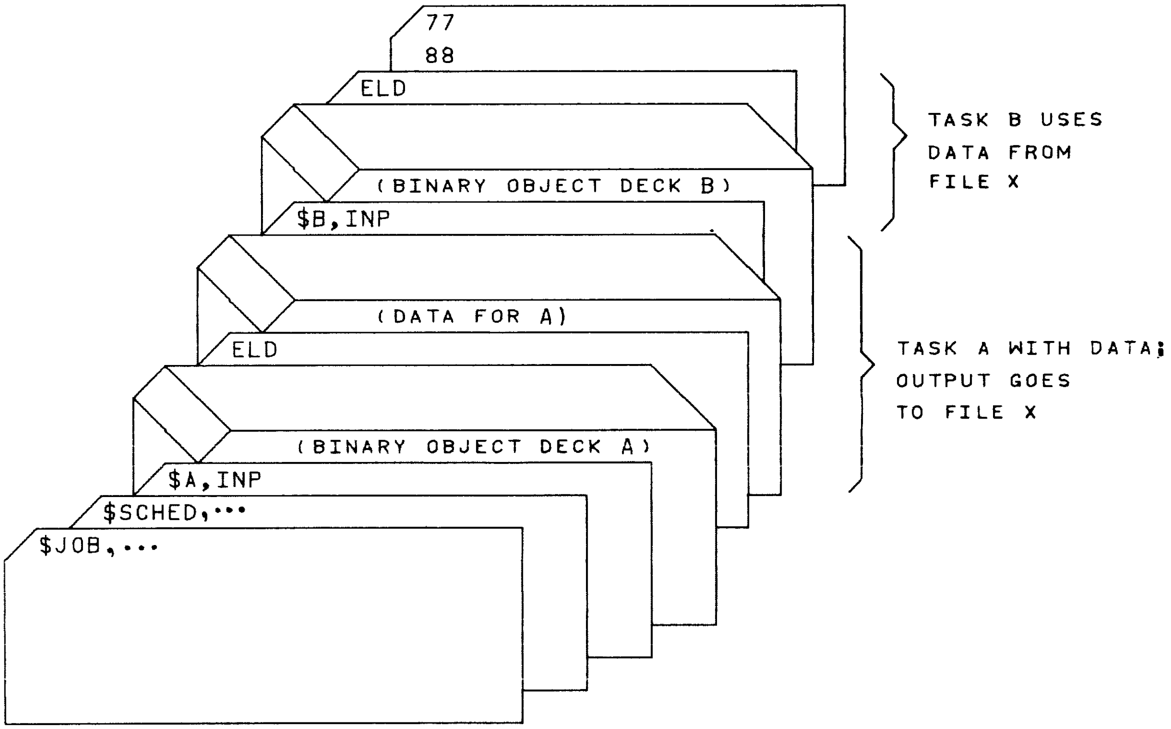
When a job consists of several Task Name cards, the named tasks will not be multiprogrammed within the job.



In this job, the FORTRAN compiler will not be multiprogrammed with the COMPASS assembler. Subtasks of any one task, in this case FORTRAN, COMPASS, or PGM, can be multiprogrammed on an internal basis. Each may consist of several tasks to be multiprogrammed or processed sequentially. The operator may be unaware of the existence of such tasks. Internal structuring of tasks within a task is accomplished through the COMPASS CALL macro described in the MASTER Reference Manual. In the above example, the COMPASS source deck being assembled into PGM may contain calls for other tasks.

3.2.4 SERIALLY DEPENDENT JOBS

If the output from one job is required before a second job can proceed, the second job must not be submitted to the system before the first has reached completion. There is no way in which the two jobs† can be linked so that one will wait for the other. Some serially dependent jobs can be made serial tasks of the same job so that the system will not multiprogram them. In the example, task B which uses the output from task A, will not be initiated until task A has completed.



[†] A job begins with a JOB or DIRECT card and terminates with an end-of-file card.

OPERATOR TASKS 4

The following three MASTER library tasks are of particular use to the operator:

*DEF, which contains file definition routines

*RMU, which includes a variety of file maintenance routines

XFER, which performs interdevice transfer or copying of files

4.1 *DEF

The MASTER operating system task named *DEF handles permanent files as directed by users and the operator. The operator may call *DEF by preparing a job deck containing task name control cards for *DEF. For example, he must use *DEF cards to prepare files for XFER and *FMU tasks.

General form of the *DEF Task Name control card:

```
$*DEF ( function, wait, ••• )
```

function   Character identifying name of the *DEF routine called (allocate, modify, expand, release, open, open segment, open unit, close).

wait   Defines action to be taken when the call causes a conflict with the use of the file across jobs.

W   For mass storage, if the requested file is in use, *DEF places the calling task in file wait status. When the conflict is resolved, *DEF honors the request.

For a unit record device (magnetic tape unit, card punch, printer, etc.) wait has meaning on a U function only if the device is of a type maintained in the background pool. *DEF will wait for a unit of the requested type to be released and then open it.

R   If a conflict occurs, *DEF terminates the job making the call and types a message to the operator.

Other   If wait is other than W or R, MASTER interprets it is a W.

Parameters must appear in the order shown. A comma must appear for an unspecified field if any fields appear to its right. Otherwise, the right parenthesis suffices.

Permanent mass storage files require use of the file handling and data transmission functions. Unit record devices require only data transmission functions.

Label Handling Functions

Label handling functions manipulate the permanent file definitions. The operator uses these functions to create a file definition (Allocate), modify an existing definition (Modify), increase the defined size of a file (Expand), remove a file definition from the directories (Release).

Data Transmission Functions

The data transmission functions do not themselves perform data transmission on a file; an open function assigns a unique data set identifier (dsi) to a file and signifies that the job making the open request is going to transmit data to or from the file, thus prohibiting any other job in the system from concurrently writing on the file. Open functions include open (O), open segment (S), and open unit (U). The file remains open until the job closes it using a close (C) function, or until the job ends and the MASTER job terminator automatically closes all files open to the job.

4.1.1 ALLOCATE

The *DEF allocate function creates a file definition in the system and updates the *LAB, *IDF, and *MSD files accordingly.

```
$*DEF (A, wait, owner, filename, edition, acsc, mdsc, bksize, nbks,
          expdate, s, c, dt, dn, •••, d )
```

Identification Parameters

owner 1-8 alphanumeric characters

filename 1-30 alphanumeric characters

edition 1 or 2 alphanumeric characters

Security Parameters

acsc 1-4 alphanumeric character access security code

mdsc 1-4 alphanumeric character modification security code

Structure Specifications

bksize Number of 6-bit characters per logical block (1 to 131071). The bksize parameter must be present.

nbks Number of logical blocks in the file (1 to 8388607). The nbks parameter must be present.

expdate Expiration date to be assigned to file; it has the form yymmdd (year, month, and day). When the parameter is not specified, the current date is assigned.

Hardware Requirements

s T: File to be in track mode.

S or other: File to be in sector mode, if applicable to specified device type.

c C: File to be allocated to contiguous area (one segment); it may not be segmented.

S or other: File may be segmented.

dt Model number of disk or drum on which file is to be allocated:

852
853
854
813
863

Other: File to be allocated on model specified by installation parameter.

dn Up to nine Class B device numbers to be used for allocation of the file in the order listed. Each device is assigned a number when it is entered in the configuration. (See 4.3.3 and 4.3.4.)

If no device number is specified, the file is allocated on Class A devices of the type specified.

Label information that *DEF automatically sets when it creates a mass storage file:

File protection	Set to input and/or output (O) on an allocate request. It can be changed to input only (I) on a modify request (4.1.2)
Usage count	Set to zero
Creation date	Set to current date
Last access date	Set to 00 00 00

Allocation may not be across storage classes (part on Class A and part on Class B). Device numbers, when listed, must be Class B; hardware type and mode must be specified.

4.1.2 MODIFY

An operator may use the *DEF modify function to change owner's name, file name, edition, access security code, modification security code, file protection, and expiration date in the file definition.

```
$*DEF(M, wait, owner, filename, edition, acsc, mdsc, new owner, new filename,
    new edition, new acsc, new mdsc, new protection, new expdate)
```

The following parameters are mandatory:

owner	Owner named in current definition
filename	Name of file in current definition
edition	Number in current definition
acsc	Current access security code of definition
mdsc	Current modification security code of definition

The following optional parameters specify changes to be made to the file label. A null or blank parameter indicates that field of the label is to be unaltered.

new owner	1-8 alphanumeric characters identifying new owner
new filename	1-30 alphanumeric characters specifying new name for file
new edition	1 or 2 characters giving new edition number of file

new acsc	1-4 characters giving new access security code
new mdsc	1-4 characters giving new modification security code
new protection	O: File may be used for input and/or output
	I: File may be used for input only
	blank: Protection does not change
	other: File may be used for input only
new expdate	New expiration date in the form yymmdd (year, month, and day).

The modified file label identification must be unique.

If an error occurs, *DEF does not modify the file definition, but terminates the job making the call.

If wait is requested and the file is open for data transmission in another job, the call will be completed as soon as the file is closed.

4.1.3 EXPAND

The operator may increase the mass storage space for an existing definition through the expand function of *DEF. The operator specifies on the control card how many additional blocks are needed; he may specify Class B device numbers to which additional blocks may be allocated.

```
$*DEF( E, wait, owner, filename, edition, acsc, mdsc, nbks, seg, dn,•••, dn_x )
```

owner	Owner of file to be expanded
filename	Name of file to be expanded
edition	Edition number of file to be expanded
acsc	Access security code
mdsc	Modification security code
nbks	Number of blocks to be added to the file. The sum of nbks and the current number of blocks specified in the label must not exceed 8388607.

seg	C: Added blocks must be contiguous with each other but not necessarily with the original file. S or other: Added blocks may be segmented.
dn	Up to nine Class B device numbers to be used for allocation of the file in the order listed. Each device is assigned a number when it is entered in the configuration (4.2.4). New device numbers must be for the same device type as the original file. If no device number is specified and the original definition was on Class B, the expand will use the same device numbers as were used in the allocate. If the original definition was on Class A, expansion will be on Class A.

If an error occurs, *DEF does not expand the file definition but terminates the job making the call.

4.1.4 RELEASE

The operator uses the *DEF release function to remove the definition of a file from MASTER file directories or to remove unused space from the definition. Released space returns to available mass storage.

```
$*DEF( R, wait,owner, filename,edition, acsc, mdsc, amount )
```

owner	Owner of file being released or truncated
filename	Name of file
edition	Edition number of file
acsc	Access security code
mdsc	Modification security code
amount	ALL: Release the definition and all its associated mass storage UNUSED: Release all blocks above last block written n: Number of trailing blocks to be released (1 to 8388607). If amount is zero or blank, *DEF handles the call as a no operation.

If the file to be released is open in another job, the call will be completed as soon as the file is closed.

If an error occurs, *DEF does not release the file, but terminates the job making the call.

4.1.5 OPEN

Before data transmission can take place on a file, the operator must call the *DEF open function. The open function opens Class A and Class B mass storage. MASTER places all mass storage devices associated with a file definition on-line during an open unless a partial open is specified.

```
$*DEF( 0,wait,dsi,owner,filename,edition,acsc,usage,S, block, B )
```

dsi — 1-4 alphanumeric character data set identifier. A dsi may not be blank or start with *. This abbreviated dsi takes the place of the definitions total identification in most references to the file made by the user on control cards and macro requests.

owner — Owner of file being opened

filename — Name of file

edition — Edition number of file

acsc — Access security code

usage — 0: File may be used for output or input and output.

I or other: File may be used for input only. Usage must not conflict with protection specified in the definition (4.1.1 and 4.1.2).

S — Definition is to be partially opened. When S is omitted, the definition is to be opened normally.

block — The parameter appears only with S. The device containing the specified block number (1 to 8388607) is to be placed on-line.

B — Bypass all input and output requests. When B is omitted, input and output requests are to be processed normally.

S and block appear only for partial openings of a definition. Only one device associated with a definition is placed on-line at a time. When it becomes necessary to open a new segment of a definition, and the new segment is on a different device, the operator calls the open segment function (4.1.6) and specifies a new block number on the device.

If drive requirements for definitions on Class B storage are not scheduled (3.1.3), the open function is rejected. When a call is for a device that is not on line, *DEF types the request for the operator to mount the device on the assigned drive (2.2) and waits for the operator to respond.

Only definitions allocated on Class B mass storage can be opened partially. Definitions on Class A are always on-line. An attempt to partially open Class A storage is handled as a normal request. If the dsi in the open matches a dsi already open in the same job, the conflict causes the job to be terminated. If the dsi in the open matches a dsi open in another job, there is no conflict; MASTER is able to distinguish between dsi's of different jobs.

If an error occurs, *DEF does not open the file but terminates the job making the call.

4.1.6 OPEN SEGMENT

When it becomes necessary to open a new segment of a definition following a partial opening of a definition (4.1.5), and the new segment is on a different device, the operator calls the open segment function and specifies a new block number on the device.

```
$*DEF( S, wait,dsi,block)
```

dsi Data set identifier of definition previously opened with partial open (4.1.5)

block A block number in the segment to be placed on-line. Block may be from one to the highest block number allocated to the file.

Segments of partially open files must not have blocks that cross from a segment on one device to a segment on another because the MASTER Input/Output Control System is unable to read them.

When an open segment request is for a device that is not on line *DEF types a request for the operator to mount the device on the assigned drive (2.2) and waits for a response.

4.1.7 OPEN UNIT

The *DEF open unit function associates a data set identifier with a unit record device (card reader, punch, printer, tape, etc.) making the unit available for I/O. When MASTER processes a valid open unit call, it logs the request on the console typewriter and waits for the operator to confirm that he has readied the unit (2. 5).

```
$*DEF( U, wait, dsi, dt, reel, id, filename, edition,
                              reel no, usage, B )
```

dsi 1-4 character data set identifier to be associated with unit device; first character may not be *.

dt Type of unit record device:

405 Card reader (must be more than one card reader in configuration)

415 Card punch

501
505 Line Printer

603
604
606
607 Magnetic tape unit

The following parameters appear only if dt is a magnetic tape unit and should be null (only comma appears) for other unit record devices. They can also be null when the tape to be mounted is a scratch tape.

real id 1-5 alphanumeric character reel identifier of tape to be mounted

file name 1-4 alphanumeric character file name that corresponds to name given on label of tape to be mounted

edition 1 or 2 alphanumeric character edition number that corresponds to edition number given on label of tape to be mounted.

reel no 1 or 2 alphanumeric character reel number that corresponds to the reel number given on label of tape to be mounted.

The following parameter is significant only for an open unit request which specifies one or more of the four preceding parameters. It is assumed 0 for scratch tapes:

usage O: Input and/or output

I or other: Input only

The following parameter may apply to any unit record device:

B Bypass all input/output requests. When B is omitted, input and output requests are to be processed normally.

When dt is magnetic tape, *DEF interprets the magnetic tape parameters as follows:

1. If reel id, file name, edition, and reel no are blank, *DEF assumes a scratch tape with no label. It automatically sets usage to 0 (input and/or output).
2. If file name, edition, and reel number are not null, MASTER checks them for validity against those in the standard magnetic tape label on the mounted tape. If file name, edition, and reel number are all blank in the open request, *DEF assumes that the tape to be mounted is not labeled and sets usage according to the parameter given in the request.
3. If the tape is unlabeled, it remains at load point after it is mounted. After a labeled tape is mounted and its label is verified, it is positioned to the first record following the label.

MASTER provides no other handling of standard magnetic tape headers and trailer labels.

4.1.8 CLOSE

The close function of *DEF removes a definition of a file or unit record device from MASTER tables making the definition or unit record device unavailable for data transmission.

```
$*DEF(C,wait,dsi)
```

dsi Data set identifier of file to be closed. If dsi is blank, *DEF closes all files associated with the job making the request, except INP, OUT, PUN, and job scratch.

A request to close a system or job file is a no-operation. If an error occurs, *DEF does not close the file but terminates the job making the call.

4.1.9 *DEF ABORT CODES

When a *DEF function is called by a control card and a condition arises resulting in termination of the operation, a message of the following form is written in the header on the job's OUT file:

```
VAC=xx      message
```

xx	Message
51	FUNCTION ER
52	DEVICE ERR
54	FILE SIZE
55	FILE ID ERR
56	FLD FULL
57	DEV NO. ERR
58	NO SPACE
59	SEG. NO. ERR
60	NO CONTIG.
61	NO SUCH ID
62	FILE OPEN
63	SECURITY ER
64	DSI ERR
65	TABLE SPACE
66	FILE BUSY
67	USE ERR
69	SCHED. ERR
70	VALUE ERR
71	2ND OPEN
72	SE, NO DRIV
73	BLOCK ERR
74	HRDWARE ERR
75	ILLEGAL DSI

The exact nature of the error depends to a large extent on the function called. If xx = 51, the *DEF card contains an illegal function code (not A, M, E, R, O, S, U or C).

4.2 *FMU

The File Maintenance Utility task includes routines that permit the operator to:

- List all or part of the Mass Storage Directory entries
- List all or part of the labels on the File Label Directory
- Enter a new disk pack, drum or disk file into the Mass Storage Directory
- Delete an entry from the Mass Storage Directory
- List all files having expired dates
- Rearrange files by dumping them one at a time and reloading them more compactly at new locations
- Make more mass storage available by dumping files and reloading them when the need for additional storage has passed
- Generate a back-up copy of a file

To run a routine, the operator submits a *FMU control card as part of a job. The task name for the routines is *FMU. Task Name control cards have the general form:

```
$*FMU (parameters)
```

Parameters specify routines and their options. They must appear in the order shown. A comma must appear for an unspecified field if any fields appear to its right. Otherwise, the right parenthesis suffices.

A *FMU control card may be placed after a SCHED card or a Task Name control card (another *FMU card, *DEF card, etc.).

4.2.1 LIST MSD

```
$*FMU (LIST,dsi,MSD,dt,dn)
```

dsi An open file to receive listing (e.g., OUT). If dsi is omitted, listing is written on job OUT file. dsi must not refer to a segmented file.

dt Device type:

852
853
854
813
863

dn Device number

To obtain a listing of the entire MSD, omit dt and dn.

To obtain a listing of all devices of type dt, omit dn.

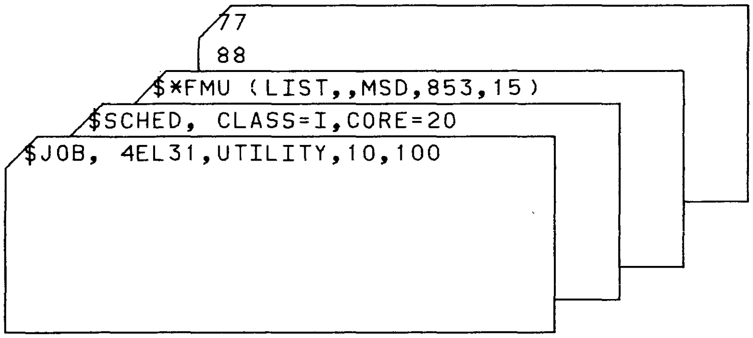
To obtain a listing of all devices with device number dn, leave dt blank.

To obtain a listing of an entry for a single device, specify both dt and dn.

When dsi is OUT, the listing is automatically printed; otherwise the XFER routine must be used to print it. For dsi other than OUT, *FMU sets up a 320 word output block.

When dsi refers to magnetic tape, each file generated by a LIST function is followed by an end-of-file.

Each entry is printed on a single page such as shown in the example.



```
77
88
$*FMU (LIST,,MSD,853,15)
$SCHED, CLASS=I,CORE=20
$JOB, 4EL31,UTILITY,10,100
```

```
JOB,4EL31,UTILITY,10,100
SCHED,CLASS=I,CORE= 30
*FMU(LIST,,MSD,853,15)

    MASS STORAGE DIRECTORY INFORMATION LIST      DATE  01/15/67

DEVICE IDENTIFICATION-

    EXTERNAL IDENTIFIER               MASTER
    DEVICE TYPE                          853
    DEVICE NUMBER                         15
    DEVICE CLASS                     CLASS A
    RECORDING MODE                    SECTOR
    LOW MS10 ADDRESS                       0
    HIGH MSIO ADDRESS                  15984
    STORAGE  CAPACITY (TRACKS)          1000
    SPACE ALLOCATED (TRACKS)            1000
    SPACE AVAILABLE (TRACKS)               0

DEVICE STORAGE MAP -

      FROM        THROUGH      NUMBER OF      RUNNING       RUNNING
      TRACK        TRACK        TRACKS         TOTAL         TOTAL
      NUMBER      NUMBER     USED   OR FREE     FREE          USED

           0          999     1000   USED                      1000
```

4.2.2 LIST FLD

```
$*FMU (LIST,dsi,FLD, owner, file name, edition)
```

dsi — Name of open file to receive listing (e.g., OUT). When dsi is omitted, listing is written on job OUT file. dsi must not refer to a segmented file.

owner — Name of person listed as owning file in FLD

file name — Name of file as it appears in FLD (assigned when file definition was allocated or modified)

edition — Edition number of file

To obtain a listing of the entire FLD, omit owner and name. When owner and file name are omitted but edition is specified, edition is ignored.

To obtain a listing of all labels of files belonging to the owner, omit file name and edition.

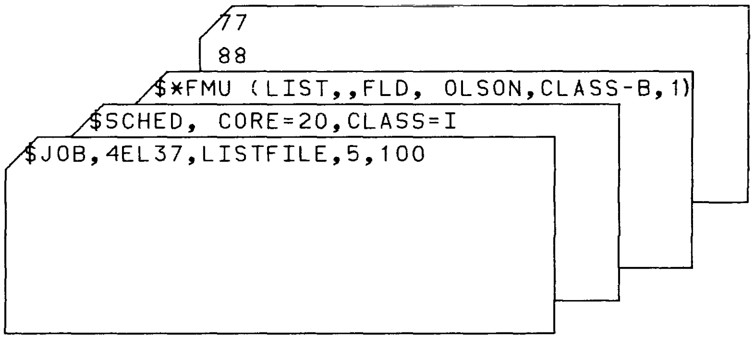
To obtain a listing of all labels having the same name, omit owner and edition.

To obtain a single listing of a particular label, specify owner, file name, and edition.

When dsi is OUT, the listing is automatically printed; otherwise the XFER routine is necessary to print it. For dsi other than OUT, *FMU sets up a 320-word output block.

When dsi refer to magnetic tape, each file generated by a LIST function is followed by an end-of-file.

Each entry is printed on a single page as shown in the example.



```
77
88
$*FMU (LIST,,FLD, OLSON,CLASS-B,1)
$SCHED, CORE=20,CLASS=I
$JOB,4EL37,LISTFILE,5,100
```

```
JOB,4EL37,LISTFILE,5,100
SCHED,CORE=20,CLASS=I
*FMU(LIST,,FLD,OLSON,CLASS-B,1)
                                                   DATE 01/15/67

        FILE LABEL LISTING FROM FILE LABEL DIRECTORY

FILE IDENTIFICATION-

    FILE NAME       CLASS-B
    OWNER IDENTIFICATION    OLSON
```

```
FILE DESCRIPTION.

    EDITION                            1
    CREATION DATE                1/15/67
    EXPIRATION DATE              1/15/67
    DATE LAST ACCESSED            / 0/
    NUMBER OF TIMES USED               0
    STORAGE MODE                  SECTOR
    FILE SIZE (TRACKS)               900
    BLOCK SIZE (CHARS)               259
    TOTAL ALLOCATED BLOCKS          7200
    HIGHEST BLOCK WRITTEN              0
    NUMBER OF SEGMENTS                 1

SEGMENT DESCRIPTION-

    SEGMENT    1

      DEVICE TYPE                    853
      DEVICE NUMBER                   16
      SEGMENT LENGTH (TRACKS)        900
      LOW SEGMENT LIMIT               16
```

4.2.3 ADD DISK PACK

```
$*FMU (ENTER,dt ,dn ,mode , class,CLEAR,lma , hma ,exid)
```

dt Device type:

852
853
854

dn Device number (1 to 262143)

mode T: Track
S or omitted: Sector

class A or omitted: Class A mass storage device
B: Class B mass storage device

CLEAR Appears when zeros are to be written on the disk pack; otherwise, CLEAR is omitted and no zeros are written.

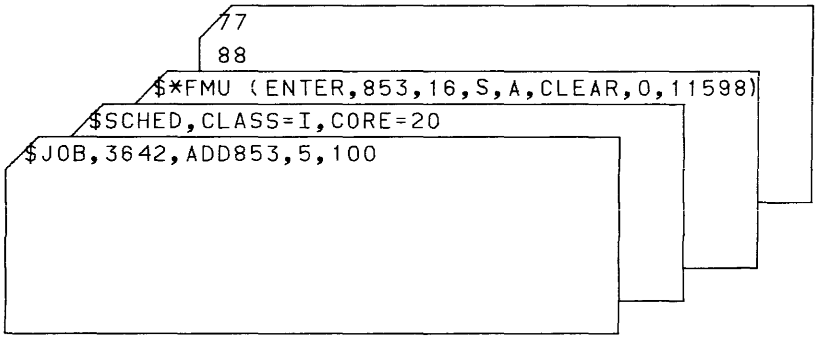
lma,hma Lowest and highest track addresses of MSIO accessibility on the disk pack. If either is omitted, its extreme limit is assigned.

exid Optional 6-character external identifier

The device being entered need not be scheduled for *FMU. When *FMU interprets this card, it logs channel, equipment, and unit on the console typewriter. The operator may comply with or refuse the request. (See logging, section 2.2.)

When the new disk pack is mounted and *FMU resumes execution, it writes a device label on the new pack and, if CLEAR is specified, writes zeros on it. If the operator refuses the request to mount the device, *FMU execution is terminated.

Example:



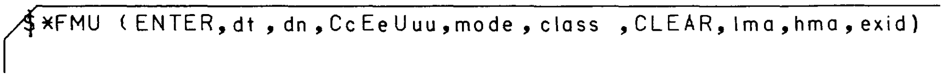
```
77
88
$*FMU (ENTER,853,16,S,A,CLEAR,0,11598)
$SCHED,CLASS=I,CORE=20
$JOB,3642,ADD853,5,100
```

If, during the CLEAR cycle, an irrecoverable write error is encountered, *FMU performs a voluntary abort with abort code = 4 (irrecoverable write error).

Logging:

```
Rr *FMU LOGGING
853,     16,C1E2U01
END LOG *FMU
```

4.2.4 ADD DRUM OR DISK FILE



```
$*FMU (ENTER,dt ,dn,CcEeUuu,mode , class  ,CLEAR,lma,hma,exid)
```

dt   Device type:

813
863

dn   Device number (1 to 262143)

CcEeUuu Hardware connect code:

c Channel to which new device is connected (1 to 7), c cannot be a real-time (dedicated) channel because real-time devices are not entered in the MSD.

e Equipment to which new device is connected (0 to 7)

uu Unit number of new device (0 to 77_8)

mode T: Track
S or omitted: Sector

class A: Class A mass storage device (An 813 and 863 must be Class A).
B or omitted: Class B mass storage device

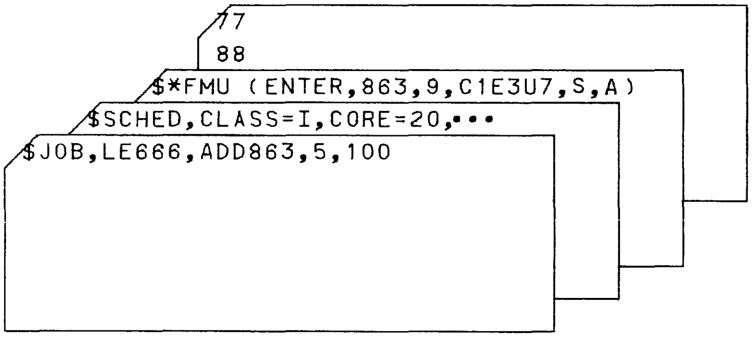
CLEAR Appears when zeros are to be written on the disk file or drum; otherwise, CLEAR is omitted.

lma,hma Lowest and highest address of MSIO accessib ility on the drum or disk file. If either is omitted, its extreme limit is assigned.

exid Optional 6-character external identifier.

When *FMU interprets this card, it enters the new device in the MSD, writes a device label on the device, and if CLEAR is specified, writes zeros on it.

Example:



```
77
88
$*FMU (ENTER,863,9,C1E3U7,S,A)
$SCHED,CLASS=I,CORE=20,•••
$JOB,LE666,ADD863,5,100
```

If, during the CLEAR cycle, an irrecoverable write error is encountered *FMU performs a voluntary abort with abort code = 4 (irrecoverable write error).

4.2.5
DELETE MSD ENTRY

```
$*FMU (DELETE,dt,dn)
```

dt Type of device

dn Number of device

Both parameters must be specified.

*FMU rejects requests for deletion of a device to which files are allocated, with the message on OUT file, DELETE BYPASSED -- ALLOCATED TRACKS ON DEVICE.

4.2.6
LIST EXPIRED FILES

```
$*FMU (PURGE)
```

When *FMU interprets this card, it scans the FLD and lists on the job OUT file all expired files according to owner, file name, and edition. It does not in any way release or alter the files.

Sample Output:

```
                                                                 DATE  01/03/67
PAGE 1
      THE FOLLOWING FILES HAVE ATTAINED OR EXCEEDED THEIR EXPIRATION DATE


FILE OWNER              FILE NAME                    FILE EDITION NO.

EMB               RESA                                       1
GARDNER           $FILE                                      1
BT2               TSKA                                       1
KAW               TEST                                       30
                  CON8                                       1
GAYLEN            ALPHA1                                     AA
GAYLEN            ALPHA2                                     VI
                  BIG1                                       IX
OLSON             CLASS-B                                    1

                               **** END****
```

4.2.7 DUMP AND LOAD FILES

The dump and reload options in *FMU permit the operator to create a back-up copy of a file, make additional file space available for a job with high storage requirements, re-arrange files by temporarily removing them from mass storage and reloading them more compactly, etc.

The schedule card for *FMU dump and load options must request two segments of scratch.

To dump a file, use the *FMU cards:

```
$*FMU (DUMP,rdsi,wdsi)
```

rdsi dsi of open file to be dumped

wdsi dsi of open file to receive dump

The dumped file, rdsi, is not released after it is dumped onto wdsi. For release, refer to *DEF (R,...), Section 4.1.

To reload the file, use the *FMU card:

```
$*FMU (LOAD,rdsi,wdsi)
```

rdsi dsi of open file containing the file to be loaded.

wdsi dsi of open file to be loaded

The dump file, rdsi, is not released after its data is reloaded onto wdsi.

LOAD rules:

1. The data set identifiers, rdsi and wdsi, may not begin with an asterisk.
2. Files to be loaded from a multi-file source must be loaded in the same relative order in which they were dumped, so that multi-reel dump files on magnetic tape need not be rewound and searched for each file to be loaded.
3. A file must be reloaded onto the same hardware type from which it was dumped.

4. When a file is to be loaded, it must first be allocated with the following features compatible with the file as it existed at the time it was dumped:

   Access and modification security codes must be identical

   Highest block written field of the newly allocated file must be zero

   The number of allocated blocks on the newly allocated file must be greater than or equal to the corresponding field of the file as it existed at the time it was dumped.

   The block size of the dumped and newly allocated files must be identical.

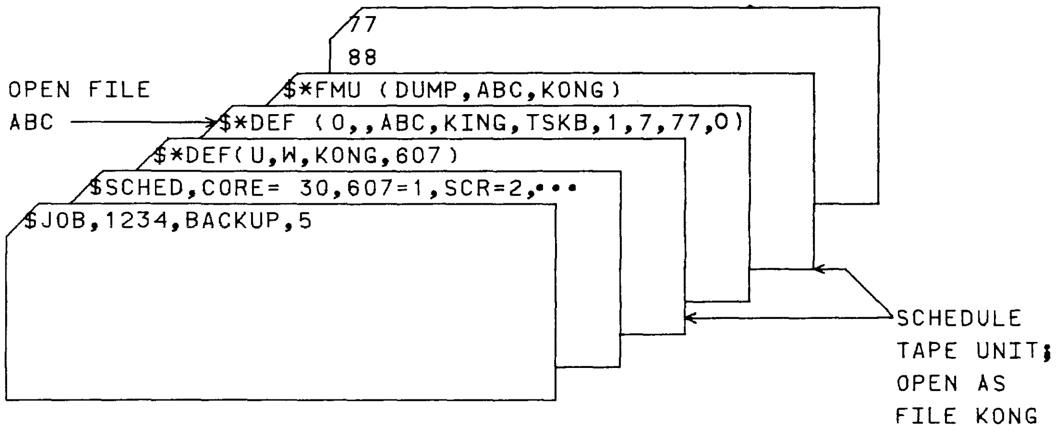
   The modes of the file previously dumped and the one newly allocated must be identical.

5. Load should not be run in a multiprogramming batch.

DUMP rules:

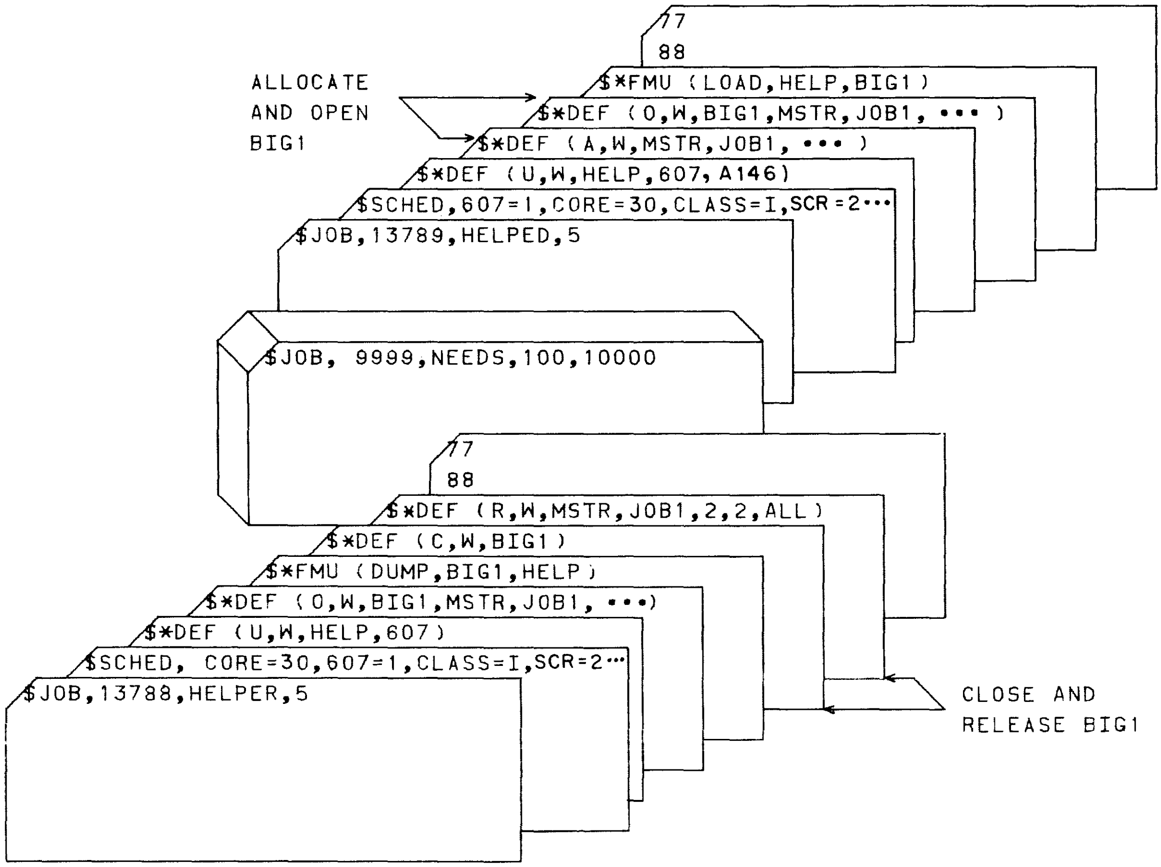
1. Dump should not be run in a multiprogramming batch.
2. The dump receiving file must be on mass storage or magnetic tape.

GENERATE BACKUP COPY
OF FILE ABC



```
77
88
$*FMU (DUMP,ABC,KONG)
$*DEF (0,,ABC,KING,TSKB,1,7,77,0)
$*DEF(U,W,KONG,607)
$SCHED,CORE= 30,607=1,SCR=2,•••
$JOB,1234,BACKUP,5
```

MAKE ADDITIONAL SPACE
AVAILABLE FOR JOB 9999



```
77
88
$*FMU (LOAD,HELP,BIG1)
$*DEF (O,W,BIG1,MSTR,JOB1, ••• )
$*DEF (A,W,MSTR,JOB1, ••• )
$*DEF (U,W,HELP,607, A146)
$SCHED,607=1,CORE=30,CLASS=I,SCR=2•••
$JOB,13789,HELPED,5

$JOB, 9999,NEEDS,100,10000

77
88
$*DEF (R,W,MSTR,JOB1,2,2,ALL)
$*DEF (C,W,BIG1)
$*FMU (DUMP,BIG1,HELP)
$*DEF (O,W,BIG1,MSTR,JOB1, •••)
$*DEF (U,W,HELP,607)
$SCHED, CORE=30,607=1,CLASS=I,SCR=2•••
$JOB,13788,HELPER,5
```

- FILE BIG1 WILL BE TEMPORARILY STORED ON MAGNETIC TAPE WITH DSI HELP
- ONCE RELEASED, BIG1 CANNOT BE USED BY JOBS IN THE MULTIPROGRAMMED BATCH
- JOB 9999 IS INITIATED AS SOON AS STORAGE IS AVAILABLE
- JOB 13789 RETURNS THE FILE TO MASS STORAGE WHEN STORAGE IS AVAILABLE

4.2.8 *FMU ERRORS

VOLUNTARY ABORT CODES

When *FMU is abnormally terminated, it writes one of the following voluntary abort codes in the heading on the OUT file in the form:

```
VAC=*FMUx
```

x	Significance
0	*FMU not in Primary Task List
1	Cannot open next segment on input media
2	Cannot open next segment on output media
3	Irrecoverable read error
4	Irrecoverable write error
5	End of allocated area on write file
6	SURCH subroutine failed
7	Cannot find dsi in Primary Task List
8	Sequence error on read file

BLOCKER/DEBLOCKER CODES

If abnormal termination is caused by an error in the Blocker/Deblocker, the following error is written on the OUT file:

```
PACK
PACKD ERROR x
```

where x is the error code as returned in the Q register by the Blocker/Deblocker. See MASTER Reference Manual.

NON-ABORT MESSAGES

When *FMU detects an error that does not cause its termination, it writes one of the following messages on the OUT file:

```
CONTROL CARD ERROR
DELETE BYPASSED -- ALLOCATED TRACKS ON DEVICE
ERROR - *MSD NOT IN PRIMARY TASK LIST
SYSTEM ERROR -- MSD-DEVTYP HARDWARE NON-COMPARE
DT/DN ERR (hhhh DEVNO. NOT UNIQUE)
```

```
DEVICE TYPE NOT IN TABLE/HARDWARE
CANNOT LOCATE DEVICE TYPE IN MASS STORAGE TABLE
MST-UNIT TABLE HARDWARE TYPE NON-COMPARE
ENTER ERROR - NO CHANNEL AVAIL. IN UNIT TABLE
ASSIGNED HARDWARE CONFIGURATION NOT LEGAL
ERROR-ILLEGAL DSI
ERROR-DSIS ARE IDENTICAL
ERROR-UNDEFINED DSI
ERROR-FILE LABELS INCONSISTENT
INSUF SCR SCHED
```

4.3 XFER

The MASTER operator can call from the library a general-purpose copy routine (XFER) that enables him to create, maintain, and dispose of standard and non-standard files. XFER transfers a file from one medium to another. It is useful for conversion of non-standard files to MASTER standard blocked format, transfer of tasks to files, preparation of large volumes of data, transfer of a file to special forms on the printer or punch, etc.

4.3.1 CALL XFER

To call XFER, create a job containing the XFER Task Name control card.

```
$XFER(idsi,n,odsi,m,f,SS,mode,N)
```

idsi Data set identifier of the input file. When idsi is not specified, the input file is INP.

n Number of words per input block (its value may not exceed standard block size). When n is not specified, block size is standard.

odsi Data set identifier of output file. When odsi is not specified, the output file is OUT.

m Number of words per output block (its value may not exceed standard block size). When m is not specified, block size is standard.

f A form number defined at an installation, 1 to 8 alphanumeric characters, identifying a punched card form, printer form, or printer tape format to be mounted for odsi.

SS Appears only when odsi is a printer and indicates that printer carriage control is single space. When SS is omitted, carriage control is program-controlled.

mode Specifies mode of output; overrides mode parity of odsi.

BN	Binary
BD	BCD

When mode parameter is omitted, output is in mode of input.

N Appears for a mass storage file when idsi was prepared by other than a standard blocking routine (PACK) or the input backgrounder. Otherwise, N is omitted. N specifies standard error recovery if input is mass storage, since special recovery is performed for records prepared by PACK or the input backgrounder.

Parameters must appear in the order shown. A comma must appear for an unspecified field if any fields appear to its right. Otherwise, the right parenthesis suffices. For example:

```
$XFER   (TAPE,,PRNT)
```

Using limited blocking and deblocking, XFER copies file idsi until it reads an end-of-file. Both files must have been opened prior to the transfer using *DEF cards. Also any unit devices required must have been scheduled on the SCHED card (3.3); for example, a tape drive and printer must be scheduled for a tape-to-print operation. To transfer directly from the card reader, the job must be declared a DIRECT job (3.3).

When a transfer is unsuccessful, the job containing the card is terminated; a message is typed on the console typewriter and written on the job's OUT file. If idsi or odsi is a system scratch file, it is released.

4.3.2 SPECIAL FORMS

The printing or punching of files requiring special forms such as checks represents one of the most useful applications of XFER. Instead of putting the print file on OUT for automatic printing by the output backgrounder, which makes no allowance for special forms, the operator may create a special output file and call XFER to print it.

An installation can also designate form numbers that specify changes of format tape for the printer. When the f parameter is used, the operator is directed to change to the specified form with the message:

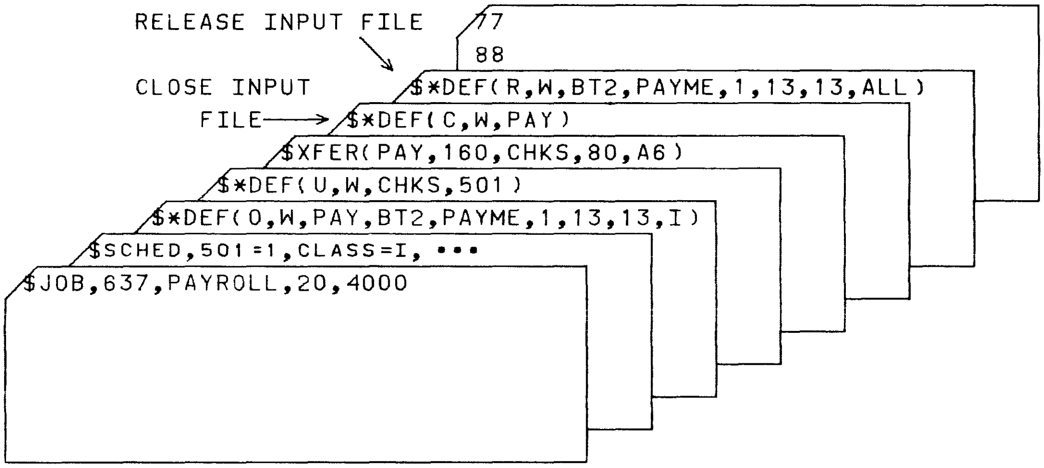
Rr XFER 001 (MOUNT FORM f ON ht CcEeUuuu)

The operator may comply with or refuse the request (2.4).

When the transfer is complete, the operator is directed to return to the standard form for the installation with the message:

Rr XFER 002 (REMOVE FORM f FROM ht CcEeUuuu)

PRINT FILE ON SPECIAL FORMS



```
77
88
$*DEF(R,W,BT2,PAYME,1,13,13,ALL)
$*DEF(C,W,PAY)
$XFER(PAY,160,CHKS,80,A6)
$*DEF(U,W,CHKS,501)
$*DEF(O,W,PAY,BT2,PAYME,1,13,13,I)
$SCHED,501=1,CLASS=I, •••
$JOB,637,PAYROLL,20,4000
```

4.3.3 XFER ERROR CONDITIONS

Conditions detected by XFER that result in termination of the task produce the following message on the console typewriter:

D JOB i XFER 003 (SEC=xx XEC=yy)

xx System error codes
See MASTER Reference Manual, error processing, for both mass storage and unit record devices.

yy XFER error codes

01	disk format error
02	XFER reached end of allocated file
03	file lockout condition

04 XFER unable to define abnormal condition
05 write attempted on read-only file
06 end-of-tape condition
07 irrecoverable error
10 undefined dsi; illegal hardware type request for I/O (for example, a request to read from the printer); or illegal control card parameter
11 the deblocker is attempting to pass XFER a blocked record with an illegal logical record header.
12 an irrecoverable read error has occurred while attempting to read XFER in from the library
13 the operator has refused to mount special forms in the printer, or the caller of XFER has specified the use of special forms for a device other than the printer or punch.

4.3.4 BLOCKING/ DEBLOCKING CONVENTIONS

When input and output block sizes differ, some form of blocking takes place on an XFER. Deblocking is restricted to files being transferred to the printer or punch in which case the input file is assumed to be in MASTER standard blocked format.

Deblocking takes place if specified input block size is larger than specified output block size (n > m). Deblocking must be specified if blocked records are to be printed or punched using on-line printers or punches.

Blocking takes place if specified input block size is smaller than specified output block size as shown below (n < m):

1. Card reader

   If the specified input block size is less than the specified output block size, blocking will be performed. 40-word reads are given, and the hardware determines whether the card is binary or BCD: The count is adjusted accordingly. The block is filled until the remaining area defined by output block size cannot hold another 40-word image. (If card images are being read from the INP file using a direct card reader, blocking must be specified (n < m) if the card images are to be blocked on the output file.)

2. Magnetic tape

   The input block size specifies the largest record appearing on the tape. XFER uses this value in the request itself, although it uses the word count of the read to determine the actual record length. The output block is filled until the remaining area cannot hold another record of the specified input size.

3. Disk

   The input block size specifies the actual size to be read. The output block is filled until the remaining area cannot hold another input block.

SYSTEM-TO-OPERATOR MESSAGES 5

5.1 FORMAT

MASTER communicates with the operator through the console typewriter, and most system messages are in standard format:

xy JOB i ssss nnn (des)

xy Message type:

D Destructive
I Informative
A Operator action required, no response
Rr Operator decision and response required; r (0-9) is the message number assigned by the system (appears optionally)

i Job identifier taken from JOB card

ssss Name of task; 4 characters maximum

nnn Message number relative to the task

(des) Optional description enclosed in parentheses; maximum 65 characters

The System Executive and all tasks associated with MASTER may type non-destructive messages. MASTER types a job abort message when a system error or malfunction occurs external to the user's task, and job termination results.

File Logging

When *DEF or *FMU opens a file, it logs hardware assignments on the console typewriter. The logging format is given in section 2.2.

Begin and Terminate Messages

B i is typed when Job i is initiated.

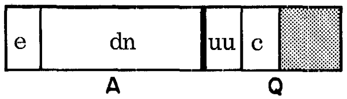
T i is typed when Job i is terminated.

System Messages

The following table lists all job abort and non-destructive messages typed on the console typewriter. Symbols in the messages are defined as follows:

c	channel number
e	equipment number
ht	hardware type
uuu	unit number

5.2
ACTION MESSAGES

MESSAGE TO OPERATOR					Cause	Action
Type	Job	Task	nnn	Optional Message		
A		*BKO	01		Irrecoverable error in reading a punch file block from mass storage. An error card is punched and offset in the deck; punching continues. Job identifier is unknown.	Mark deck as having an error card in it. The deck is useless since information is missing. The job should be re-run. If this occurs repeatedly, the system punch file should be re-allocated.
A		BOOT	02	(DEV HAS NO *LAB)	Device that should have the File Label Directory has no entry for FLD address in device label. Computer halts.	1. Set (AQ) = new DVNT entry for *LAB as shown:  e 6-bit equipment number dn device number $\leq 777777_8$ uu 6-bit unit number c 3-bit channel number 2. Press GO
A		EXEC	101	(ht CcEeUuuu CR ESxxxx ISxxxx) RE, UNIT	Connect reject ESxxxx 12-bit external status (CDC 3000 Series Peripheral Equipment Reference Manual ISxxxx 12-bit internal status (CDC 3300 Computer System Reference Manual)	Dial in correct equipment and unit and press FINISH. If they are already correctly dialed, type in DWN/ and press FINISH. The system will mark the unit as inoperable and continue processing.

MESSAGE TO OPERATOR					Cause	Action
Type	Job	Task	nnn	Optional Message		
A		EXEC	103	(ht CcEeUuuu RDY)	Hardware not in ready status.	Place unit in ready status.
A		EXEC	161	(CR CcEeUuuu CMP ERR)	Read compare error on last card read.	Remove last card from output stacker and place it as first card in the input stacker. Ready card reader and it will re-read the card.
A		SINT	001	(CAN'T CON ON dn)	Connect rejected on device number dn.	In all cases, computer halts. Press GO to retry I/O operation. If retry is unsuccessful, reautoload. If autoload is not successful, a GLIB run may be necessary.
				(CAN'T INP ON dn)	Input instruction reject on dn.	
				(CAN'T OUT ON dn)	Output unit reject on dn.	
				(CAN'T SEL ON dn)	Select rejected on device number dn.	
A		SINT	002	(ERROR ON RD DEV. NO=dn STATUS=ssss ADDR=aaaaa)	Error ssss occurred on read operation initiated at address aaaaa on device number dn.	In either case, computer halts. Press GO to retry operation.
				(ERROR ON WR DEV. NO=dn STATUS=ssss ADDR=aaaaa)	Error ssss occurred on write operation initiated at address aaaaa on device number dn. ssss hardware status for device. See CDC Peripheral Equipment Manual.	

5.3 DESTRUCTIVE MESSAGES

MESSAGE TO OPERATOR					Cause	Action
Type	Job	Task	nnn	Optional Message		
D		EXEC	00	(MEMORY PARITY ERROR)	System incurred a memory parity error.	Restart computer.
D	JOB i	*BKI	02		Source deck of Job i is too big for mass storage. Job is not run.	Return job to programmer or resubmit as DIRECT job.
D		BOOT	001	(CAN'T FIND ffff LABEL)	BOOT cannot find label for ffff on File Label Directory.	Computer halts. If ffff is *LAB, *IDF, or *MSD, re-install. If ffff is *LIB, or *DIR, ask for new edition on autoload.
D	JOB i	*DEF	56	(MAXIMUM FILE COUNT EXCEEDED)	Job i was terminated because it requested that a file be allocated when File Label Directory was full.	Release expired files and resubmit Job i.
D	JOB i	*DEF	77	(I/O ERROR ON *LAB/*MSD/*IDF)	Job i was terminated because *DEF encountered irrecoverable errors on label directory file, mass storage directory file, or label id file while processing a request.	If error persists, it may be necessary to recreate file.
D	JOB i	*DEF	aaaaa	IRREGULAR CONDITION	Highly improbable condition at address aaaaa in *DEF.	Consult systems analyst.
D	JOB i	*EST	000	(*LIB, name)	Job i was terminated because of irrecoverable read errors on library file during loading task name.	If error persists, it may be necessary to recreate library.

MESSAGE TO OPERATOR					Cause	Action
Type	Job	Task	nnn	Optional Message		
D	JOB i	*EST	001	(*LIB, name)	Job i was terminated because it attempted to call task name from library when task was not defined in library directory.	Return Job i to programmer.
D	JOB i	*EST	005	(*LIB, code)	Job i was terminated because a locate error occurred during load from library. Code is error code returned to *EST from the I/O system. (MASTER Reference Manual, Sec. 6.3.3)	
D	JOB i	*EST	07	(name)	Establisher detected checksum error when loading named task.	If name is a library task, it may be necessary to recreate library.
D	JOB i	JMTR	01		DIRECT, JOB, or SCHED card followed Task Name card.	1. Check to see if end-of-file card should have preceded JOB card. If so, JOB i probably ran successfully 2. Correct card sequence and resubmit job or return job to programmer.
D	JOB i	*SCH	00		JOB card is missing or unreadable; x is meaningless.	Examine input deck for end-of-file card not followed by JOB card.

MESSAGE TO OPERATOR					Cause	Action
Type	Job	Task	nnn	Optional Message		
D	JOB i	*SCH	01		No account number on JOB card. Job i will not be run.	Return Job i to programmer for corrections.
D	JOB i	*SCH	02		Illegal separator on JOB card. Job i will not be run.	Return Job i to programmer for corrections.
D	JOB x	*SCH	03		No job identifier on JOB card; x is meaningless. Job will not be run.	Return job to programmer for corrections.
D	JOB i	*SCH	04		Time field on JOB card not in correct form. Job i will not be run.	Return Job i to programmer for corrections.
D	JOB i	*SCH	05		Segments requested by SCR parameter, for mass storage scratch, exceed the system capacity. Job i will not be run.	Return Job i to programmer for corrections.
D	JOB i	*SCH	06		Space requested by CORE parameter for job exceeds system capacity. Job i will not be run.	Return Job i to programmer for corrections.
D	JOB i	*SCH	07		Peripheral requirements on SCHED card exceed the system capacity. Job i will not be run.	Return Job i to programmer for corrections.
D	JOB i	*SCH	08		JOB card OUT file limit exceeds maximum mass storage segment limit for a file.	Return Job i to programmer for corrections.

MESSAGE TO OPERATOR					Cause	Action
Type	Job	Task	nnn	Optional Message		
D	JOB i	*SCH	09		JOB card PUN file limit exceeds maximum mass storage segment limit for a file.	Return Job i to programmer for corrections.
D	JOB i	*SCH	10		A printer or punch called for on a DIRECT card was not scheduled on a SCHED card.	Return Job i to programmer for corrections.
D	JOB i	*SCH	11		SCHED card field unrecognized by job scheduler. Job i is not run.	Return Job i to programmer for correction.
D		SINT	000	(CAN'T FIND DEV TYPE dt IN LIST)	Device type table (DEVTYPE) contains no entry for device type dt. Computer halts.	Re-autoload.
D		SINT	004	(TBLE OVRFLOW DVNT)	No room in device number table (DVNT).	Consult systems analyst. A GLIB run may be required.
D		SINT	006	(CAN'T FIND dsi IN UNITTBL DT=dt)	System initialize routine cannot find entry for file dsi, with device type dt in the resident unit table (UNITTBL). Computer halts.	A GLIB run may be required. Re-autoload.
D		SINT	014	(LACK CORE)	Available core is insufficient for loading MASTER resident. Computer halts.	Generate library with MASTER resident small enough for existing core. Re-autoload using new library.

MESSAGE TO OPERATOR					Cause	Action
Type	Job	Task	nnn	Optional Message		
D		SINT	016	(ERR IN EXEC LENGTH)	Length of MASTER resident computed by initialize routine not same as length in *DIR SEPT table. Computer halts.	Check GLIB control cards and recreate library.
D		SINT	019	(LOC ERR ON *LAB)	Irrecoverable error on locate request on *LAB, the File Label Directory.	Press STOP to halt computer.
D		SINT	020	(RD ERR ON *LAB EC=eeee)	Irrecoverable error on read request on *LAB. eeee is status from MIOCS indicating type of error. (MASTER Reference Manual, Sec. 6.3.3)	If eeee indicates end-of-file condition, one or more system scratch files are probably not in File Label Directory. Press STOP to halt the computer.
D		SINT	021	(DEVTYPE SEARCH FAIL)	Device type table (DEVTYPE) does not contain entry for a device type specified for a system scratch file.	Press STOP to halt computer.
D		SINT	022	(SYSTEM FILE ERR)	System scratch files are not in sector mode, not all of same device type, or non contiguous.	Press STOP to halt computer.
D		SINT	025	(BKGFILE NOT CORR. TO FILEID)	BKGFILE length does not correspond to the length of the unit record area of the resident FILEID table. Computer halts.	

MESSAGE TO OPERATOR					Cause	Action
Type	Job	Task	nnn	Optional Message		
D	JOB i	XFER	003	(SEC=xx XEC=yy)	XFER requested termination. xx System code (MIOCS status, MASTER Reference Manual, Sec. 6.3.3) yy XFER code (4.3.3)	

5.4 RESPONSE MESSAGES

MESSAGE TO OPERATOR						
Type	Job	Task	nnn	Optional Message	Cause	Action
Rr		XFER	001	(MOUNT FORM f ON ht CcEeUuuu)		1. Mount requested form. 2. Press MANUAL INTERRUPT. 3. Type Rr, OK on console. 4. Press FINISH. If form not mounted: 1. Press MANUAL INTERRUPT. 2. Type Rr, NO. 3. Press FINISH.
Rr		XFER	002	(REMOVE FORM f FROM ht CcEeUuuu)		1. Remove requested form. 2. Press MANUAL INTERRUPT. 3. Type Rr, OK. 4. Press FINISH.
R		SINT	003	(DEV. d IS ON CcEeUuu)	Device number d is mounted on channel c, equipment e, unit uu.	1. After system types C, type correct channel number for device. 2. After system types E, type correct equipment number 3. After system types U, type unit number. 4. After system types), press FINISH

MESSAGE TO OPERATOR					Cause	Action
Type	Job	Task	nnn	Optional Message		
R		SINT	005	(REPEAT)	Last SET response typed by operator contains a format error.	1. Retype response correctly. 2. Press FINISH
R		SINT	007	(DATE)	MASTER date initialization	1. Type mmddyy (month, day, year) 2. Press FINISH
R		SINT	008	(TIME)	MASTER time initialization	1. Type hhmmss (hours, minutes, seconds; ss is optional) 2. Press FINISH
R		SINT	009	(SET)	Enable operator to change system parameters	To change parameters see Section 2.1. Otherwise, press FINISH when no parameters are to be altered.
R		SINT	010	(DIRCT)	Enables operator to release and reallocate standard files.	To allocate files see Section 2.1. Otherwise, press FINISH
R		SINT	015	(EDITN)	Enables operator to select edition of MASTER library	1. Type ee edition code of MASTER library 2. Press FINISH
R		SINT	018	(MOUNT DEV. NO. d)	Device d should be mounted.	Mount requested device. Press FINISH
Rr				CKSM ERR ON *MSD	Checksum error on Mass Storage Directory	Before answering, consult systems analyst. Correct response is Rr, mdsc mdsc master modification security code for the installation

5.5 INFORMATIVE MESSAGES

MESSAGE TO OPERATOR					Cause
Type	Job	Task	nnn	Optional Message	
I[††]	JOB i	*BKI	01		A parameter on DIRECT card following Job i is not in correct form. Parameter is ignored and normal processing continues.
I	JOB i	*BKI	03		Card following source deck of Job i was not JOB or DIRECT. Cards are passed up to next JOB or DIRECT card.
I	JOB i	*DEF	63	(FILE SECURITY CODE ERROR)	Task in Job i gave incorrect security code. NOTE: Repeated occurrence of this message may indicate that a user is attempting to tamper with the file security system.
I		EXEC	71	(ILL R-T INT xxxx)	Interrupt within EXEC was not real-time, MANUAL, or associated processor. xxxx interrupt code 00LCh[†] External interrupt 010Ch I/O channel interrupt 0110 Real-time clock interrupt 0111 Arithmetic overflow fault 0112 Divide fault 0113 Exponent overflow fault 0114 BCD fault 0115 Search/move interrupt 0116 Manual interrupt 0117 Associated processor interrupt 0120 Executive interrupt
I		EXEC	72	(ASSOC PROC INT)	Associated processor interrupt when no processor was present.

[†] L = Line 0-7 and Ch = channel designator, 0-7

[††] An informative message should be brought to the attention of the system's analyst or customer engineer.

MESSAGE TO OPERATOR					Cause
Type	Job	Task	nnn	Optional Message	
I		EXEC	102	(ht CcEeUuuu SRyyyy ESxxxx ISxxxx)	Select reject SRyyyy 12-bit select code ESxxxx 12-bit external status (CDC 3000 Series Peripheral Equipment Reference Manual) ISxxxx 12-bit internal status (CDC 3300 Computer System Reference Manual)
I		EXEC	104	(ht CcEeUuuu DWN)	Hardware inoperable. System or operator has removed unit from available list.
I		EXEC	105	(ht CcEeUuuu FLO FI=dsi JI=i)	Reference was made to a file on an inoperable unit. FI=dsi file identifier JI=i JOB identifier
I		EXEC	111	(UNDEF INT xxxx)	Interrupt was not expected by the I/O system. xxxx-interrupt code (See EXEC 71)
I		EXEC	112	(PAR ERR CHAN c)	Parity error occurred on channel c.
I		EXEC	151	(MT CcEeUuuu $\frac{F}{I}$ ESxxxx FI=dsi JI=i)	System recovery procedure cannot recover from error on magnetic tape. F - Error occurred on first try I - System has not recovered See EXEC 171 for ES, FI and JI
I		EXEC	162	(CP CcEeUuuu CMP ERR)	Punch compare error. Punch background routine automatically offsets the card in error and the following card and repunches both.

MESSAGE TO OPERATOR					Cause
Type	Job	Task	nnn	Optional Message	
I		EXEC	171	(DP CcEeUuuu $\substack{F\\I}$ A=xxxx ESxxxx ECxx DRx FI=dsi JI=i)	Error on mass storage device. F - Error occurred on first try I - System has not recovered A=xxxx Address at which error occurred ESxxxx 12-bit external status of unit (CDC 3000 Series Peripheral Equipment Reference Manual) ECxx - Error code (MIOCS code, MASTER Reference Manual) DRx - Driver code: 0 Five rejects on select or I/O instructions when EC51 is indicated. 1 Parameter to mass storage driver in (A) is not 0-4. 2 An illegal disk address was in the file call instance entry or the unit table entry. 3 Unit on which a new request is to be processed is busy. 4 The L3 string between the equipment and unit table entries is empty when an interrupt is still outstanding. 5 The unit performing last I/O operation is busy even though an end of operation interrupt was received. 6 The unit is busy after the record address register in the controller was reset. 7 The mass storage driver was entered because of a transmission parity error. FI=dsi File identifier JI=i JOB identifier
I		SINT	011	(aaaaa=cccccccc)	Contents of octal address aaaaa have been set to octal value cccccccc by a SET command (Section 2.1).

MESSAGE TO OPERATOR					Cause
Type	Job	Task	nnn	Optional Message	
I		SINT	012	(AV. CORE=nnnn QP)	nnnn quarter pages of core are available for non-resident tasks.
I		SINT	013	(eeeeeeee NOT FOUND)	System initialize routine cannot find system entry point eeeeeeee used in SET request. Entry point is not in SEPT table of *DIR. SET request is ignored.
I		SINT	017	(FMT ERR)	Control card SET request preceding this message contains format error. SET command is ignored.
I		SINT	023	(CHKSUM ERR FOR task name)	Initialize routine detected checksum error while loading resident task from library.
I		SINT	024	(TOO MANY R-T TASKS)	Number of real-time tasks exceeds limit set by RTLIMIT in SINT.

CONTROL DATA

CORPORATION

COMMENT AND EVALUATION SHEET

3300/3500 Computer Systems
MASTER Operator's Guide

Pub. No. 60178600 April, 1967

THIS FORM IS NOT INTENDED TO BE USED AS AN ORDER BLANK. YOUR EVALUATION OF THIS MANUAL WILL BE WELCOMED BY CONTROL DATA CORPORATION. ANY ERRORS, SUGGESTED ADDITIONS OR DELETIONS, OR GENERAL COMMENTS MAY BE MADE BELOW. PLEASE INCLUDE PAGE NUMBER REFERENCE.

FROM NAME : ____________________

BUSINESS ADDRESS : ____________________

NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

FOLD ON DOTTED LINES AND STAPLE

STAPLE STAPLE

FOLD FOLD

FIRST CLASS
PERMIT NO. 8241

MINNEAPOLIS, MINN.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION
Documentation Department
3145 PORTER DRIVE
PALO ALTO, CALIFORNIA

FOLD FOLD

STAPLE STAPLE

3300/3500 MASTER OPERATOR'S GUIDE

CONTROL DATA
CORPORATION

CORPORATE HEADQUARTERS, 8100 34th AVE. SO., MINNEAPOLIS, MINN, 55440
SALES OFFICES AND SERVICE CENTERS IN MAJOR CITIES THROUGHOUT THE WORLD

Pub. No. 60178600

Litho in U.S.A.