CSS Doc. #01-TM059
Progrmng. Ref. Manul. (DTSS)
10/31/79

System Programming Reference Manual

This document contains information necessary for system programming on the DTSS. Topics include Executive services and Job/Executive Interface.

Pub.1059

31 OCT 79

title System Programming Reference Manual

authors S. Garland
A. Dwyer
E. Pepper
A. Colvin
N. Bourbaki

publication number 1059

date 31 OCT 79 (revision 1)

category System Management — Programming

supersedes DTSS Programming Reference Manual, TM059 (1 OCT 74)

related documents GMAP Reference Manual, Pub.1077

File Access and Protection, Pub.1102

Description of File System and Job Trees, Pub.1335

System Management Publications List, Pub.1350

A postage-paid remarks form is included at the end of this manual for your convenience. Information about publications may be found in the Publication Price List and Publication Abstracts, of which single copies may be obtained free of charge.

The named authors would like to thank the many, many people who have authored, prepared, or worked on the various parts of this manual. Large portions of this manual come directly from the superseded edition by S. Garland and A. Dwyer.

Contents

Chapter 1

Job/Executive Interface

When a slave job is in execution, it has full control of a processor in slave mode and may perform any legal slave mode operations, as described in the "Honeywell 600/6000 Programming Reference Manual". Whenever a processor leaves slave mode, the Executive gains control.

A slave job may enter the Executive in various ways: it may request service from the Executive, it may execute an instruction that faults, or some external event (such as a physical I/O operation completing or timer running out) may cause the Executive to suspend the job. Executive entries that are not caused by a job are invisible to a job; when it resumes execution, neither its memory nor its environment will be changed.

The Executive conveys information to a slave job by the following means:

faults	indicating that the job executed an instruction that faulted
traps	indicating that asynchronous operations have completed
interrupts	usually indicating that another slave job wishes to communicate with this job

This sketchy description covers the entire interface between the Executive and a slave job, with the exception of special considerations in the case of job 1 (the Slave Loader) having to do with system startup). We now consider this interface in more detail.

1.1 Running in Slave Mode

A job actually running (executing instructions) is not affected by the Executive, except as noted here.

1.1.1 Base Address Register

The base address register (BAR) limits memory references by the slave job. The base portion of the register is set according to the location of the job in main memory, and may be reset by the Executive at any time that the job is interrupted (is not actually executing instructions). The length portion is set according to the length of the job, and changes only when the job issues a MEMORY.REQUEST call (on a MME fault, as described below).

1.1.2 Timer Register

The timer register is set according to the needs of the Executive, and may change any time the job is interrupted. The slave job should not therefore depend on the setting of the timer register.

1.1.3 Input and Output (I/O)

An I/O activity requested by the job may proceed concurrently with its execution. (I/O is requested with Executive calls, as described below.) A slave job should not re-use an I/O buffer until it has been notified by the Executive (by means of a trap) that the previously requested I/O has been completed on that buffer.

1.2 Job to Executive

1.2.1 Leaving Slave Mode

The Honeywell 600/6000 hardware provides two general methods for leaving slave mode: I/O interrupts and faults. I/O interrupts cannot be generated by a processor in slave mode. Certain types of faults can be generated deliberately by a slave job; other types are due to outside causes. The action taken by the Executive will depend on the type of interrupt or fault, and possibly on the state of the job's registers and main memory when the fault occurs.

1.2.2 Transparent Interrupts

Certain types of interrupts are normally transparent to a slave job. That is, the job is restarted after the interrupt with all its memory and working registers (including the instruction counter or IC) preserved, so that it need not even be aware that the interrupt occurred. Included in this category are all connect faults, timer runout faults, and startup and shutdown faults. These events are not completely undetectable by the slave job, though. Any time a job is interrupted, including transparent interrupts, the Executive may take advantage of its opportunity to give a pending trap, slave fault, or special interrupt (cf. Section 1.4). It may also modify the BAR or timer register as previously described.

If it is imperative to the slave job that not even these events occur, it may prevent all transparent interrupts by the use of the inhibit bit. This hardware feature prevents any of the transparent interrupts from occurring. Care should be taken, however, to insure that the inhibit bit is not used more than is absolutely necessary. Excessive use of the inhibit bit has the following disadvantages:

(1) If the slave job executes inhibited code for too long, a lockup fault is generated. This fault may occur in the middle of the execution of one machine instruction, making it impossible for the slave job to continue.

(2) Excessive use of the inhibit bit prevents the Executive from recognizing I/O interrupts as quickly as possible, thus slowing down the rate of I/O activity and reducing the total efficiency of the time-sharing system.

1.2.3 MME Faults; Executive Calls

A slave job may request certain services from the Executive by generating a MME (master mode entry) fault. The address field of the instruction causing the fault contains a code for the type of call, and the job's registers and memory contain any additional parameters necessary. The specifications for the valid calls are contained in Chapter 2, "Specifications for Executive Commands".

Executive calls are of three types: nontrapping, trapping, and trapping with automatic pause. The action requested by a nontrapping call is performed before the slave job regains control. From the point of view of the slave job, its action is instantaneous. This action may include modification of the slave job's registers or memory, as described in the specifications. Except for such modifications, a nontrapping call is essentially a transparent interrupt.

Trapping calls also function as transparent interrupts. However, after a trapping call has been issued and the slave job has regained control, the action may or may not be complete. A trapping call must specify a trap address in X6 at the time of the MME fault. When the action is complete, a trap (described in Section 1.4) occurs to that address in the slave job's memory area.

Trapping calls with the automatic pause feature share features of trapping and nontrapping calls. The calls are set up much like a trapping call; e.g. a trap address is necessary, but a pause for one interrupt is assumed at the time of the call. This feature may make the call look much like a nontrapping call. It is considered good programming technique, however, to check to make sure that the particular trap for the current MME was returned, even after the automatic pausing type of Executive calls.

If a job is in squeezed mode, or if the address field of the MME instruction does not specify a valid call, or if the trap address for a trapping call is zero, negative, or out of bounds, the MME fault is returned to the slave job as a slave fault (see below). The upper three bits of the code for a valid Executive call are always 101 to reduce the chance of an undebugged job accidentally generating and issuing a valid Executive call.

1.3 Slave Fault Vector

The first 40 (octal) words of a slave job's memory form its slave fault vector. This area is divided into two-word fault cells, one for each of the sixteen types of faults described in the "Honeywell 600/000 Programming Reference Manual". Some of these cells are used for special purposes by the Executive, while others are used for slave faults. The position of the individual cells within the slave fault vector corresponds to the hardware-defined locations in the master mode fault vector. They are:

Location	Use
0	initial entry point/special interrupt tally
2	memory fault
4	MME fault
6	fault tag fault
10	CRU timer runout fault
12	command fault
14	DRL fault
16	lockup fault
20	special interrupt
22	unused (reserved)
24	illegal procedure fault
26	incomplete op fault
30	unused
32	overflow/truncation fault
34	divide check
36	real time timer runout fault

1.3.1 Slave Fault Procedure

When certain faults occur, they are returned to the slave job which generated them as slave faults. To give a slave fault, the Executive stores the job's IC (instruction counter) and IR (indicator register) registers at the time of the fault in the first word of the appropriate fault cell in the slave fault vector. The job's IC is reset to point to the second word of the fault cell.

If the first word of the fault cell was zero before the IC/IR were stored, then control is transferred to the second word of the fault cell. However, if it was nonzero, the job had a "dirty fault vector" and is aborted; that is, it is suspended, and the job which ran it is notified by a trap that the job had a dirty fault vector. The supra job then has the option of continuing or terminating the aborted slave job. This feature is intended to prevent an undebugged job from looping indefinitely due to a fault in its fault processing routine.

1.3.2 Slave Faults

Whenever a memory, fault tag, DRL, lockup, illegal procedure (IPR), incomplete op (ONC), overflow/truncate, or divide check fault occurs, it is returned as a slave fault. Command faults are normally returned as slave faults. However, if the job in question is highly privileged (enabled) it may use a command fault to request an abort of the entire system (see ENABLE MME). This feature is to allow certain high-level slave jobs to request a system abort in case they detect a serious error. Command faults are generated when a privileged (master mode) instruction is executed in slave mode.

MME faults normally indicate Executive calls. However, if the job generating the MME is in squeezed mode (entered by the Executive call SQUEEZE), if the address field of the instruction causing the fault does not specify a valid Executive call, or if the trap address specified for a trapping call is zero, negative, or out of bounds, the MME fault will be returned as a slave fault.

1.3.3 Simulated Slave Faults

Two software-simulated timers are available to a slave job. A run-time timer is set by the Executive call SET TIMER, and a real-time (elapsed time) timer is set by the Executive call CLOCK. Both these calls are nontrapping.

As soon as possible after either of these timers has counted down to zero, the timer is disabled and a slave fault is given. A timer runout fault is returned if the run-time timer counts to zero, and an execute fault is given if the real-time timer runs out. Either of these simulated faults may cause the job to be aborted for a dirty fault vector. The simulated faults are the only slave faults given in these fault cells; neither the hardware timer runout nor execute faults are returned as slave faults.

1.3.4 Special Locations

The shutdown and connect fault cells have special uses in the slave fault vector. Word 0 (the first word of the shutdown fault cell) is the initial entry point to the job. When the job is first run, its IC is set to zero. Word 1 (the second word of the shutdown fault cell) is the job's special interrupt tally word. The connect fault cell is used as the special interrupt cell. Special interrupts are described in detail in Section 1.4.

The parity fault cell is not currently used, and is reserved for future use. The startup fault cell is not used by the Executive, and is available to the slave job, if it wishes to use it.

1.3.5 Parity Faults; Execute Faults

When a job generates a parity fault, the job is terminated. The supra job is notified in the trap that the job was terminated due to an error detected by the Executive (recoverable error status 400). In case of an execute fault, the system is aborted. An execute fault can be generated only by the operator at the processor control panel, and it is never a slave fault.

1.4 Executive to Job

In addition to slave faults, the Executive also communicates with the slave job by using traps and special interrupts.

1.4.1 Traps

A trap occurs whenever the action requested by a trapping executive call is completed, or when it is terminated prematurely by an error or invalid parameter. A trap address is specified when the call is issued. This address is the start of a block of at least four words in the slave job's memory. When the trap is ready to "spring", the Executive, at its first opportunity, stores two words of status information in the first two words of this block. The job's current IC and IR are stored in the third word of the trap block (the return word), and its IC is reset to point to the fourth word. (This procedure should be compared with the similar procedure used to give a slave fault.) The fourth word should contain the beginning of the slave job's trap processing routine. Bits 9-17 of the first status word always contain a status code indicating whether the operation was completed successfully, and if not, why not. Other fields give more detailed information. A precise description of the two status words is given in Section 2.3.

The following sequence of events occurs when a slave job is trapped upon completion of a trapping command:

(1) Execution of the slave job is suspended;

(2) Two words of status information are stored in the first two words of the trap block pointed to by the slave job's X6-register when the command was issued;

(3) The current value of the slave job's instruction counter (IC) and indicator register (IR) are stored in the third word of the trap block; and

(4) The slave job is restarted at the fourth word of the trap block.

A trap block may contain any sequence of instructions starting in the fourth word. This trap routine should end with an RET instruction whose address field points to the third word of the block in order to resume execution of the code interrupted by the trap.

It is possible for the time between the issuance of a command and the resulting trap to vary considerably. Some traps may occur immediately and more than one trap may occur at once. A program which exits from each trap routine by a RET instruction will encounter all traps in the order in which they occurred; it will also encounter all traps before encountering any special interrupt which occurred at the same time as the traps. Note, however, that commands are not necessarily trapped in the same order in which they were issued.

1.4.2 Special Interrupts

A special interrupt is used by the Executive to notify a job of some event other than the completion of a trapping Executive call. A job will receive special interrupts only if its special interrupt tally word (word one) is not zero. If the word is zero, the job is not accepting specials. In this case, any special interrupts which would be given to the job are either discarded or bounced to a job higher in the job tree, depending on the origin of the special interrupt (see Section 1.4.6).

1.4.3 Special Interrupt Stack

If a job is accepting specials, the special interrupt tally word (word 1) contains a tally pointer to the job's special interrupt stack. The tally word is typically generated by using the GMAP pseudo-operation TALLYD, as in:

```
SPTAL:TALLYD STACK,NUMBER,2   SPECIAL INTERRUPT TALLY
```

Here STACK is the symbolic location of the area in the slave job set aside as the special interrupt stack; NUMBER is the number of 2-word, special interrupt data entries (special interrupt pairs) which will fit in the stack, and the "2" refers to the length of a special interrupt pair. The last field need not be set, but most programmers find it convenient to do so.

1.4.4 Special Interrupt Procedure

At its first opportunity after a special interrupt is generated (caused), the Executive will do the following:

(1) If the job is not accepting specials (the special interrupt tally word is zero), the special is discarded or "bounced", and the procedure is complete. A special interrupt on a slave end of a communications file can be "bounced" to a higher slave end which is accepting special interrupts.

(2) Otherwise, any pending roadblocks on the job are released (see the description of the PAUSE call in Chapter 3).

(3) If the tally field (bits 18-29) of the special interrupt tally word is zero, indicating that the special interrupt stack is full, the special interrupt is "saved" (see below), and the procedure terminates.

(4) Otherwise, if the address field (bits 0-17) of the special interrupt tally word points out of bounds (outside the job's memory), the special is discarded, and the procedure ends.

(5) Two words of special interrupt data (special interrupt pair) are stored in the special interrupt stack. The address field of the tally word is incremented by two, and the tally field is decremented by one.

(6) If word 20 (the first word of the slave connect fault cell) is zero, a slave connect fault is simulated. If word 20 is nonzero, no fault takes place. No dirty fault vector aborts are generated. This feature can be used to prevent a slave job from receiving a special interrupt (connect fault) while it is executing its special interrupt service routine, if the slave job does not clear word 20 until the end of its special handling routine.

1.4.5 Saved Special Interrupts

If the job's special interrupt stack is full (the tally field of the tally word is zero) in step 3 above, the special interrupt pair will be saved by the Executive and given to the job as soon as the tally field becomes nonzero. The order of the special interrupts is preserved when each pair is saved. However, once a special interrupt pair has been saved, the special interrupt cannot be bounced. If the job stops accepting special interrupts (clears the special interrupt tally word to zero) before making room in the special interrupt stack, all saved special pairs will be discarded.

1.4.6 Bouncing Special Interrupts

Only special interrupts which are generated by a DRIVE command at the master end of a communication file are bounced. The method by which the Executive determines which job will receive the special interrupt is covered in detail in Section 5.2.

1.4.7 Special Interrupt Pair Format

The two data words which are placed by the Executive in the special interrupt stack have the following format:

0 — 13	14 — 17	18 — 35
RSVD	TYPE	FRN
DATA		

RSVD - Reserved for future use. These bits may contain garbage.

TYPE - Special interrupt type (see below for a list of types).

FRN - Depending on the type of the special interrupt, this field may contain a file reference number. On specials generated by a DRIVE on a communication file, this field contains the FRN of the end of the communication file held by the job receiving the special. On passed file and returned file specials (types PF and RF), the FRN of the file passed or returned is given here. On file closed specials (type FCLO), the FRN of the former communication file end is given here.

DATA - The use of this field depends on the type of the special interrupt. If the special was generated by a DRIVE or CLOSE on a communication file, the lower half of the DATA field always contains the drive function data (from the lower half of the issuing job's A-register). Other uses of the DATA field are listed below.

1.4.8 Types of Special Interrupts

Type	Mnemonic	Use
0	STM	Set mode - used to notify the master end job that a slave end job issued a set mode drive (see DRIVE-slave end in Section 5.2). The lower half of DATA contains the mode.
1	RCF	Read communication file - used to notify a slave end job that the master end job wishes the slave to issue a READ. By convention the lower half of DATA contains the length to read. (See DRIVE-master end, Section 5.2.)
2	SRS	Slave issued reset - notifies the master end job that a slave end job issued a RESET STATUS on a COPY to which the master end job had not yet responded. (See RESET STATUS, Section 5.2.)
3	BRK	Break - notifies a slave end job that the master end job has issued a break drive. (See DRIVE-master end, Section 5.2.)

4	PF	Passed File - notifies the job that a file has been passed to it via the PASS command. This special interrupt is not caused by the implicit PASS command in the RUN command. If DATA is nonzero, the passed file special is due to a "passback" from an immediate infra-job, and the lower half of DATA contains the FRN of the job file from which the file was passed.
5	RF	Returned file - notifies the job that a file which it passed with the return bit set has now been returned to it. The lower half of DATA contains the FRN of the job file to which the file being returned was passed.
6	FCLO	File closed - notifies the job that a communication file of which it held an end has been destroyed because a CLOSE was issued at another end. DATA contains the contents of the A-register of the job requesting the close at the time of the close MME.
7		Reserved for future use.
10	ITA	Illegal trap address - notifies a job that the Executive attempted to give it a trap, but discovered that the trap address was outside the job's memory. The lower half of DATA contains the illegal trap address. This condition can arise if memory containing the trap block is released with a MEMORY REQUEST command before the trap occurs.
11		Reserved for future use.
12	D	Slave issued drive - notifies the master end job that a slave end job issued a device drive. The lower half of DATA contains the device command (MODE). (See DRIVE-slave end.)

13	R	Slave issued read - notifies the master end job that a read-type command was issued at a slave end. DATA contains the requested length. (See slave end COPY commands, Section 5.1.)
14	W	Slave issued write - notifies the master end job that a write-type command was issued at a slave end. DATA contains the requested length. (See Slave end COPY commands, Section 5.1.)
15	RS	Slave issued request status - notifies the master end job that a REQUEST STATUS was issued at a slave end. DATA contains the requested length of the status block.
16	T	Slave issued truncate - notifies the master end job that a TRUNCATE or SCRATCH command was issued at a slave end. DATA contains the requested truncation length (zero for SCRATCH). (See TRUNCATE-slave end only, Section 5.2.)
17	SP	Slave issued set pointer - notifies the master end job of a SET POINTER command issued at a slave end. DATA contains the requested pointer setting. This special is automatically given to a master end job when a COPY to or from a slave end specifies a starting location pointer (see COPY MME).

1.4.9 Simultaneous Traps and Special Interrupts

It is possible for more than one trap, special interrupt, or slave fault to be given to a job each time that job is interrupted. In this case, the traps are linked, with the IC in the return word of one trap pointing to the next trap. It is thus important that a job be programmed so as always to return from its traps, special interrupts, and slave faults by issuing a RET instruction to the appropriate return word. This ensures that if more than one trap, or if a trap and a special interrupt occur simultaneously, both will be serviced by the slave job. Even if the programmer believes the situation cannot occur, it is still a good practice to return from all traps through the return word. It typically simplifies program design and increases readability to do so.

1.4.10 Typical Trap-Handling Code (Non-Multiprogrammed)

The following sequence of instructions is a typical way for a nonmultiprogrammed slave job to handle a trapping Executive call. It illustrates the use of the return word.

```
         EAX6    TRAP       point to trap block
         STZ     TRAP+2     clear return word as flag
         MME     (type)     issue trapping Executive call
         INHIB   SAVE,ON    inhibit pause routine
CHECK:   SZN     TRAP+2     check return word
         TNZ     SPRUNG     if not zero, trap has sprung
         LDX5    1,DU       load pause count
         MME     PAUSE      wait for one trap or interrupt
         INHIB   RESTORE    reset inhibit bit to normal
         TRA     CHECK      check flag again
SPRUNG:  LDX0    TRAP       pick up status word 1
         ANX0    =0777,DU   mask to status field
         CMPX0   GOOD,DU    check for expected status
         TNZ     ERROR      go analyze error
         .
         .                  remainder of program
         .
TRAP:    BSS     2          two status words
         BSS     1          return word
         RET     *-1        return immediately from trap
```

This routine always waits after issuing an Executive call. Note that the trap routine returns immediately through the return word, and the main routine checks the status words. The return word is also used as a flag, since the Executive will not put the IC/IR into this word until it springs the trap. If Executive calls are issued in many places in a program, it is usually profitable in terms of memory size to place portions of the above routine, such as the pause loop, in a common subroutine.

1.4.11 Ordering of Simultaneous Traps

The Executive will link simultaneous traps and specials in such a way that a job which returns through the appropriate return words (as all jobs normally should) will encounter first all traps and simulated slave faults in the order in which they actually occurred, then the connect fault (special interrupt) if any, and then any slave fault other than simulated slave faults. The job will thus process all traps before specials. The return word in a slave fault cell will always point to the location where the fault actually occurred.

DTSS

1.4.12 Traps from Squeezed Mode

When a job running in squeezed mode (see SQUEEZE, MME 500007 in Chapter 3) receives a trap, special interrupt, or slave fault, it is unsqueezed. The trap or interrupt takes place normally, except that the saved IC/IR is relative to the squeezed BAR, and has bit 35 (the squeeze bit) set on to indicate this fact. Note that the above sample trap-handling routine cannot be used by a job which expects to be trapped from squeezed mode, since the RET instruction will not "resqueeze" the job.

Chapter 2

Specifications for Executive Commands

The specifications for Executive commands are divided into three sections:

- o brief summary of the parameters for and the status returns from each command;
- o detailed specifications for the nontrapping commands;
- o detailed specifications for the trapping commands.

As described in Section 1.2, a slave job issues an Executive command by executing a MME (master mode entry) instruction with an address field specifying a legitimate command; execution of a MME instruction whose address field does not specify a legitimate command will result in a MME fault. (A job receives a MME fault by having the current value of its IC/IR stored in location 4 of its job memory and then having its IC reset to location 5.) There is one exception to this rule: if the address field of the MME instruction is exactly 100 (octal) greater than a valid <u>trapping</u> MME number, this is equivalent to issuing that MME followed immediately by a PAUSE MME with a pause count of one. Such MME's are called 200 series, or pausing MME's. The contents of the slave job's registers upon execution of a MME instruction specify parameters relevant to the command.

Except in the case of a SQUEEZE or a TERMINATE command, a slave job will regain control at the location following a MME instruction unless a fault, trap, or special interrupt was generated as the Executive was processing the command. In the case of nontrapping commands, the Executive will have completed processing the command by the time the slave job regains control.

In the case of trapping commands, the Executive may not have completed processing the command; instead, execution of the command and of the slave job may continue in parallel, with the slave job being trapped upon completion of the command.

Numbers occurring in the specifications of this section can be of two types: octal and decimal. In general, numbers which can occur as the contents of a register are in octal; thus, addresses, status returns, or special interrupt numbers are in octal. Numbers which do not occur as the contents of a register are in decimal; for example, bit numbers and fractions of milliseconds are given in decimal.

2.1 Nontrapping Executive Commands

Command	MME	X0	X1	X2	X3	X4	X5	X6	X7	A	Q	Return
Terminate	500000					Access						
Set Timer	500001									Time		A = old
Running Time	500002											bin. A
Time of Day	500003											ASCII AQ
Date	500004											ASCII AQ
Pause	500005						N					
Memory Request	500006						Bound					X5=0 if ok
Squeeze	500007					IC/IR*	BAR		Regs*			
Time Since Bootload	500010											bin. A
Pure	500011	FR							N*			X7=0 if ok
Job Time	500012											bin. A
Long Pause	500013								N			
Clock	500014									Time		A = old
Unused	500015											
Unused	500016											
Enable	500017									Type	Mask	A = result

2.2 Trapping Executive Commands

Command	MME	X0	X1	X2	X3	X4	X5	X6	X7	A/Q
Open Scratch	500100					Access	Pref	Trap*		Cat MAX
Open	500101	FR Cat	Name*		Pass*	Access		Trap*	N	Slave trap I.D.
Old Erase	500102	FR Cat	Name*		Pass*			Trap*		Slave trap I.D.
Catalog	500103	FR Cat	Name*	FR	Pass*	Access*	Date*	Trap*		
Uncatalog	500104	FR						Trap*		
Close	500105	FR						Trap*		Comfile special
Overlay	500106	FR				Flag bits		Trap*	Reg*	
Truncate	500107	FR						Trap*		Length
Scratch	500110	FR						Trap*		
Change Catalog Entry	500111	FR	Name*		Pass*	Access*	Date*	Trap*		
Erase	500112	FR Cat	Name*	Zero	Pass*	Trap mask	Zero	Trap*	N	Slave trap I.D.
Set Pointer	500113	FR						Trap*		Pointer
Read Catalog	500114	FR Cat	M1*		M2*			Trap*	N*	
Request Status	500115	FR			M2*			Trap*	N*	
Replace	500116	FR Cat	Name*	FR	Pass*	Trap mask	Zero	Trap*	N	Slave trap I.D.
Execute	500117	FR	Ser		Type	Limits	List*	Trap*	Reg*	Time/Acc
Run	500120	FR	Ser		Type	Limits	List*	Trap*	Reg*	Time/Acc
Continue	500121	FR Job				Limits		Trap*		Time
Pass	500122	FR Job	Msg*	FR		Access	Pref	Trap*	N*	
Alter Accesses	500123	FR				Access		Trap*		
Change Catalog MAX	500124	FR Cat						Trap*		Cat MAX
Old Read Cat & Open	500125	FR Cat	M1*		M2*	Access	Cod Date	Trap*	N*	Slave trap I.D.
Provide DAs	500126	FR			M2*			Trap*	N*	
Read Catalog & Open	500127	FR	M1*	Zero	M2*	Access	Date*	Trap*	N*	Slave trap I.D.
Old Replace	500130	FR Cat	Name*	FR	Pass*			Trap*		Slave trap I.D.
Copy	500131	FR1	M1*	FR2	M2*	Flag bits		Trap*	N*	
Drive	500132	FR				Flag bits		Trap*		Function
Read	500133	FR			M2*	Flag bits		Trap*	N*	
Write	500134		M1*	FR		Flag bits		Trap*	N*	
Reset Status	500135	FR				Flag bits		Trap*		
Tally Open	500136	FR Cat	Tally*	Zero	FR Alt Cat	Access	Last Name*	Trap*	Zero	Slave trap I.D.
Tally Erase	500137	FR Cat	Tally*	Zero	FR Alt Cat	Trap mask	Last Name*	Trap*	Zero	Slave trap I.D.
Tally Replace	500140	FR Cat	Tally*	FR	FR Alt Cat	Trap mask	Last Name*	Trap*	Zero	Slave trap I.D.
Tally Catalog	500141	FR Cat	Tally*	FR	FR Alt Cat	Access*	Last Name*	Trap*	Date*	
Log	500142	Zero	M1*	Zero	Zero	Flag bits	Zero	Trap*	N*	
Duplicate	500143	FR	Zero	Zero	Zero	Access	Zero	Trap*	Zero	

* indicates pointer to word containing parameter

2.3 Trap Block Format

	0	8 9	17 18	35
Status word 1	(Access)	Status	(File Reference Number)	
Status word 2	(Length or Counts)			
Return word	IC		IR	

Trap Routine

Access (bits 0-8 of status word 1)

(1) On access error (status 100), gives the access bit needed for successful completion of the command.

(2) On OPEN, ALTER ACCESSES, DUPLICATE, and REPLACE, gives the access with the file is open or could have been opened.

(3) On RUN, EXECUTE and CONTINUE, gives termination access of infra job.

(4) On DRIVE with DCW's, gives record count residue.

(5) Zero in all other cases.

Status (bits 9-17 of status word 1)

Major status (see individual trapping Executive command descriptions).

2.4 Summary of Status Returns

ALTER ACCESSES

0 Successful
1 Partial success
2 File closed

SW2= Length of file

CATALOG

0 Successful
1 Illegal trap protection
2 Duplicate entry in catalog
3 File not a scratch file
4 File preference too low
13 Illegal usage or dates

TALLY CATALOG

0 Successful
1 Illegal trap protection
2 Duplicate entry in catalog
3 File not found
4 Protection violation
6 Bad tree name
7 Fetch error
12 Format error
13 Illegal usage and dates
14 File preference too low

SW2= Substatus/character count and position (if status is 12)

CHANGE CATALOG ENTRY

0 Successful
1 File not cataloged
2 Duplicate file name found in catalog
3 Illegal trap bits
13 Illegal usage and dates

CHANGE CATALOG MAX

0 Successful
1 Rejected: specified MAX is less than current ALOC

CLOSE

0 Successful
6 Communication file busy

COPY/READ/WRITE/READ CATALOG/REQUEST STATUS/READ CATALOG AND OPEN

0 All words transferred
1 Source file exhausted
2 Destination file exhausted
3 Command inappropriate for state of communication file
4 Source file pointer out of bounds
5 Destination file pointer out of bounds
6 Communication file busy
7 Master end of communication file not accepting special interrupts
14 State vector full (READ CATALOG AND OPEN only)

SW2= Number of words transferred - number requested

DRIVE

0 Successful
1 Not all data transferred
2 Destination file exhausted
3 Command inappropriate for state of communication file
6 Communication file busy
7 Other end of communication file not accepting special interrupts
10 Bad DCW (Drive Type 24 only)

SW2= Device status on device file drives

LOG

0	Successful
1	Rejected: log already outstanding
2	No log buffer available
3	Escape sequence error

OPEN/ERASE/REPLACE/TALLY OPEN/TALLY ERASE/TALLY REPLACE

0	Successful
1	Partial success
2	Lockout
3	File not found
4	Protection violation
5	Fail
6	Bad tree name
7	Fetch error
10	Off-line file
11	Rejected: specially cataloged files (device files) cannot be removed from catalogs (ERASE and REPLACE only)
12	Format error (TALLY MME's)

SW2= Length of file or catalog MAX (zero on ERASE) or substatus/character count and position (if status is 12 on TALLY MMEs)

OVERLAY

1	Rejected: operation outstanding
2	Rejected: file too long

PASS

0	Successful: message, if any, was read
2	Message not read

SW2= Number of words read - number of words in message

RESET STATUS

0 Successful
1 Reset status already in progress
6 Communication file busy
7 Master end of communication file not accepting special interrupts

RUN/EXECUTE/CONTINUE

0 Job terminated
1 CRU limit exhausted
2 Job aborted
5 Error in run list
6 Swap error; job terminated

SW2= CRUs consumed by job

TRUNCATE/SET POINTER/SCRATCH

0 Successful
1 Length too long or pointer out of bounds
6 Communication file busy
7 Master end of communication file not accepting special interrupts

SW2= Length of file - length requested (if status is 1 on TRUNCATE)

UNCATALOG

0 Successful
1 File not cataloged

Chapter 3

Nontrapping Executive Commands

This chapter contains detailed specifications for all nontrapping commands (in alphabetical order; for a list sorted by number, see Section 2.1).

Clock	MME 500014
Date	MME 500004
Enable	MME 500017
Job Time	MME 500012
Long Pause	MME 500013
Memory Request	MME 500006
Pause	MME 500005
Pure	MME 500011
Running Time	MME 500002
Set Timer	MME 500001
Squeeze	MME 500007
Terminate	MME 500000
Time of Day	MME 500003
Time Since Bootload	MME 500010

MME 500014: CLOCK

A Time

The CLOCK command enables a job to receive an execute fault (real-time timer runout fault) after a specified amount of real time has elapsed. If no CLOCK command is issued, or if a CLOCK command with an argument of zero is issued, then no real time timer runout fault will occur. A CLOCK command with a nonzero argument sets a real-time timer for the issuing job; this timer decrements continually, and a real time timer runout fault occurs when it reaches zero (if the issuing job happens to be road-blocked at this time it will be unroadblocked to receive the fault). After a real time timer runout fault has occurred, the real-time timer is disabled, and no further such fault will occur until a new CLOCK command is issued.

Time is a 36-bit integer which is decremented 64 times per millisecond, thus accurate to the nearest 15.625 microseconds.

Since it is possible for a job to be swapped out of memory when its real-time timer reaches zero, the Executive cannot guarantee that a job will be restarted as soon as its real-time timer runs out. It can guarantee only that the job will receive an execute fault the first time it regains control after its real-time timer runs out.

A job receives an execute fault by having its current IC and IR stored in location 36 of its job memory and then having its IC reset to 37.

When a CLOCK command is issued, the old value of the job's real-time timer is returned to the A-register.

Note: A CLOCK command with a zero argument will not prevent an execute fault from occurring if a previously set CLOCK has already run down.

MME 500004: DATE

No arguments

DATE loads the ASCII date in the standard format "month/day/year" into the AQ-registers, e.g. 03/13/73.

MME 500017: ENABLE

A Desired setting for job type bits masked by Q (bits 0-8)

Q Mask for job type bits to set or reset (bits 0-8)

The job type of a running job consists of two 9-bit quantities: the permissions allowed for the job and the permissions for which the job is currently enabled. These permissions are as follows:

Bit	Function
0	Monitor permission (affects scheduling only)
1	Core Residence permission
2	Large State Vector permission (permits a state vector with more than 64 files)
3	Special Catalog permission (allows a job to specify infinite catalog MAXes, and to estimate the number of entries when creating a catalog)
4	Load-Dump permission (allows preallocation of scratch files)
5	Priority Scheduling (PDQ)
6	Crash (allows slave job to abort system)
7	Log (allows job to do MME LOG)

In addition the load-dump permission suppresses quota checks involving catalog MAXes, enables the PROVIDE DEVICE ADDRESSES command, and provides device addresses in the READ CATALOG AND OPEN FILES command.

When a job is run, bits 0-8 of index register X3 on the RUN command are ANDed with the job type bits allowed to the issuing job to form the permissions allowed to the infra job. The only permissions which are enabled on the run are Large State Vector permission, Priority Scheduling, Crash, and Log; all other permissions must be enabled by the ENABLE command.

When an ENABLE command is issued, only those permissions corresponding to bits for which there are ones in the Q-register are affected. Each such permission is enabled if it is allowed for the job. When the job regains control, the corresponding bit in the A-register contains the current state of all permission bits that were affected.

MME 500012: JOB TIME

No arguments

The total running time (CRU's) for the issuing job is loaded right-justified in the A-register; running times for jobs running below the issuing job are not included in the total. The running time is in timer units of 1/64 milliCRU. (See also RUNNING TIME.)

MME 500013: LONG PAUSE

X5 Number of traps

LONG PAUSE is identical in effect to PAUSE. LONG PAUSE should be used in place of PAUSE when the issuing job does not expect to be restarted while residing in main memory. For example, it should be used when awaiting a wake-up special interrupt or when pausing after issuing a READ on a terminal communication file.

MME 500006: MEMORY REQUEST

X5 Upper memory limit

MEMORY REQUEST causes the BAR for the issuing job to be extended or compressed so as to allow a maximum valid address of N-1, where N is the least nonzero multiple of 1K (1024 words) greater than or equal to the specified upper memory limit. The new BAR will be in effect when the job regains control. Memory obtained will be zeroed out; memory released is lost.

If the command is successful, then X5 will be zero on return. If the requested memory is not available, then the amount that is available will be returned in X5 and another MEMORY REQUEST can be issued to obtain this amount. If a job has previously issued a PURE command, then any MEMORY REQUEST which would release memory containing pure procedure will be rejected, and the least amount of memory for which a memory request will succeed will be returned in X5.

If the job's IC is no longer within the range of the new BAR after memory is released, then a memory fault will occur.

The amount of memory that can be requested by a job is limited by the current system limit on maximum memory length and by the memory limit imposed by the supra job in X4 of the RUN or CONTINUE command which initiated the job.

MME 500005: PAUSE

X5 Number of traps

PAUSE causes the issuing job to be roadblocked until N traps, one special interrupt, or one real-time timer runout fault has occurred. If an N greater than the number of outstanding trapping executive commands is specified, then an N equal to the number of outstanding trapping executive commands will be assumed. If N is zero, the job will be roadblocked and will restart as soon as there is an interrupt, trap, or fault. The job will be restarted in a fault cell or trap block with the location after the PAUSE command in the return address.

MME 500011: PURE

X0 File reference number of source file for pure procedure

X7 Points to length N

A segment of job memory is said to contain "pure procedure" if no location in that segment is ever altered during the execution of the job. The PURE command indicates that the issuing job consists of pure procedure from word 100 to word N-1 if X7 is nonzero, or that the issuing job contains no pure procedure if X7 is zero. This PURE facility is designed to avoid the necessity of swapping and restoring common packages of code.

Any error in the parameters for the PURE command will cause the command to be rejected and X7 to be nonzero on return. The length N must be a multiple of 100 and must be less than or equal to both the current length of the issuing job and the length of the source file specified in X0. The file specified in X0 must be open with Read permission.

If a PURE command with a nonzero value in X7 is successful, then a zero will be returned in X7. On subsequent swaps of the job, words 100 to N-1 will not be swapped, but will be restored from the source file specified in X0. This file will remain busy as long as the PURE command is in effect.

If a job does not issue a PURE command, it will be assumed to contain no pure procedure. A job which is running with pure procedure can at any point declare itself impure by issuing a PURE command with X7 zero or by issuing a RESET STATUS on the source file for the pure procedure. Either of these two actions will cause subsequent swaps to occur normally and will allow access to the source file for pure procedure.

Copies to and from the pure region of a job file will be aborted if they occur when that job file is swapped out of main memory.

MME 500002: RUNNING TIME

No arguments

The total running time (CRU's) for the issuing job and all jobs running below it is loaded right-justified in the A-register. The running time is in timer units of 1/64 milliCRU. (See also JOB TIME.)

MME 500001: SET TIMER

A Time

The SET TIMER command enables a job to receive a timer runout fault after a specified amount of processing. This timer decrements only when the job is actively running or doing I/O, and a timer runout fault occurs when this timer reaches zero or becomes negative. If no SET TIMER command is issued, or if a SET TIMER command with an argument of zero is issued, then no timer runout fault will occur. A SET TIMER command with a nonzero argument sets a timer for the issuing job. After a timer runout fault has occurred, the timer is disabled, and no further timer runout fault will occur until a new SET TIMER command is issued.

When a SET TIMER command is issued, the old value of the job's timer is returned in the A-register.

When a SET TIMER command is issued, the old value of the job's timer is returned in the A-register.

A job receives a timer runout fault by having its current IC and IR stored in location 10 of its job memory and then having its IC reset to 11.

MME 500007: SQUEEZE

X4 Points to IC/IR

X5 BAR

X7 Points to registers

The SQUEEZE command enables a slave job to simulate a master mode program by setting a pseudo base address register. SQUEEZE causes the BAR for the issuing job to be changed to resemble the BAR specified in X5, where the format of the BAR in X5 is as it would be if the slave job were being executed in master mode. For example, to squeeze a job so that address 2000 becomes 0 and the upper memory limit is 6000 (i.e. 6000 above the base of 2000), the BAR in X5 would be 002006.

After the BAR has been changed, the registers are loaded by an LREG instruction from the location pointed to by X7, and a RET instruction is executed through the specified IC/IR. If a fault, trap, or special interrupt occurs while the job is in the squeezed mode, the BAR is unsqueezed and the fault, trap, or special interrupt takes place normally with bit 35 of the saved IC/IR in the trap or fault cell set to 1. Thus a squeezed job can never call the Executive since all MME faults will be returned to the unsqueezed job. All memory outside of the squeezed BAR is retained and becomes available again after the BAR is unsqueezed.

If any of the parameters in a SQUEEZE command are out of range, then the command will be ignored and the next instruction in sequence will be executed.

MME 500000: TERMINATE

X4 Termination access

TERMINATE causes the termination of the issuing job and all jobs running below it. All files open for the job are closed and, if the return bit is set in their access, returned to the supra job. If the supra job is accepting specials, then a special interrupt type 5 (returned file) is generated for each returned file. Upon termination, a job's remaining scratch and catalog word allotments are added to the corresponding allotments for the supra job.

On a normal program, the termination access specified in X4 is returned in bits 0-8 of status word 1 in the trap for the RUN or CONTINUE command which initiated the job. On a trap program protecting a cataloged file, the access in X4 will be the access permitted for the opened file.

MME 500003: TIME OF DAY

No arguments

The ASCII time of day in the standard format "hours:minutes:seconds" is loaded into the AQ-registers, e.g. 20:15:45.

MME 500010: TIME SINCE BOOTLOAD

No arguments

The time since system bootload is loaded right-justified into the A-register in timer units of 1/64 millisecond. This time is not affected by the operator's entering the time of day.

Chapter 4

Trapping Executive Commands

This chapter contains detailed specifications for all trapping commands (in alphabetical order; for a list sorted by number, see Section 2.2).

MME 500123: ALTER ACCESSES

X0 File reference number of cataloged file/catalog

X4 Access to retain on file/catalog (bits 0-8)

X6 Trap location

Any accesses present on the specified file/catalog which are not specified in X4 are relinquished. Any accesses specified in X4 and not present on the file/catalog will be granted if they would have been allowed at the time the file/catalog was opened, and if they do not conflict with accesses currently held by other jobs having the given file/catalog open. If all accesses present on the file/catalog are relinquished, then the file/catalog will be closed; i.e. its file reference number will become invalid (cf. CLOSE).

Upon completion of the command, a trap occurs to the location specified in X6. Bits 0-8 of status word 1 contain the access bits remaining on the file/catalog.

Status word 2 contains the current length/MAX of the file/catalog. Note that this may not be the same as the length given when the file/catalog was originally opened if another job has modified the file/catalog.

Status Returns

0 Successful: the accesses in bits 0-8 remain and are the same as those requested.

1 Partial success: the accesses in bits 0-8 remain but differ from those requested. This status return can occur if X4 contains an access bit not previously present or if bit 0 of X4 is 0 (is 1) and the file/catalog is a catalog (is a file).

2 File closed: all accesses present were relinquished.

120 Busy: another operation is outstanding on the file/catalog.

200 X0 parameter error: X0 does not contain the file reference number of a cataloged file/catalog.

MME 500103: CATALOG

X0	File reference number of catalog
X1	Points to name of scratch file/catalog to be cataloged
X2	File reference number of file/catalog to be cataloged
X3	Points to password or to name of trap program
X4	Points to access word
X5	Points to usage information or is zero (load-dump enabled jobs only)
X6	Trap location

The scratch file/catalog specified in X2 is entered in the catalog indicated in X0 with the specified name and protected by the specified password or trap program. If bit 18 of the access word is 0, then X3 points to a password; otherwise X3 points to the name of a trap program. The catalog specified in X0 must be open with Append permission. The scratch file/catalog must not have the return bit set in its access word.

If the job issuing the CATALOG MME is enabled with Load-Dump permission, and if X5 is non-zero, then X5 points to two words of usage information to be placed in the catalog entry. The first of these two words contains an integer in the upper half, which is used to set the file's days-used counter (see Chapter 9). The lower half of the first word contains type information or is zero. Type information has the following format:

Bits	Meaning
18-20	Must be zero
21-28	Variable information, not checked by Executive
29-31	Must be zero
32-35	File type

There is currently only one allowable file type, which is "off-line" (14 octal). Files to be cataloged with this type must have no physical storage associated with them (zero length; no device addresses).

The upper half of the second word contains a coded date used to initialize the file's Date Last Used attribute. The lower half of the same word contains a coded date used to set the file's Date Last Modified attribute. If the job is not Load-Dump enabled, or if X5 is zero, then the days-used count is set to zero, and the date-last-used and date-last-modified are set to the current date. A coded date contains in bits 0-8 the binary representation of the year (modulo 100), in bits 9-12 the binary representation of the month (JAN=1, DEC=12), and in bits 13-17 the binary representation of the day of the month.

The Executive sets bit 0 of the access word to zero if the file/catalog being cataloged is a file and to one if it is a catalog. (For further details on the catalog access word consult the publication "File Access and Protection".) The catalog structure is described in more detail in Chapter 9.

A job which successfully catalogs a scratch file/catalog has its scratch word allotment incremented by the length of the file or of the catalog header. At the same time its catalog word allotment is decremented by this quantity and also by the length of the entry created in the catalog.

Upon completion of the command a trap occurs to the location specified in X6.

Status Returns

0 Successful.

1 Illegal trap protection request: a master trap protection bit which is not allowed to the issuing job has been specified.

2 An entry with the given name is already in the catalog.

3 The file/catalog is already cataloged; i.e. it is not a scratch file/catalog.

4 The file/catalog specified in X2 has a preference of 0, 1, 2, or 4, and may be cataloged only in a core catalog.

13 Illegal usage and dates: illegal type or extraneous bits specified in usage word pointed to by X5.

40 Quotas exceeded: either

(1) Cataloging the file/catalog would cause the destination catalog's allocated storage to exceed the permissible MAX; or

(2) The length of the file or of the catalog header to be cataloged exceeds the job's remaining catalog word allotment.

60 Out of storage, for one of three reasons:

(1) The catalog is full:

(a) It already has 4095 entries; or

(b) It is not possible to append to the catalog since it either is a main-memory catalog or is already allocated in twelve fragments; or

(2) The Executive could not allocate sufficient storage for a required expansion of the catalog; or

(3) The Executive's table of opened files is full.

100 Access error: the catalog is not open with Append permission; bits 0-8 of status word 1 contain the missing permission bit.

120 Busy: another command is outstanding on either the catalog or the file/catalog to be cataloged.

200 X0 parameter error: X0 is zero or does not contain the file reference number of the catalog.

220 X1 parameter error: the pointer to the name is out of bounds.

240 X2 parameter error; either:

(1) X2 does not contain the file reference number of a scratch file/catalog,

(2) The scratch file/catalog had the return bit set in its access word (see the description of the EXECUTE command), or

(3) X5 points to usage information specifying a file type of off-line but the file has device addresses allocated (see Chapter 9).

300 X4 parameter error: either the pointer to the access word is out of bounds or the access word contains extraneous bits.

320 X5 parameter error: the job is enabled with load-dump permission, and the pointer to the coded dates is out of bounds.

400 Recoverable error: an error occurred in reading the catalog.

420 Unrecoverable error: information in the catalog has been destroyed.

MME 500111: CHANGE CATALOG ENTRY

X0 File reference number of cataloged file

X1 Points to new name or is zero

X3 Points to new password or is zero

X4 Points to new accesses or is zero

X5 Points to new usage information or is zero (load-dump enabled jobs only)

X6 Trap location

The CCE MME attempts to alter information in the catalog entry (file control block) of a cataloged file. X0 contains the file reference number of the file whose entry is to be changed. The file must be cataloged, and the job must be able to obtain Read, Write, and Append permission on the file using the ALTER ACCESS MME.

Catalog entries are described in some detail in Chapter 9.

If X0 contains a valid file reference number, the Executive attempts the following changes to the catalog entry of the file:

(1) If X1 is nonzero and points to two valid memory locations, the two words pointed to will become the new name of the file. If the first nine-bit character is not a null, all lowercase alphabetic characters will be mapped to uppercase.

(2) If X3 is nonzero and points to two valid memory locations, the two words pointed to will become the new password or slave-trap program name. If the first nine-bit character is not a null, all lowercase alphabetic characters will be mapped to uppercase.

(3) If X4 is nonzero and points to a valid memory location, the word in that location will become the access word

for the file. If this word contains trap bits which the issuing job does not possess, the MME is rejected with a status of 3. The file/catalog status of an entry cannot be changed, and any attempt to do so will be ignored.

(4) If a job is Load-Dump enabled, and if X5 is nonzero and points to two valid memory locations, those two words contain the new usage information for the file. The format of these two words is exactly the same as those pointed to by X5 of the CATALOG MME.

Upon completion of the command, a trap occurs to the location specified by X6.

Status Returns

0 Successful: the catalog entry for the file reflects all changes indicated.

1 File not cataloged: the file specified by X0 is a scratch file or has been uncataloged.

2 Duplicate file name: the catalog in which this file is located contains an entry whose name is equivalent to the one pointed to by X1.

3 Illegal trap bits: the access word pointed to by X4 specifies trap bits which the job does not possess.

13 Illegal usage and dates: illegal type or extraneous bits specified in usage word pointed to by X5. Either:

(1) The date last used was zero,

(2) The file is not a regular cataloged file/catalog,

(3) The file was passed with the return bit set (see the description of the EXECUTE command) and is being changed to an off-line file, or

(4) The variable information was specified in bits 21-28 of the first word pointed to by X5, but the file is not being changed to an off-line file.

100 Access error: job could not have altered access for RWA.

200 X0 parameter error: X0 does not contain the file reference number of a cataloged file or catalog. This status is also returned if X0 specifies a file which has a nonzero length or which has storage allocated to it and X5 points to a type word specifying a type of "off-line".

220 X1 parameter error: pointer is out of bounds.

260 X3 parameter error: pointer is out of bounds.

300 X4 parameter error: X4 is out of bounds or points to access word containing unrecognized bits.

320 X5 parameter error: pointer is out of bounds.

400 Recoverable error: an error occurred in reading the catalog.

420 Unrecoverable error: the file was being uncataloged during the MME.

MME 500124: CHANGE CATALOG MAX

X0 File reference number of catalog

X6 Trap location

A New MAX for catalog

X0 contains the file reference number of a catalog for which the job would be able to obtain Read, Write, and Append accesses via the ALTER ACCESS MME. The previous MAX for the specified catalog is replaced by the new value, which becomes the new maximum number of words which can be cataloged in that catalog. A negative MAX sets this maximum to infinity and can be specified only by those jobs enabled with the special Catalog permission. Likewise only such jobs may set the MAX for a catalog below its current ALOC.

ALOC's and MAX's are described in Chapter 9.

Upon completion of the command a trap occurs to the location specified in X6.

Status Returns

- 0 Successful.
- 1 Unsuccessful: the specified MAX is less than the current ALOC for the catalog and the job is not enabled with the Special Catalog permission.
- 40 Quotas exceeded: the catalog is cataloged elsewhere and the new MAX would cause the supra catalog's storage (ALOC) to exceed its MAX.
- 100 Access error: the maximum accesses which the job could obtain on the catalog do not include Read, Write, and Append.
- 120 Busy: another operation is outstanding on the catalog.

140 A-register parameter error: the job is not enabled with the Special Catalog permission and is specifying a negative MAX.

200 X0 parameter error: X0 does not contain the file reference number of the catalog.

400 Recoverable error: an error occurred in reading or writing the catalog or the one in which it is cataloged.

420 Unrecoverable error: information in the catalog or in the one in which it is cataloged has been destroyed.

MME 500105: CLOSE

X0 File reference number of file/catalog to close

X6 Trap location

A Special word for communication file

The command CLOSE closes the file/catalog and invalidates the file reference number assigned to it.

DTSS

If the return bit is set in the access word of the file/catalog, then the file/catalog is returned to the job which passed it, generating a special interrupt number 5 (returned file). If it is a scratch file/catalog and the return bit is not set in its access word, then it disappears from the system and the job's scratch storage allotment is incremented by the length of the file or of the catalog header which disappeared.

If the master end of a communication file is closed, or if a slave end to which the communication file was passed without the return bit is closed, then a special interrupt of type 6 (file closed) is generated at all other ends of the file. The data word of this special interrupt is equal to the contents of the A-register of the closing job at the time of the CLOSE MME. The file reference number at the end issuing the CLOSE becomes invalid, and the file reference numbers at all other ends become invalid parameters for all MME's except CLOSE and RESET STATUS.

If a suspended job file is closed, then that job and all jobs below it are terminated (see the description of the EXECUTE command for a description of suspended jobs). All files open for these jobs are closed; if the return bit is set in the access word of a file/catalog open for any job being terminated then the file/catalog will be passed back up to the terminating job's supra job. If the supra job is accepting specials, each returned file will generate a special interrupt number 5 (returned file). The job issuing the CLOSE will also have its scratch and catalog word allotments incremented by the remaining allotments for the jobs being terminated. Status word 2 on the trap of the CLOSE will contain the CRU usage accumulated by the closed job file and all jobs which ran below it, in units of 1/64 milliCRU. (See the description of the TERMINATE command.)

If a terminated job file is closed, no special action is taken, but status word 2 will contain the CRU usage for that job and all jobs which ran below it.

Upon completion of the command a trap occurs to the location specified in X6.

Status Returns

0 Successful.

6 Communication file busy: the file is a communication file on which an operation is outstanding.

120 Busy: either a command is outstanding on the file/catalog or the file/catalog is passed to another job.

200 X0 parameter error: either X0 is zero or it does not contain a valid file reference number.

MME 500121: CONTINUE

X0 File reference number of job file

X4 New memory limits or zero

X6 Trap location

A Additional CRU limit for job

This MME restarts and runs jobs which have been suspended because:

(1) The job exceeded its allotted CRU usage;

(2) The job encountered a fault vector abort; or

(3) The job was stopped by a RESET STATUS.

The A-register will contain the amount of additional CRUs that the job can consume (in 1/64 milliCRU). A negative value indicates there is no limit on CRU usage. Any jobs which were running below the supra job and whose execution was suspended as a result of the supra job's being suspended, are also restarted.

If X4 is nonzero, then the maximum amount of main memory that the spawned job can request is reset to the minimum of the amount specified in X4 and the amount allowed to the spawning job. If X4 is zero, this limit remains the same as it was before.

A trap occurs to the location specified in X6 whenever the additional CRUs allotted to the job are exceeded, when status are reset on the job file, or when the job terminates or aborts. Status word 2 on this trap contains the total CRU usage accumulated by the spawned job and any jobs which ran below it.

If the trap occurred because the job was suspended (for the reasons listed above), the job may be restarted by another CONTINUE command. If the job terminated, however, another

CONTINUE may not be issued. A program can recognize a terminated file by its length, which is always zero, (this can be verified by a REQUEST STATUS command), whereas the length of a suspended job is never zero.

Status Returns

0 Job terminated sucessfully; bits 0-8 of status word 1 contain the access bits specified by the spawned job in the TERMINATE command.

1 Time limit exceeded; the job may be continued.

2 Job aborted for receiving a fault whose return word was nonzero. All jobs running below this one are suspended also. The job may be continued.

6 Job terminated due to an error on while swapping out.

20 Status was reset; the job may be continued.

120 Busy: another operation is outstanding on the job file.

200 X0 parameter error: X0 does not contain the file reference number of a nonterminated job file.

300 X4 parameter error: the new memory limits specified are less than the current length of the job file.

400 Recoverable error: this is a catch all status indicating an abnormal termination. It may indicate any of the following:

(1) I/O error swapping in the job

(2) Parity error while the job is running

(3) Unsuccessful overlay MME

(4) Job has somehow grown larger than slave memory

(5) Job's state vector has grown larger than 3K and job has not enabled large state vector permission.

MME 500131: COPY

X0 File reference number of source file

X1 Points to pointer to starting location M1 in source file or pointer to logical record number of device file

X2 File reference number of destination file

X3 Points to pointer to starting location M2 in destination file or pointer to logical record number of device file

X4 Flag bits

X6 Trap location

X7 Points to number of words N to copy

N contiguous words are copied from the source file (starting at word M1) to the destination file (starting at word M2). All other words of the destination file are unaffected.

Read permission is required on the source file; write permission is necessary on the destination file if the copy will alter any existing data in the destination file, and append permission is required if the copy will make the destination file longer. In any case, the destination pointer M2 must not be greater than the destination file's length.

During the actual copy, if the end of the source file is reached, an end-of-file condition occurs and data transmission stops. If any of the words transferred to the destination file are outside the range of that file, the file is extended if it is open with Append permission. Otherwise, an end-of-file condition occurs and data transmission stops. Upon completion of the copy, a trap occurs to the location specified in X6.

If X0 (X2) is zero, then the source (destination) file is assumed to be the issuing job's memory with Read and Write permission set. If X1 (X3) is zero, then the current position of the source (destination) file's read/write pointer is used. This option cannot be used with core, which has no implicit read/write pointer. Upon completion of a copy, the read/write pointers for both files are updated to point to the word following the last word accessed.

A COPY issued at a slave end of a communication file will generate a special interrupt number 11 (slave end issued READ) or 12 (slave end issued WRITE) at the master end. The second word of the interrupt will contain the length of the copy. (See COPY command in Chapter 5.) No data will be transferred until the master end issues a corresponding COPY. If the communication file is busy, either because another end has an operation outstanding on it, or because another end has reserved it, the slave end issuing the COPY command will be trapped with a status of 6. If the job at the master end of a communication file is not accepting special interrupts when a slave end issues a COPY, then that slave end will be trapped with a status of 7.

The slave end will be trapped with a 20 status (status reset) if the master end chooses to abort the copy by issuing a RESET STATUS command on the communication file.

The master end job will receive a slave issued SET POINTER special interrupt if:

(1) A job issues a COPY to the slave end of a communication file, and X3 is nonzero; or

(2) A job issues a COPY from the slave end of a communication file, and X1 is nonzero.

The pointer specified by the slave end is placed in the second word of the special interrupt pair. This special interrupt will immediately precede the slave issued READ or slave issued WRITE special received by the master end job.

The flag bits in X4 are divided into two 9-bit fields. Bits 0-8 are flags referring to the source file (specified in X0). Bits 9-17 refer to the destination file (specified in X2). They have the following meanings:

Bit	Function
0(9)	If the source (destination) file is a master communication file, trap the corresponding slave end only if a nonzero status return occurs or if this bit is zero. This bit must be zero if the source (destination) file is not a master communication file.
1(10)	If the source (destination) is a slave communication file, reserve that file for this end if this bit is 1. Release any prior reservation by this end if this bit is zero. If the source (destination) file is opened in shared mode, do not allow any copy-type operations to be initiated on this file after this copy operation completes, except for operations by this job on this file reference number. If the file is neither a communication file nor a shared file, this bit must be zero.
2(11) . . 8(17)	Reserved for future use. These bits must be zero.

A COPY in which both the source and destination files are communication files is not allowed. If the source (destination) file in a COPY is a device file, then the destination (source) file must be a core file, the issuing job's core, or a communication file whose other end is being copied to or from a core file or some job's core. If the source (destination) file is opened in shared mode, then the destination (source) must be the issuing job's memory.

Upon initiation of a COPY into a scratch (cataloged) file which is open with append permission, the job's scratch (catalog) word allotment is decremented by the amount by which the destination file may be extended. At the completion of the COPY, the job's scratch (catalog) word allotment is incremented by the number of words charged to this allotment but not appended to the destination file.

If the status return in status word 1 is not between 100 and 360 on completion of the COPY, then status word 2 contains the difference between the number of words transferred and the number of words requested (i.e. it contains minus the number of words not tranferred). If the status in word 1 is between 100 and 360, then status word 2 is zero. If either the source or the destination file is a device file, then the lower half of status word 1 contains the status return from that device.

Status Returns

0 Successful: all words were transferred.

1 Source file exhausted: more words were requested but could not be transferred.

2 Destination file exhausted: some words in the source file were not transferred since the destination file was full and was not open with Append permission.

3 Command inappropriate: a COPY issued at the master end of a communication file does not match a corresponding operation outstanding at a slave end.

4 The source file pointer is out of bounds: either,

   (1) The pointer was initially out of bounds;

   (2) A MEMORY REQUEST released job storage involved in the copy; or

   (3) An attempt was made to copy from the pure region of a job file which was swapped out of main memory.

5 The destination file pointer is out of bounds: either,

   (1) The pointer was initially out of bounds;

(2) A MEMORY REQUEST released job storage involved in the copy; or

(3) An attempt was made to copy into the pure region of a job file which was swapped out of main memory.

6 The communication file at whose slave end this command was issued is busy (a) because another end has an operation outstanding, or (b) because another end has reserved the communication file.

7 The master end of the communication file at whose slave end this command was issued is not accepting special interrupts.

20 Status was reset on the COPY.

40 Quotas exceeded: it is possible for the COPY to exceed the job's remaining scratch or catalog word allotment; no words were transferred.

60 Out of storage: the Executive could not allocate sufficient storage for a required extension of the destination file.

100 Access error: either the source file is not open with Read permission, or the destination file is not open with the necessary Write or Append permission. Bits 0-8 of status word 1 contain the missing permission bits.

120 Busy: another command is outstanding on either the source or the destination file.

200 X0 parameter error: X0 does not contain the file reference number of a file, or X2 specifies a shared file and X0 is not zero.

220 X1 parameter error: X1 points to an out-of-bounds location, or both X0 and X1 are zero.

240 X2 parameter error: either X2 does not contain the file reference number of a file, both source and destination files are communication files, or X0 specifies a shared file and X2 is not zero.

260 X3 parameter error: X3 points to an out-of-bounds location or both X2 and X3 are zero.

300 X4 parameter error: an invalid flag bit is set in X4.

360 X7 parameter error: X7 points out of bounds.

400 Recoverable error: an error occurred during transmission of data. The lower half of status word 1 contains the status from the device which generated the error. See Chapter 6 for a description of device statuses. This status can occur if a device file is being accessed via a communications file and the job has swapped out of main memory.

420 Unrecoverable error: a COPY was attempted between a device file and another file not appropriate for device file copies.

MME 500132: DRIVE

X0 File reference number of file

X1 Pointer to Data Control Words (Drive with DCW's only)

X4 Flag bits

X6 Trap location

X7 Number of DCW's (Drive with DCW's only)

A Drive type (AU) and function or data (AL)

The DRIVE command may be issued only on a communication file or on a device file. Drives on communication files generate special interrupt at some other end of the file. Drives on device files cause the drive function to be applied to the device. Upon completion of the command, a trap occurs to the location specified in X6.

Drive types valid at the slave end of a communication file are 0 (set communication file mode) and 12 (single device action). These drives generate special interrupt numbers 0 and 12, respectively, at the master end of the communication file. The lower half of status word 2 in such special interrupts contains the drive data from AL.

Valid drive types at the master end of a communication file are 1 (read communication file) and 3 (break), which generate the corresponding special interrupt:

(1) At the lowest slave end accepting special interrupts, providing no slave end has reserved the communication file; or

(2) At the slave end which has reserved the communication file.

Drive type 3 will generate a special interrupt number 3 at the lowest slave end which is accepting specials, which has eXecute permission on the communication file, and which is not lower than the slave end to which the file is reserved (if it is indeed reserved to a slave end).

If a DRIVE is issued on the slave end of a communication file that is busy due to the action of another end, the DRIVE is rejected with a status of 6. If on any communication file drive the end to which the special interrupt should be given is not accepting special interrupts, the DRIVE is trapped with a status of 7.

The flag bits in X4 have the following meanings:

Bit	Function
0	Must be zero.
1	Must be zero unless X0 contains the file reference number of a slave communication file. In that case, the communication file will be reserved for the end issuing the drive if this bit is 1. Any prior reservation by this end is released if this bit is zero.
2-17	Must be zero.

reservation bit

Following is a list of drive functions which may be issued on the slave end of a terminal communication file. The drive type (contained in AU) for these drives is zero.

000000	Set line-by-line input mode
000001	Set file building input mode
000002	Set file building mode for paper tape
000003	Set direct input mode
000004	Set command input mode
000005	Set command mode for paper tape
000007	Answer back drum request
000020	Set full-duplex output mode
000021	Set half-duplex output mode

000022	Set direct output mode
000023	Reset direct output mode
000024	Set Friden output mode
000025	Reset Friden output mode
000026	Stop output
001000	Output any currently pending warning

There are two valid drive types for device files: 12 (single device action) and 24 (drive with DCW's). A drive of type 12 applies the function specified in AL to the device before the drive is trapped. The valid function codes are listed in Chapter 6.

A drive type of 24 also applies the function in AL to the specified device and then traps. The valid function codes are listed in Chapter 6.

A list of Data Control Words (DCW's) pointed to by X1 must be supplied for this drive. See Chapter 6 for a description of DCWs.

Status word 2 on a completed single action device drive contains the status return from the device. The format of this status return is as follows:

Bits	Meaning
0	ICM sync bit (normally on).
1-5	Major status.
30-35	Record count residue (number of records requested minus the number transferred).
6-11	Minor status.
12-17	Queue address bits; the operation was terminated by an initiate interrupt if between 20 and 37 and by a terminate interrupt if between 40 and 57.
18-23	ICM status (0 is good)

24-29 Return code from Executive I/O routine.

Status Returns

0 Successful.

1 Incomplete data transfer; no error.

2 End-of-file encountered on tape or last batch button depressed on card reader.

3 Recoverable error; details in major and minor status.

4 Unrecoverable error; do not reissue command.

5 Operation on device was timed out.

If a DRIVE with DCW's fails with a bad DCW status, status word 2 contains the number of the faulty DCW. If a drive with DCWs fails with a status of 1, 2 or 400, status word 2 contains the DCW residue of the last DCW accessed. Bits 0-5 of status word 1 of a DCW DRIVE contain the record count residue if the device was accessed. Bits 18-35 of status word 1 contain the first 18 bits of the device status return.

For further information concerning DRIVE functions and status returns for device DRIVEs, see Chapter 6, "Device Files".

Except as noted above for device drives, status word 2 in the trap of a DRIVE command is zero.

Status Returns

0 Successful.

1 Not all data was transferred on a Drive with DCWs, but data transferred was correct.

2 End-of-file condition detected on device file drive.

3 Command inappropriate: the DRIVE was issued at the master end of a communication file while a slave end of that file had an operation outstanding.

6 Communication file busy: the DRIVE was issued on the slave end of a communication file which is busy, either because another end has an operation outstanding on it, or because another end has reserved it.

7 The end of the communication file which was to have received the special interrupt was not accepting special interrupts.

10 Bad DCW: DCW points to nonexistent address, invalid action code, or superfluous bits were found in DCW.

20 Status was reset on a drive.

120 Busy: another operation is outstanding on the file.

140 A register parameter error: the specified drive type is not valid. On DRIVE with DCW's this may occur because of an insufficient DCW complement.

200 X0 parameter error: X0 does not contain the file reference number of a communication file or of a device file.

220 X1 parameter error: DCW list pointer out of bounds.

300 X4 parameter error: X4 contains invalid flag bits.

360 X7 parameter error: the number of DCW's is zero or indicates DCW's past the boundary of main memory.

400 Recoverable error: an error occurred on a device file drive; the major and minor status bits in status word 1 identify the error.

420 Unrecoverable error: the device drive function specified is not valid.

MME 500143: DUPLICATE

X0 File reference number to duplicate

X1 Reserved for future use (must be zero)

X2 Reserved for future use (must be zero)

X3 Reserved for future use (must be zero)

X4 Accesses to request for new file reference number

X5 Reserved for future use (must be zero)

X6 Trap location

X7 Reserved for future use (must be zero)

The DUPLICATE command creates another file reference number for the file/catalog specified with X0.

Register X0 must contain the file reference number of a cataloged file/catalog. Any accesses not specified in X4 will not be available with the new file reference number. Accesses specified in X4 will be granted if they would be allowed on an ALTER ACCESSES and they do not conflict with accesses which this and other jobs have on the file.

On completion of the command a trap occurs to the location specified in X6. If the status return is less than 2 then the lower half of status word 1 contains the new file reference number.

Status Returns

0 Successful: the accesses in bits 0-8 are available on the new file reference number and are the same as those requested.

1	Partial success: the accesses in bits 0-8 are available on the new file reference number but differ from those requested.
2	Lockout: due to usage conflicts the file cannot be open with another file reference number
3	Inappropriate: X0 does not contain the file reference of a cataloged file/catalog
40	Quotas exceeded: the job has filled its allotted state vector length
120	Busy: another command is outstanding on the file/ catalog specified in X0
200	X0 parameter error: X0 does not contain the file reference number of a file/catalog

MME 500112: ERASE

X0 File reference number of initial catalog to search

X1 Points to name of file/catalog to erase

X2 Reserved for future use (must be zero)

X3 Points to password (or is zero)

X4 (bits 0-8) must be zero
(bits 9-17) trap bit mask

X5 Reserved for future use (must be zero)

X6 Trap location

X7 Number of entries in treename (if zero, then 1 is assumed)

AQ Slave trap identification words

The ERASE command attempts to open the specified file/catalog (with the specified password if X3 is nonzero and without a password otherwise) in the specified catalog with Read, Write, and Append permissions. If it is successful it then destroys the file/catalog together with its catalog entry.

Catalog quota checks are suspended for the ERASE command; hence, it can be used to destroy files in catalogs whose quotas have been exceeded. Conflict checks with accesses currently held by other jobs having the given file/catalog open are suspended for all accesses except Write. An ERASE command may therefore be used to destroy a file/catalog which is open, provided that it is not open with Write permission. With the exception of this and the catalog quota check, an ERASE command is identical in effect to the sequence of commands OPEN, UNCATALOG, and CLOSE.

Upon completion of the command a trap occurs to the location specified in X6.

Status Returns

0 Successful: the file/catalog was erased.

2 Lockout: the file/catalog is currently in use with Write permission.

3 Not found: Search permission was available on some intermediate catalog and the next file/catalog was not found in it.

4 Protection violation: Search permission was available on the next-to-last catalog and the password specified for the file/catalog was wrong.

5 Fail: Search permission was available on the next-to-last catalog, and one of the Read, Write, or Append permissions was not available.

6 Bad tree name: Search permission was available on some intermediate catalog and the next entry in the treename was not the last entry, but was the name of a file rather than a catalog.

7 Fetch error: either

(1) An error of type 3, 4, 5, or 6 occurred and Search permission was not available on the catalog in which the search was being performed; or

(2) Search permission was not available on the (N-1)st catalog and Fetch permission was not available on the file/catalog.

11 ERASE disallowed: the named file/catalog is a device file.

120 Busy: another command is outstanding on the catalog.

200 X0 parameter error: X0 does not contain the file reference number of a catalog.

220 X1 parameter error: the pointer to the name is out of bounds.

240 X2 parameter error: X2 is nonzero.

260 X3 parameter error: the pointer to the password is out of bounds.

300 X4 parameter error: bits 0-8 of X4 are nonzero.

320 X5 parameter error: X5 is nonzero.

360 X7 parameter error: X7 is greater than 10 (maximum of 10 names in the list of entries pointed to by X1).

400 Recoverable error: an error occurred in reading the catalog.

420 Unrecoverable error: information in the catalog has been destroyed.

MME 500117: EXECUTE

X0 File reference number of source file

X1 Length of scratch area to append to source file

X3 Job type (bits 0-8)

X4 Maximum amount of memory job is allowed

X5 Pointer to run list

X6 Trap location

X7 Pointer to eight-word block containing registers

A Job CRU limit (in 1/64 milliCRU)

Q Job access mask

The source file, which must be open with execute permission, is run as a new job with the CRU limit specified and with the files in the run list open for it. The new job is started at location zero with the specified registers and with all indicators off except the zero indicator, which is on.

A job file is created for the new job, and its file reference number is returned in bits 18-35 of status word 1 of the trap block specified by X6 before execution of the creating job is resumed. The job file remains open until closed by the creating job.

The amount of memory with which the new or spawned job is run is determined as follows: the length of the source file is rounded up to a multiple of 1K (1024 words) and to this length is appended a scratch area of the length specified in X1, also rounded up to a multiple of 1K. The spawned job can issue a MEMORY REQUEST to change the size of its memory, but can request no more memory than the lesser of the amount specified in X4 and the amount of memory allowed to the job which issued the EXECUTE command.

Bits 0-8 of X3 are ANDed with the job type bits allowed to the job issuing the EXECUTE to form the permissions allowed to the spawned job. (See the description of the ENABLE command for a list of these permissions.) At the time the job is run, the permissions for which it is enabled are Large State Vector, Priority Scheduling, Crash, and Log. The spawned job must enable itself for any other allowable permissions by issuing an ENABLE command.

Run List (Pointed to by X5)

```
+---------------------+
|                     |
+--- Spawner's ID ----+
|                     |
+---------------------+
|    Catalog Quota    |
+---------------------+
|    Scratch Quota    |
+---------------------+
|                     |
+---             -----+
.
+... passed files ...+
.
+---------------------+
|          Ø          |
+---------------------+
```

The run list pointed to by X5 must be at least five words long and be terminated by a zero. The maximum length of a run list is 32 words. If no zero word is encountered in the first 32 words, the Executive will stop scanning the run list. The first two words in the run list are the identifying words for the spawned job. These words can be read by the spawned job and all jobs which are run below it by issuing a REQUEST STATUS on file reference number 0.

The third and fourth words of the run list limit the number of words that the spawned job can append to cataloged and scratch files. These limits should be less than the limits of the supra job. If they are not, the Executive will substitute the supra job's limits. The limits actually passed to the spawned job will be indicated to the supra job in the third and fourth words of the run list. If any file operation issued by the spawned job results in a situation where these limits are exceeded, then that operation will be trapped with a status return of 40 (quotas exceeded). The catalog and scratch word allotments for the spawned job are subtracted from the corresponding allotments for the spawning job. Upon termination of the spawned job, the job which ran it will have its catalog and scratch word allotments incremented by the allotments remaining for the terminating job.

The remainder of the run list contains a list of file/catalogs to be passed to the new job. A zero word indicates the end of the list. Each word in the list contains the following information:

Passed File Word

ACCESS	PREF	FRN or Ø
0 8	9 17	18 35

Bits	Meaning
0-8	Access to pass with file
9-17	Preference (for created file)
18-35	File reference number or zero

The nth word in this list causes the file/catalog with the file reference number specified in bits 18-35 to be passed to the new job with the access specified in bits 0-8. In the new job this file/catalog will have file reference number n.

(1) If the file reference number in bits 18-35 is 0, then a file will be created and passed to the new job.

(2) If bit 1 of the access is 0, then a scratch file will be created with the preference specified in bits 9-17 and passed.

(3) If bit 1 of the access is 1, then a communication file will be created and its slave end passed.

(4) If the return bit (bit 2) is set in the access word for a created scratch file or if a communication file is created, then the file reference number of the file for the creating job will be returned in bits 18-35 of the run list entry before execution of the job is resumed.

Job files, master ends of communication files, and file/catalogs which have been passed already or which are currently busy cannot be passed to the new job. The file reference number of a file/catalog which has been passed without the return bit set becomes invalid. File/catalogs other than slave ends of communication files which are passed with the return bits set become busy until they are closed and returned by the spawned job. A file/catalog which has the return bit set in its access word must be passed with the return bit set. A scratch file/catalog which is passed without the return bit set automatically has all accesses set for the new job. As opposed to files passed by the PASS command, no messages accompany files passed through the run list. The current setting of the read/write pointer for a file/catalog is preserved when it is passed.

Job Access Mask (Q register)

	TRAP	PREF
0 17 19	20 28	29 35

The job access mask of the issuing job is ANDed with the job access mask specified in the Q-register to form the access mask for the new job. The bits in this access mask have the following significance:

Bits	Meaning
0-19	Unused
20-28	Permission bits for master trap program (see CATALOG)
29-35	Preferences to allow when creating scratch file/catalogs (see OPEN SCRATCH)

The job file for a successfully spawned job remains busy until a trap occurs to the location specified in X6. The lower half of status word 1 contains the file reference number of the job file. Status word 2 contains the total running time (in CRU's) accumulated by the spawned job and any jobs which ran below it. The trap occurs whenever the CRUs allotted to the spawned job is exceeded, status is reset on the job file, or the job terminates or is aborted.

If the trap occurred because the CRU limit was exceeded, status was reset, or the job aborted, the job is suspended, and its execution may be continued by using a CONTINUE command. If, however, the job terminated, either successfully, or because of an error on swap-out, or because of a recoverable error, a CONTINUE may not be issued. If a suspended job file is closed rather than being continued, status word 2 on the trap of the CLOSE command will contain the total CRU usage (in 1/64 milliCRU) of the job. This will reflect any additional time charged to the job by the Executive in the process of terminating it and all jobs below it. If a terminated job file is closed, the CRU usage is returned in the trap of the CLOSE. (See CLOSE.) A terminated job file can always be recognized by the fact that its length (available through a REQUEST STATUS command) is always zero, while a suspended job never has zero length.

Status Returns

0 Job terminated successfully: bits 0-8 of status word 1 contain the access bits specified by the spawned job in the TERMINATE command.

1 Time limit exceeded: the job may be continued.

2 Job aborted: the job received a fault whose return word was nonzero. All jobs running below this one are suspended also. The job may be continued.

5 Run list error: the job was not run. The lower half of status word 2 contains a pointer to the entry in the run list in error. The upper half of status word 2 contains one of the following error types:

- 0 Oversize run list: a maximum of twenty files may be passed.
- 60 System out of storage: the Executive's table of opened files is full.
- 120 Busy: the file reference number in the pass list is that of the source file for the RUN command.
- 240 Bad file reference number in bits 18-35: either
  - (1) The file reference number is invalid;
  - (2) It is the file reference number of a busy, passed, or master communication file; or
  - (3) It is zero and the job issuing the RUN command has a job access mask which does not permit it to specify a preference for creating a scratch file.
- 300 Bad access in bits 0-8: either
  - (1) Extraneous bits are set in the access word;

(2) The return bit is not set in the access word, though the file/catalog was passed to the issuing job with the return bit set;

(3) Bit 0, the catalog bit, is zero (is one) and the file/catalog being passed is a catalog (is a file);

(4) No accesses are present in bits 5-8;

(5) The catalog bit is one and bits 18-35 are zero; or

(6) The return bit is set in the access word for the creation of a communication file.

DTSS

6 Job terminated due to an error on a swap-out.

20 Status was reset: the job may be continued.

40 Quotas exceeded: A job possessing no master trap bits may have at most one run/execute outstanding.

60 Out of storage: either

(1) The Executive's table of running jobs is full;

(2) The length of the new job including the appended scratch area, if any, exceeds the system limit on job length; or

(3) The depth of the job in the job tree is too great to allow it to issue EXECUTE commands.

100 Access error: the source file is not open with eXecute permission; bits 0-8 of status word 1 contain the missing permission bit.

120 Busy: another operation is outstanding on the source file.

140 A register parameter error: the time specified for the new job to run is zero.

200 X0 parameter error:

(1) X0 is zero;

(2) X0 is not the file reference number of a scratch or cataloged file;

(3) X0 is the file reference number of a file whose length is zero or greater than 2^18.

300 X4 parameter error: the initial size of the new job, including the appended scratch area, if any, exceeds the limits specified.

320 X5 parameter error: the pointer to the run list is out of bounds.

360 X7 parameter error: the pointer to the registers is out of bounds.

400 Recoverable error: this is a catch all status indicating an abnormal termination. It may indicate any of the following:

(1) I/O error swapping in the job

(2) Parity error while the job is running

(3) Unsuccessful overlay MME

(4) Job has somehow grown larger than slave memory

(5) Job's state vector has grown larger than 3K and job has not enabled large state vector permission.

MME 500142: LOG

X0 Reserved for future use (must be zero)

X1 Pointer to pointer to memory location M1

X2 Reserved for future use (must be zero)

X3 Reserved for future use (must be zero)

X4 Flag bits

X5 Reserved for future use (must be zero)

X6 Trap location

X7 Pointer to number of words N to copy

The ASCII message which begins at location M1 and extends for N words is entered into the system log. The flag bits in X4 perform the following functions:

Bit	Meaning
0	Ring the console alarm if the system logging device is a console.
1	Suppress log device output and direct this entry to the system log file only, if one exists.

Bit zero in X4 has no effect if the logging device is not a console, or if no logging device output occurs. Bit one in X4 has no effect if the Executive has no current log file at the time of the MME. A job which does not possess the Log enable bit may not issue this MME. A job may have at most one LOG MME outstanding at a given time. Upon completion of the MME, a trap occurs to the location specified by X6. If the logging device is accessed, the lower half of status word 1 contains the physical device status.

Status Returns

0 Successful.

1 LOG MME outstanding. An erroneous attempt has been made to perform two log MME's concurrently.

2 Buffer not available. This is a recoverable error signifying a temporary scarcity of main memory.

3 ESCAPE sequence error. The ASCII message contains too many exclamation points, question marks, or control characters other than carriage return, line feed, or horizontal tab.

200 X0 parameter error: X0 is not zero.

220 X1 parameter error: X1 is out of bounds or points to an invalid core pointer.

240 X2 parameter error: X2 is not zero.

260 X3 parameter error: X3 is not zero.

300 X4 parameter error: X4 contains undefined flag bits.

320 X5 parameter error: X5 is not zero.

360 X7 parameter error: X7 is out of bounds, points to a number N which is larger than thirty words (120 characters) or zero, or specifies a length which exceeds the available memory.

400 Recoverable error: a recoverable I/O error has occurred to the log device.

440 Not enabled: the job is not enabled with Log permission.

MME 500102: OLD ERASE

X0 File reference number of catalog to search

X1 Points to name of file/catalog to erase

X3 Points to password (or is zero)

X6 Trap location

AQ Slave trap identification words

DTSS

The OLD ERASE command is exactly like the ERASE command, except an X4 of zero is assumed (but not checked) — which means that only one filename can be specified — and the unused registers do not need to be zero.

Status Returns

0	Successful: the file/catalog was erased.
2	Lockout: the file/catalog is currently in use with Write permission.
3	The desired file/catalog was not found in the searched catalog.
4	Protection violation: the specified password was wrong.
5	Fail: one of the Read, Write, or Append permissions was not allowed.
11	OLD ERASE disallowed: the named file/catalog is a device file.
100	Access error: the catalog is not open with Search permission; bits 0-8 of status word 1 contain the missing permission bit.
120	Busy: another command is outstanding on the catalog.

200 X0 parameter error: X0 does not contain the file reference number of a catalog.

220 X1 parameter error: the pointer to the name is out of bounds.

260 X3 parameter error: the pointer to the password is out of bounds.

400 Recoverable error: an error occurred in reading the catalog.

420 Unrecoverable error: information in the catalog has been destroyed.

MME 500125: OLD READ CATALOG AND OPEN FILES

X0 File reference number of catalog

X1 Points to number M1 of first entry in the catalog to copy

X3 Points to pointer to memory location M2

X4 Access for OPEN (bits 0-8)

X5 Coded date

X6 Trap location

X7 Points to number of words N to copy

AQ Slave trap identification words

The OLD READ CATALOG AND OPEN FILES command is like the READ CATALOG AND OPEN FILES command with two exceptions. Register X2 is not reserved for future use. Register X5 does not point to coded dates, but contains a coded date last modified. Information for files other than catalogs is provided if the file's date last modified is not less than the coded date given in X5.

This is equivalent to a READ CATALOG AND OPEN FILES command with X5 pointing to a word with zero in the upper half and the coded date last modified in the lower.

MME 500130: OLD REPLACE

X0 File reference number of catalog

X1 Points to name of file/catalog to replace

X2 File reference number of scratch file/catalog

X3 Points to password or is zero

X6 Trap location

AQ Slave trap identification words

The OLD REPLACE command is exactly like the REPLACE command, except an X4 of zero is assumed (but not checked) — which means that only one filename can be specified — and the unused registers do not need to be zero.

Status Returns

0 Successful: the file/catalog was replaced and is now open with the accesses in bits 0-8.

1 Partial success: the file/catalog was replaced and is now open with the accesses in bits 0-8. Append permission has been masked off since the searched catalog's MAX is exceeded.

2 Lockout: the cataloged file/catalog is currently in use with Write permission.

3 The desired file/catalog was not found in the searched catalog.

4 Protection violation: the specified password was wrong.

5 Unsuccessful: one of the Read, Write, or Append permissions was not allowed.

11 OLD REPLACE disallowed: the named file/catalog is a device file.

40 Quotas exceeded: either

(1) The length of the scratch file or catalog header exceeds the job's remaining catalog word allotment; or

(2) The specified catalog's MAX is exceeded, and the length of the scratch file or MAX of the scratch catalog is greater than the length or MAX of the file/catalog to be replaced.

60 Out of storage: either

(1) The Executive's table of opened files is full, or

(2) A new entry had to be made in the catalog for the replacement and

(a) The catalog already had 4095 entries; or

(b) The Executive could not allocate sufficient storage for a required extension of the catalog; or

(c) It is not possible to append to the catalog since it either is a main-memory catalog or is already allocated in twelve fragments.

100 Access error: the initial catalog to search is not open with Search permission; bits 0-8 of status word 1 contain the missing permission bit.

120 Busy: another operation is outstanding on either the catalog or the scratch file/catalog.

200 X0 parameter error: X0 does not contain the file reference number of a catalog.

220 X1 parameter error: X1 points out of bounds.

240 X2 parameter error: either

(1) X2 does not contain the file reference number of a scratch file/catalog;

(2) The scratch file/catalog was passed to the issuing job with the return bit set;

(3) The type of the scratch file/catalog does not match the type of the file/catalog it is to replace; or

(4) The file/catalog has a preference of 0, 1, 2, or 4 and may only replace a file in a core catalog (a catalog of preference 0).

260 X3 parameter error: the pointer to the password is out of bounds.

400 Recoverable error: an error occurred in reading the catalog.

420 Unrecoverable error: information in the catalog has been destroyed.

MME 500101: OPEN

X0 File reference number of initial catalog to search (0->:MFD)

X1 Points to treename of file/catalog to open

X3 Points to password of file/catalog to open (if bit 1 of X4 is 1)

X4 Access desired on file/catalog (bits 0-8);
Trap bit mask (bits 9-17)

X6 Trap location

X7 Number of entries N in treename (if bit 4 of X4 is 1)

AQ Slave trap identification words

The specified file (denoted by a list of filenames which form a "treename") is located in the file system and opened. A file reference number is returned, which is used on all subsequent operations on the file.

The treename pointed to by X1 begins with a list of catalog names and ends with the name of the file/catalog to open. If the fetch bit (bit 4) of X4 is 1, then X7 specifies the number of entries N in the treename, which must be between one and ten. If bit 4 of X4 is 0, then N is taken to be one.

Note that if the fetch bit (bit 4 or bit 22) is not on in the access word of a file/catalog, then it can be opened only if a job can open the catalog in which it is located with search permission. If, however, the fetch bit is on in the access word of a file/catalog, then a job need not be able to open the catalog(s) above it in the catalog structure. The file/catalog may be opened by specifying its position in the catalog structure and passing only the access checks associated with the file/catalog itself. Thus files and catalogs may be protected either collectively by the accesses permitted for catalogs at higher levels in the catalog structure, or individually by their own accesses. Also note that whenever the master trap program is run during an open, the accesses of the issuing job will be limited by the trap bit mask in bits 9-17 of X4. In this case, for any bit on the mask, the corresponding trap protection bit cannot be passed to the opened file, regardless of the accesses with which the issuing job is running.

The procedure for opening the file/catalog specified by the treename is as follows. The initial catalog specified by X0 is searched for the catalog specified by the first entry of the treename; if X0 is zero, then the Master File Directory is searched. As each succeeding catalog in the treename is found, it is searched for the catalog specified by the next entry of the treename. Finally the (N-1)st catalog is searched for the file/catalog specified by the Nth entry of the tree name.

For each intermediate catalog in the tree name (i.e. catalogs 1 to N-1), the accesses available are determined in one of two manners:

(1) If Search permission was available on the previous catalog, then the accesses available on the current catalog are those allowed without a password (bits 3-8 of the current catalog's access word), or those returned by the Executive trap program if the catalog is so protected. If Owner permission was available on the previous catalog and the current catalog is not protected by the Executive trap program or the Executive trap is successfully run, then all accesses are allowed on the current catalog.

(2) If Search permission was not available on the previous catalog, then the accesses available on the current catalog without a password (bits 3-8 of the current catalog's access word), or those returned by the Executive trap program if the catalog is so protected, are examined for the presence of the fetch bit (bit 4). If the fetch bit is on, then accesses are granted on the current catalog just as if Search permission had been available on the previous catalog. If the fetch bit is not on, then no accesses are granted on the current catalog.

When the (N-1)st catalog is searched for the file/catalog specified by the last entry of the tree name, the accesses available on that file are determined as follows. If this file/catalog is found, bit 1 of X4 is 1, and file/catalog is not protected by a trap program, then the password pointed to by X3 is checked against the password protecting the file/catalog. If the password check succeeds or is not made, then the accesses allowed on the file/catalog are determined in one of two manners:

(1) If Search permission was available on the (N-1)st catalog, then the access bits requested in X4 are ANDed with the access bits allowed by the access word of the file/catalog (bits 3-8 of the access word if the file/catalog is being opened without password, bits 21-26 if the file/catalog is being opened with a password, or the access bits returned by the trap program protecting the file/catalog). If Owner permission was available on the (N-1)st catalog, and the file/catalog is not protected by a trap program and is protected by a successfully run master trap, then all requested accesses are allowed on the file/catalog. In addition, if the opening job possesses all master trap bits and has a zero mask in X4 in addition to Owner permission on the (N-1)st catalog, then any slave trap program is bypassed and all permissions are granted on the file/catalog.

(2) If Search permission was not available on the (N-1)st catalog, and the fetch bit is on in the access word of the file/catalog, that is,

(a) Bit 4 of the access word if the file/catalog is being opened without a password;

(b) Bit 22 if the file/catalog is being opened with a password; or

(c) Bit 4 of the accesses returned by the trap program protecting the file/catalog;

then the accesses allowed on the file/catalog are the same as if Search permission had been available on the (N-1)st catalog. If Search permission was not available on the (N-1)st catalog and the fetch bit is not on in the access word of the file/catalog, then no accesses are allowed on the file/catalog.

Next the access bits allowed are checked for use conflicts with accesses currently held by other jobs whose file/catalog is open. Access bits that are in conflict are masked off. In addition, Append permission is masked off if the MAX of the (N-1)st catalog has been exceeded. All access bits are masked off if the allocated storage for the (N-1)st catalog exceeds twice its permissible MAX. Finally, bit 0 of the access is set to 1 if the file/catalog is a catalog. The file/catalog is opened if any accesses remain in bits 5-8.

On completion of the command, a trap occurs to the location specified in X6. If the file/catalog was opened successfully, then the access bits obtained are returned in bits 0-8 of status word 1, the new file reference number of the opened file/catalog in bits 18-35 of status word 1, and the length of the opened file or the MAX of the opened catalog in status word 2. If the command was unsuccessful, then the upper half of status word 1 contains the reason for the failure, and the lower half contains zero.

In addition to the access bits returned in status word 1 of the trap, nine user access bits can be obtained by issuing a REQUEST STATUS. These user access bits are obtained from bits 27-35 of the opened file/catalog's access word if the file/catalog was opened with a password. If the file/catalog was opened without a password or if it was protected by a trap program, the user access bits are obtained from bits 9-17 of the access word.

The read/write pointer of the opened file/catalog is initialized to zero.

Status Returns

0 Successful: the accesses in bits 0-8 were obtained and are the same as those requested.

1 Partial success: the accesses in bits 0-8 were obtained but differ from those requested. This status return can occur if bit 0 of the requested access is 0 (is 1) and the opened file/catalog is a catalog (is a file).

2 Lockout: due to usage conflicts, none of the requested accesses were available on the file/catalog.

3 Not found: Search permission was available on some intermediate catalog and the next file/catalog was not found in it.

4 Protection violation: Search permission was available on the (N-1)st catalog and the password specified for the file/catalog was wrong.

5 Fail: Search permission was available on the (N-1)st catalog and none of the requested accesses were available on the file/catalog. This can occur if the file/catalog is slave-trapped and the slave trap program is not correctly cataloged.

6 Bad tree name: Search permission was available on some intermediate catalog and the next entry in the treename was not the last entry, but was the name of a file rather than a catalog.

7 Fetch error: either

(1) An error of type 3, 4, 5, or 6 occurred and Search permission was not available on the catalog in which the search was being performed; or

(2) Search permission was not available on the (N-1)st catalog and Fetch permission was not available on the file/catalog.

10 Off-line file.

40 Quotas exceeded: either

(1) The job has exceeded its allotted state vector length; or

(2) Allocated storage for the (N-1)st catalog is over twice the permissible MAX, and files must be erased from this catalog before any can be opened.

60 Out of storage: the Executive's table of opened files is full.

100 Access error: the initial catalog to search is not open with Search permission; bits 0-8 of status word 1 contain the missing permission bit.

120 Busy: another command is outstanding on the catalog specified in X0.

200 X0 parameter error: X0 does not contain the file reference number of a catalog.

220 X1 parameter error: the pointer to the tree name is out of bounds.

260 X3 parameter error: bit 1 of X4 is 1 and the pointer to the password is out of bounds.

360 X7 parameter error: bit 4 of X4 is 1, and X7 contains a zero or a number of file names greater than ten.

400 Recoverable error: an error occurred in reading an intermediate catalog.

420 Unrecoverable error: information in an intermediate catalog has been destroyed.

MME 500100: OPEN SCRATCH

X4 File/catalog flag (bit 0)

X5 Preference (for files only)

X6 Trap location

A Catalog MAX or preallocation length N (Load-Dump enabled jobs only)

Q Estimated number of catalog entries (special catalog enabled jobs only)

DTSS

A scratch file is opened if bit 0 of X4 is zero. Otherwise, a scratch catalog is opened. The scratch file/catalog is created with all permissions (i.e. Read, Write, Append, Search/eXecute, and Owner).

Preference is an integer between 1 and 7 which specifies the class of storage preferred for the file being created. Classes of storage are as follows:

1 Swap files
2 System files (BASIC, ALGOL, ...) or monitor scratch files
3 Catalogs
4 Scratch files
5 Special data base
6 Saved files
7 Permanent data base

The preferences which the issuing job can assign are limited by bits 29-35 in its job access mask (set by the supra job on a RUN or EXECUTE command). A job can assign a preference n only if bit 28+n in its access mask is 1. Attempts to assign other preferences will result in the Executive's rounding the preference to a higher number if possible and to a lower one otherwise. All catalogs will be assigned a preference of 3 by the Executive. Assigned preferences continue to hold when a scratch file/catalog is cataloged.

The OPEN SCRATCH command normally does not preallocate any storage for a scratch file; such storage is allocated as needed whenever information is appended to the file. Where N is specified in the A-register, an OPEN SCRATCH command issued by a job enabled with the Load-Dump permission will cause N words to be preallocated for the scratch file. This preallocation is designed to alleviate excessive fragmentation of storage.

The parameters in the A- and Q-registers are normally applicable only when a scratch catalog is being created. In that case MAX sets the maximum number of words which can occur in that catalog and in all files and catalogs cataloged beneath it. A negative MAX sets this maximum to infinity and can be specified only by those jobs enabled with the Special Catalog permission. Likewise, only such jobs can specify an estimated number of catalog entries in Q. This estimate is used as a guide by the Executive in allocating storage for the catalog so as to optimize disk accesses for catalog searches. The Executive will determine this initially allocated length for catalogs created by all nonenabled jobs.

When a scratch catalog is opened, the job's remaining scratch word allotment is decremented by the length of the scratch catalog header.

A trap to the location specified in X6 occurs upon completion of the command. If the command is successful, status word 1 contains the file reference number of the created file in its lower half.

Status Returns

0 Successful.

40 Quotas exceeded: either

(1) The job has exceeded its alloted state vector length;

(2) The job access mask for the job does not permit it to specify any preference for a scratch file/ catalog; or

(3) The charge for writing a scratch catalog header exceeds the job's remaining scratch word allotment.

60 Out of storage: sufficient storage is not available for the scratch catalog header or for preallocating a scratch file.

140 A-register parameter error: either

(1) The job is not enabled with the Special Catalog permission and is specifying a negative MAX for a scratch catalog; or

(2) The job is enabled with the load-dump permission and register A is negative.

160 Q-register parameter error: the job is enabled with the Special Catalog permission and is specifying more than 4095 entries for a scratch catalog.

400 Recoverable error: an error occurred in writing the scratch catalog header.

MME 500106: OVERLAY

X0 File reference number of source file

X4 Flags for operation

X6 Trap location

X7 Pointer to registers

The source file specified by X0, which must be open with eXecute permission, is copied over the issuing job's memory. The job's registers are loaded from the area specified by X7, which must begin at a multiple of eight words.

If bit 0 of X4 is 1 at the time of the overlay, the source file will be closed when the overlay is complete. If bit 0 of X4 is 0, the source file will remain open and unchanged.

If all parameters are correct and there are no errors during the copy operation, the job is restarted at location zero with the registers pointed to by X7, and with the zero indicator on.

If an error is detected before the copy operation begins, the issuing job is trapped to the trap location specified by X6 with one of the statuses listed below.

If an error occurs after initiation of the copy, the issuing job is aborted and the supra job's RUN or CONTINUE command is trapped with a status of 400 (recoverable I/O error).

Status Returns

1 Overlay rejected: The overlay has been rejected because the job has another trapping Executive command outstanding.

2 Overlay rejected: file is bigger than current allocated memory.

100 Access error: The file specified by X0 is not open with eXecute permission.

200 X0 parameter error: X0 is zero, does not contain a valid file reference number, or contains the file reference number of a device, job, or communication file.

300 X4 parameter error: X4 contains illegal flag bits.

360 X7 parameter error: The pointer in X7 is out of bounds or is not a multiple of 8.

MME 500122: PASS

X0 File reference number of job file or zero

X1 Points to a location which points to the starting location M1 of message

X2 File reference number of file/catalog to pass or zero

X4 Access to pass with file/catalog

X5 Preference (for created files)

X6 Trap location

X7 Points to the length N of the message or is zero

The file/catalog is passed down to the job specified in X0, generating a special interrupt number 4 (passed file) if that job is accepting special interrupts. This special interrupt gives the new file reference number of the file/catalog in the lower half of the first interrupt word and the length of the file or MAX of the catalog in the second interrupt word. The new file reference number is not necessarily equal to the one for the job passing the file/catalog.

If X7 is nonzero and points to a nonzero word, then the message specified can be read by the job to which the file/catalog is passed. This is accomplished by issuing a REQUEST STATUS as the first file operation on that file/catalog. If X7 is zero or points to a zero word, then no message accompanies the file/catalog on the pass. A trap occurs to the location specified in X6 when the message, if any, is read or when any other file operation is initiated on the passed file. Hence, the message should not be altered until the trap occurs.

If X2 is zero then the Executive will create and pass a scratch file, or create a communication file and pass its slave end, depending on whether bit 1 of the access word specified in X4 is 0 or 1. The file reference number of the created file is returned in the lower half of status word 1 in the trap for the PASS (unless the file created was a scratch file and the return bit was not set in the access in X4). No message can be passed with a communication file.

If X0 is zero, then the file/catalog will be passed upward to the immediate supra job. Currently, this feature is implemented for created communication files only.

Job files, master ends of communication files, and file/catalogs which have already been passed cannot be passed to an infra job. The file reference number of a file/catalog which is passed without the return bit (bit 2) set in its access word becomes invalid. With the exception of slave ends of communication files, file/catalogs which are passed with the return bit set become busy until they are closed and returned by the infra job. A file/catalog which has been passed to the job issuing the PASS with the return bit set must be passed to an infra job with the return bit set. A scratch file/catalog which is passed without the return bit set automatically has all accesses set for the infra job. The current setting of the read/write pointer for a file/catalog is preserved when it is passed.

Status Returns

0 Successful: the message, if any, was read.

2 Successful: a message existed but was not read.

60 System out of storage: the Executive's table of opened files is full.

120 Busy: another operation is outstanding on the file/catalog to be passed.

200 X0 parameter error: X0 does not contain the file reference number of a nonterminated job file.

220 X1 parameter error: X1 points out of bounds.

240 X2 parameter error: either

(1) X2 does not contain a valid file reference number;

(2) It contains the file reference number of a passed file/catalog or of the master end of a communication file; or

(3) It is zero and the job access mask for the issuing job does not permit it to specify a preference for creating a scratch file.

300 X4 parameter error: either

(1) Extraneous bits are set in the access word;

(2) The return bit is not set in the access word though the file/catalog was passed to the issuing job with the return bit set;

(3) Bit 0, the catalog bit, is 0 (is 1) and the file/catalog being passed is a catalog (is a file);

(4) No accesses are present in bits 5-8;

(5) The catalog bit is on and X2 is zero;

(6) The return bit is set and so is the bit to indicate the creation of a communication file; or

(7) X0 is zero and X4 does not specify the creation of a communication file

360 X7 parameter error: X7 points out of bounds.

MME 500126: PROVIDE DEVICE ADDRESSES

X0 File reference number of file

X3 Points to pointer to memory location M2

X6 Trap location

X7 Points to length N of buffer

The PROVIDE DEVICE ADDRESSES command is a privileged command which can be issued by jobs enabled for the Load-Dump permission. Nonenabled jobs which attempt to issue it will receive a status of 440 fault. The command causes the device address list for the specified file to be copied into the buffer of length N starting at memory location M2; no device addresses will be copied if the whole list does not fit into this buffer. Upon completion of the command, a trap occurs to the location specified in X6. If the command was successful, status word 2 contains the number of device addresses for the file.

This command cannot be used to provide device addresses for a catalog. Enabled jobs desiring the device addresses for a catalog can obtain them by issuing a READ CATALOG AND OPEN FILES command.

Status Returns

0 Successful.

120 Busy: another operation is outstanding on the file.

200 X0 parameter error: X0 does not contain the file reference number of a scratch or cataloged file.

260 X3 parameter error: X3 points out of bounds or points to an out-of-bounds pointer.

360 X7 parameter error: either X7 points out of bounds or the list of device addresses for the file will not fit into the specified buffer.

440 Not enabled: the job is not enabled with Load-Dump permission.

MME 500133: READ

X0 File reference number of source file

X3 Points to pointer to memory location M2

X4 Flag bits

X6 Trap location

X7 Points to number of words N to read

N words are read from the source file starting at the current position of that file's read/write pointer into the issuing job's memory starting at location M2. Read permission is required on the source file. If the source file is exhausted or if the end of the issuing job's storage is reached, an end-of-file condition occurs and transmission of data stops. Upon completion of the copy, a trap occurs to the location specified in X6.

Zero is an illegal parameter for X0 or X3. Upon completion of the copy, the read/write pointer for the source file is updated to point to the word beyond the last one read.

A READ issued at the slave end of a communication file will generate a special interrupt number 11 (slave end issued READ) at the master end. The second interrupt word will contain the length of the READ. No data will be transferred until the master end issues a corresponding WRITE or COPY. (See the description of communication files in Chapter 5.) If the communication file is busy, either because another end has an operation outstanding on it, or because another end has reserved it, the slave end issuing the READ command will be trapped with a status of 6. If the job at the master end of a communication file is not accepting special interrupts when a slave end issues a READ, then that slave end will be trapped with a status of 7.

The flag bits in X4 have the following meaning:

Bit	Meaning
0	If the source file is a master communication file, the corresponding slave end is trapped only if a nonzero status return occurs or if this bit is zero. This bit must be zero if the source file is not a master communication file.
1	If the source file is a slave communication or a shared file, that file is reserved for the end issuing the READ command if this bit is one. Any prior reservation by this end is released if this bit is zero. This bit must be zero if the source file is not a slave communication file or a shared file.
2-17	These bits are reserved for future use. They must be zero.

If a status return in status word 1 is not between 100 and 360 on completion of the READ, then status word 2 contains the difference between the number of words transferred and the number of words requested (i.e. it contains minus the number of words not transferred). If the status in word 1 is between 100 and 360, then status word 2 is zero. If the source file is a device file, then the lower half of status word 1 contains the status return from that device.

The observant reader will note that the READ command is treated as a special case of the COPY command in which X1 and X2 are assumed to contain zeros.

Status Returns

0 Successful: all words were transferred.

1 Source file exhausted: more words were requested but could not be transferred.

2 Destination file exhausted: some words in the source file were not transferred since the end of the issuing job's storage was reached.

3 Command inappropriate: the READ was issued at the master end of a communication file without a corresponding WRITE operation outstanding at a slave end of that file.

4 The source file pointer is out of bounds: either,

(1) The pointer was initially out of bounds;

(2) The other end of a communication file issued a MEMORY REQUEST which released job storage involved in the read; or

(3) An attempt was made to read from the pure region of a job file which was swapped out of main memory.

5 The destination file pointer is out of bounds: either,

(1) The pointer was initially out of bounds;

(2) A MEMORY REQUEST released job storage involved in the read; or

(3) An attempt was made to read into the pure region of a job file which was swapped out of main memory.

6 The communication file at whose slave end this command was issued is busy, either because another end has an operation outstanding, or because another end has reserved the communication file.

7 The master end of the communication file at whose slave end this command was issued is not accepting special interrupts.

20 Status was reset on the READ.

100 Access error: the source file is not open with Read permission. Bits 0-8 of status word 1 contain the missing permission bit.

120 Busy: another operation is outstanding on the source file.

200 X0 parameter error: X0 does not contain the file reference number of a file.

260 X3 parameter error: X3 is zero or points to an out-of-bounds location.

300 X4 parameter error: X4 contains extraneous or invalid flag bits.

360 X7 parameter errors: X7 points out of bounds.

400 Recoverable error: an error occurred during transmission of data. The lower half of status word 1 contains the status from the device which generated the error. This return can occur if a device file is being read via a communications file and the issuing job is swapped out of main memory.

MME 500114: READ CATALOG

X0 File reference number of catalog

X1 Points to number M1 of first entry in the catalog to copy

X3 Points to pointer to memory location M2

X6 Trap location

X7 Points to number of words N to copy

DTSS

N words of formatted catalog information, starting with the information for the first catalog entry whose number is greater than or equal to M1, are written into the issuing job's storage starting at location M2. Read permission is required on the catalog. If the formatted catalog information is exhausted or if the end of the issuing job's storage is reached, an end-of-file condition occurs and data transmission stops. Upon completion of the copy, a trap occurs to the location specified in X6.

If X1 is zero then the current value of the catalog's read/write pointer is used to specify an entry number. Upon completion of the copy, this pointer is updated to point to the entry past the last one for which information was transferred. Note that since holes can occur in catalogs and since these holes have entry numbers, the value of the catalog's read/write pointer upon completion of a copy may not be equal to the number of entries for which information was transferred. X3 may not be zero.

Information for each entry in the catalog is formatted into an eight-word block. The block for entry number zero contains data pertaining to the catalog itself; blocks for other entry numbers contain information pertaining to file/catalogs cataloged in the catalog.

The format of the information for entry number zero which pertains to the catalog itself is as follows:

Word	Function
0	MAX
1	ALOC (amount of storage used by entries)
2	Zero (unused)
3	Zero (unused)
4	Upper: access with which catalog is open Lower: preference and type of catalog
5	Upper: zero (unused) Lower: number of entries, including holes, in catalog
6	Zero (unused)
7	Length of the catalog

The format of the information pertaining to entries in the catalog is as follows:

Word	Function
0-1	Name
2-3	Password or name of trap program (zero if catalog is not open with Owner permission)
4	Catalog access word
5	Upper: days-used count Lower: preference and type of file/catalog
6	Upper: coded date of last use Lower: coded date of last modification
7	Length of file or MAX of catalog

See Chapter 9 for a more detailed explanation of these fields.

If the status return in status word 1 is not between 100 and 360 on completion of the copy, then status word 2 contains the difference between the number of words transferred and the number of words requested (i.e. it contains minus the number of words not transferred). If the status in word 1 is between 100 and 360, then status word 2 is zero.

Status Returns

0 Successful: all words were transferred.

1 Source file exhausted: more words were requested than were contained in the formatted catalog information.

2 Destination file exhausted: some words of formatted catalog information were not transferred since the end of the issuing job's storage was reached.

4 Source file pointer out of bounds: there are no entries in the catalog with an entry number greater than or equal to M1.

5 The destination file pointer is out of bounds: either,

   (1) The pointer was initially out of bounds;

   (2) A MEMORY REQUEST released job storage involved in the copy; or

   (3) An attempt was made to copy into the pure region of a job file which was swapped out of main memory.

20 Status was reset.

100 Access error: the catalog is not open with Read permission. Bits 0-8 of status word 1 contain the missing permission bit.

120 Busy: another operation is outstanding on the catalog.

200 X0 parameter error: X0 is zero or does not contain the file reference number of a catalog.

220 X1 parameter error: X1 points to an out-of-bounds location.

260 X3 parameter error: X3 points to an out-of-bounds location or is zero.

360 X7 parameter error: X7 points to an out-of-bounds location or is zero.

400 Recoverable error: an error occurred in reading the catalog.

420 Unrecoverable error: information in the catalog has been destroyed.

MME 500127: READ CATALOG AND OPEN FILES

X0 File reference number of catalog

X1 Points to number M1 of first entry in the catalog to copy

X2 Reserved for future use (must be zero)

X3 Points to pointer to memory location M2

X4 Access for OPEN (bits 0-8); Trap bit mask (bits 9-17)

X5 Points to one word of coded dates (DLU/DLM)

X6 Trap location

X7 Points to number of words N to copy

AQ Slave trap identification words

The READ CATALOG AND OPEN FILES command combines the functions of the READ CATALOG and the OPEN commands by simultaneously providing information about entries in a given catalog and by opening those entries. It is intended primarily for use in privileged system modules and serves to minimize the number of disk accesses required for file system maintenance.

When a READ CATALOG AND OPEN FILES command is issued, N words of formatted catalog information are written into the issuing job's storage starting at location M2. This operation starts with the information for the first entry in the catalog whose number is greater than or equal to M1. Formatted information is provided for all catalogs in the catalog being read. Information for migrated files is provided only if the date last used is greater than or equal to the coded DLM in the lower half of the word pointed to by X5. Information for other files is provided only if the file's date last used is less than or equal to the coded DLU in the upper half of the word pointed to by X5, or if the file's date last modified is greater than or equal to the coded DLM in the lower half of the word pointed to by X5. (Therefore, to get information on all files, the word should contain 0 in the lower half. To get information on files using only the DLU, the word should contain the desired DLU in the upper half and 777777 octal in the lower half. To get information on files using only the DLM, then the word should contain 0 in the upper half and the desired DLM in the lower half.)

In addition, an attempt is made to open without a password each file/catalog for which information is transferred. If the information is exhausted or if the end of the issuing job's storage is reached, an end-of-file condition occurs and data transmission stops. Upon completion of the command, a trap occurs to the location specified in X6. The date last used and date last modified in the catalog entry for a file/catalog opened by the READ CATALOG AND OPEN FILES command will not be updated.

Read permission is required on the catalog; search permission is also required if any file/catalogs are to be opened. Zero is an invalid parameter for X0 or X3. If X1 is zero, then the current value of the catalog's read/write pointer is used to specify an entry number. Upon completion of the command, this pointer is updated to point to the entry past the last one for which information was transferred. Note that since holes can occur in catalogs, and since these holes have entry numbers, the value of the catalog's read/write pointer upon completion of the command may not be equal to the number of entries for which information was transferred.

Information for each entry in the catalog other than entry zero is formatted into a ten-word block which contains information pertaining to that file/catalog. The information for entry zero pertains to the catalog itself and is formatted into a twenty-word block.

The format for entry zero (whose information pertains to the catalog itself) is as follows:

Word	Function
0	MAX
1	ALOC (amount of storage used by entries)
2	Upper: zero (unused)

	Lower: number of fragments in which storage for catalog is allocated
3	Zero (unused)
4	Upper: access with which catalog is open Lower: preference and type of catalog
5	Upper: zero (unused) Lower: number of entries, including holes, in catalog
6	Zero (unused)
7	Length of catalog
8-19	List of device addresses for catalog (zero if issuing job is not enabled with the Load-Dump permission)

The format of the information pertaining to entries in the catalog is as follows:

Word	Meaning
0-1	Name
2-3	Password or name of trap program (zero if catalog is not open with Owner permission)
4	Catalog access word
5	Upper: days-used count Lower: preference and type of file/catalog
6	Upper: coded date of last use Lower: coded date of last modification
7	Length of file or MAX of catalog
8	Upper: status return from OPEN of file/catalog Lower: file reference number of file/catalog if opened

9 Number of fragments in which storage for file/catalog is allocated (zero if file/catalog not opened)

The success of the open of each subfile/catalog is determined in the following manner:

(1) If the file/catalog is master or slave trapped, then the access bits are as allowed by the trapping program;

(2) If the catalog specified in X0 is open with Owner permission, then all access bits are allowed (unless the file/catalog is trapped);

(3) If the file/catalog is not trapped and the catalog being read is not open with Owner permission, then the accesses allowed are those in bits 3-8 of the catalog access word;

(4) The allowed accesses are checked for usage conflicts and masked off if any conflicts occur;

(5) Unless the job issuing the command is Load-Dump enabled,

(a) Append permission is masked off if the catalog's MAX has been exceeded,

(b) All access bits are masked off if the allocated storage for the catalog exceeds twice its permissible MAX;

(6) Any remaining bits are ANDed with the requested accesses, and bit 0 is set to 1 if the file/catalog being opened is a catalog.

If any accesses remain in bits 5-8, then those accesseses are returned in bits 0-8 of word 8 of the entry information block. In addition, the nine user access bits from 9-17 of the catalog access word are ORed into bits 9-17 of the access word in the file/catalog's file control block. The status return from the open which occurs in word 8 of the entry information block is one of the following:

Status Returns

0 Successful: the accesses in bits 0-8 were obtained.

1 Partial success: the accesses in bits 0-8 were obtained but differ from those requested.

2 Lockout: all allowable accesses are currently busy.

5 Unsuccessful: no accesses were allowed. Bit 0 will be 1 if the file/catalog is a catalog.

10 File off line: the upper half of status word 2 contains the purged date, and the lower half-word contains the reel number, preference, and TYPE.

40 Quotas exceeded: either

(1) The job has exceeded its allotted state vector length, or

(2) The job is not enabled for Load-Dump permission and the allocated storage for the catalog is over twice the permissible MAX.

60 Out of storage: the Executive's table of opened files is full.

100 Access error: the catalog is not open with Search permission; bits 0-8 contain the missing permission bit.

On completion of the command, if the status return in status word 1 is not between 100 and 360, then status word 2 contains the difference between the number of words transferred and the number of words requested (i.e. it contains minus the number of words not transferred). If the status in word 1 is between 100 and 360, then status word 2 is zero.

Status Returns

0 Successful: all words were transferred and all available file/catalogs were opened.

1 Source file exhausted: more words were requested than were contained in the formatted catalog information.

2 Destination file exhausted: some words of formatted catalog information were not transferred since the end of the issuing job's storage was reached.

4 Source file pointer out of bounds: there are no entries in the catalog with an entry number greater than or equal to M1.

5 The destination file pointer is out of bounds: either,

(1) The pointer was initially out of bounds;

(2) A MEMORY REQUEST released job storage involved in the copy; or

(3) An attempt was made to copy into the pure region of a job file which was swapped out of main memory.

14 The current length of the issuing job's state vector will not accommodate another file control block.

20 Status was reset.

100 Access error: the catalog is not open with Read permission; bits 0-8 of status word 1 contain the missing permission bit.

120 Busy: another operation is outstanding on the catalog.

200 X0 parameter error: X0 is zero or does not contain the file reference number of a catalog.

220 X1 parameter error: X1 points out of bounds.

240 X2 parameter error: X2 was not zero.

260 X3 parameter error: X3 points out of bounds or is zero.

320 X5 parameter error: X5 points out of bounds.

360 X7 parameter error: X7 points out of bounds.

400 Recoverable error: an error occurred in reading the catalog.

420 Unrecoverable error: information in the catalog has been destroyed.

MME 500116: REPLACE

X0 File reference number of initial catalog to search

X1 Points to name of file/catalog to replace

X2 File reference number of scratch file/catalog

X3 Points to password (or is zero)

X4 (bits 0-8) must be zero; (bits 9-17) trap bit mask

X5 Reserved for future use (must be zero)

X6 Trap location

X7 Number of entries in treename (if zero, then 1 is assumed)

AQ Slave trap identification words

The REPLACE command attempts to open the named file/catalog with all permissions and to replace it with the specified scratch file/catalog.

In this respect it is roughly equivalent to the sequence of commands ERASE, CATALOG; however, the REPLACE command has a number of advantages over this combination of commands. It minimizes the number of catalog accesses required to perform the combined function, and it permits the replacement of a file/catalog with a shorter version in a catalog whose MAX has been exceeded even though the MAX may still be exceeded after the replacement.

The replacement will not be allowed if another job has the file/catalog open with Write permission; however Append permission may be locked out at the time of the REPLACE.

The replacement will be made if Read, Write, and Append permissions are present in the available accesses, unless the allocated storage for the catalog exceeds its permissible MAX. In this latter case the replacement will be made only if the length of the scratch file or MAX of the scratch catalog is less than or equal to the length or MAX of the file/catalog which it replaces. If after performing such a replacement, the catalog's MAX is still exceeded, then Append permission will be removed from the accesses available.

If the replacement is successful, then the file/catalog specified in X2 becomes a cataloged file/catalog open with the accesses which were determined to be available, and the catalog entry is updated correspondingly. In addition to the available accesses which are returned in status word 1 of the trap, nine user access bits can be obtained by issuing a REQUEST STATUS. These user access bits are obtained from bits 27-35 of the searched catalog's access word if the file/catalog was opened with a password, and from bits 9-17 of this access word if the file/catalog was opened without a password or if it was protected by a trap program. If the replacement was unsuccessful, then the file/catalog remains a scratch file/catalog and the catalog entry remains unchanged.

A job which attempts to replace a file/catalog first has its catalog word allotment decremented by the length of the scratch file or of the scratch catalog header. If the replacement succeeds then his scratch quotas are incremented by the same amount. If the replacement does not succeed, then this allotment is incremented by the amount of the original debit.

A file may not be replaced by a catalog, nor may a catalog be replaced by a file. Replacement of a cataloged file/catalog does not affect the name, password, or access word in the catalog entry. The replacement scratch file/catalog must not have been passed to the issuing job with the return bit set.

Upon completion of the command, a trap occurs to the location specified in X6. If the replacement was successful, bits 0-8 of status word 1 contain the accesses with which the file/catalog is now open. Status word 2 contains the length of the file or MAX of the catalog.

Status Returns

0 Successful: the file/catalog was replaced and is now open with the accesses in bits 0-8.

1 Partial success: the file/catalog was replaced and is now open with the accesses in bits 0-8. Append permission has been masked off since the searched catalog's MAX is exceeded.

2 Lockout: the cataloged file/catalog is currently in use with Write permission.

3 Not found: the desired file/catalog was not found in the searched catalog.

4 Protection violation: the specified password was wrong.

5 Unsuccessful: one of the Read, Write, or Append permissions was not allowed.

6 Bad treename: Search permission was available on some intermediate catalog and the next entry in the treename was not the last entry, but was the name of a file rather than a catalog.

7 Fetch error: either

(1) An error of type 3, 4, 5, or 6 occurred and Search permission was not available on the catalog in which the search was being performed; or

(2) Search permission was not available on the (N-1)st catalog and fetch permission was not available on the file/catalog.

11 REPLACE disallowed: the named file/catalog is a device file.

40 Quotas exceeded: either

(1) The length of the scratch file or catalog header exceeds the job's remaining catalog word allotment, or

(2) The specified catalog's MAX is exceeded, and the length of the scratch file or MAX of the scratch catalog is greater than the length or MAX of the file/catalog to be replaced.

60 Out of storage: either

(1) The Executive's table of opened files is full, or

(2) A new entry had to be made in the catalog for the replacement and either

(a) The catalog already had 4095 entries,

(b) The Executive could not allocate sufficient storage for a required extension of the catalog, or

(c) It is not possible to append to the catalog since it either is a main-memory catalog or is already allocated in twelve fragments.

120 Busy: another operation is outstanding on either the catalog or the scratch file/catalog.

200 X0 parameter error: X0 does not contain the file reference number of a catalog.

220 X1 parameter error: X1 points out of bounds.

240 X2 parameter error: either

(1) X2 does not contain the file reference number of a scratch file/catalog,

(2) The scratch file/catalog was passed to the issuing job with the return bit set,

(3) The type of the scratch file/catalog does not match the type of the file/catalog it is to replace, or

(4) The scratch file/catalog has a preference of 0, 1, 2, or 4 and may only replace a file in a core catalog.

260 X3 parameter error: the pointer to the password is out of bounds.

300 X4 parameter error: bits 0-8 of X4 were not zero.

320 X5 parameter error: X5 was not zero.

360 X7 parameter error: X7 was greater than 10 (maximum of 10 names in the list of entries pointed to by X1).

400 Recoverable error: an error occurred in reading the catalog.

420 Unrecoverable error: information in the catalog has been destroyed.

MME 500115: REQUEST STATUS

X0 File reference number of file/catalog or zero

X3 Points to pointer to memory location M2

X6 Trap location

X7 Points to number of words N to read

N words of status information for the file/catalog are read into the issuing job's memory starting at location M2. Upon completion of the command, a trap occurs to the location specified in X6.

DTSS

If X0 is zero, then the status information pertains to the job issuing the command and has the following format:

Word	Meaning
0	Job's remaining catalog word allotment
1	Job's remaining scratch word allotment
2	CRU limit for job (in 1/64 milliCRU) — not CRUs remaining
3	Number of I/O units used
4-5	Identifying words from run list of the RUN command which spawned this job
6-7	Identifying words from run list of the RUN command which spawned the job which spawned this one
...	Etc., for each job above this one in the job tree

If X0 contains the file reference number of a suspended or terminated job file, then the status information for the job file specified has the following format:

Word	Meaning
0	Accesses/Type (see below)
1	Job length (core size) [will be zero for a terminated job file]
2	Computer resource units (CRUs) consumed
3	I/O units consumed
4	Central processor units consumed
5	Core units consumed

Note: The quantities in words 2 through 5 include resources consumed by all terminated infra jobs.

If X0 is nonzero and is not the file reference number of a job file or the slave end of a communication file, then the status information for the file/catalog specified has the following format:

Word	Meaning
0	bits 0-17 (upper): accesses with which the file/catalog is open bits 18-26: zero bits 27-28: preference adjustment bits 29-31: preference (see Chapter 9 for preferences) bits 32-35: type (see Chapter 9 for types)
1	Length of file or MAX of catalog
2	Read/write pointer
3	Upper: Maximum accesses available on the file through an ALTER ACCESSES command (q.v.). Lower: Unique Identifier. This field can be used by a job to determine whether two FRNs actually access the same file. The identifier will be the same for the two FRNs if and only if they access

the same file. The value of this field is undefined for other than cataloged files.

4-N Message specified by PASS command

The message starting in word 4 is available only if the file/catalog was passed to the issuing job by a PASS command and if the REQUEST STATUS is the first file operation issued on the file/catalog. (See PASS.)

If X0 contains the file reference number of the slave end of a communication file, then a special interrupt number 13 (slave end issued REQUEST STATUS) is generated at the master end. The second interrupt word contains the number of words requested. No status information for the file is supplied by the Executive. Instead, it is the duty of the master end of the communication file to write status information to the slave end via a WRITE or COPY command. In other words, a REQUEST STATUS issued at the slave end of a communication file behaves in the same manner as a READ command, except that the communication file can never be reserved by a REQUEST STATUS command. If the communication file is busy, either because another end has an operation outstanding on it, or because another end has reserved it, the slave end issuing the REQUEST STATUS command will be trapped with a status of 6. If the job at the master end of a communication file is not accepting special interrupts when a slave end issues a REQUEST STATUS, then that slave end will be trapped with a status of 7. If the communication file is reserved for this end, it will be released.

If the status return in status word 1 is not between 100 and 360 on completion of the command, then status word 2 contains the difference between the number of words transferred and the number of words requested (i.e. it contains minus the number of words not transferred). If the status in word 1 is between 100 and 360, then status word 2 is zero.

Status Returns

0 Successful: all words of status information were transferred.

1 Source file exhausted: more words of status information were requested than existed.

2 Destination file exhausted: some words of status information were not transferred since the end of the issuing job's storage was reached.

4 Source file pointer out of bounds: either,

(1) This file/catalog has been passed with a message and the supra job has released the job storage containing the message; or

(2) This is the slave end of a communication file and the master end is attempting to supply status information from a source file whose pointer is out of bounds.

5 The destination file pointer is out of bounds: either,

(1) The pointer was initially out of bounds;

(2) A MEMORY REQUEST released job storage involved in the request; or

(3) The issuing job was swapped out of main memory while the master end of a communication file was writing into the job's pure region.

6 Communication file busy: the communication file at whose slave end this command was issued is busy because another end has an operation outstanding on it, or has reserved it.

7 The master end of the communication file at whose slave end this command was issued is not accepting special interrupts.

20 Status was reset.

120 Busy: another operation is outstanding on the file/catalog.

200 X0 parameter error: X0 does not contain a valid file reference number or contains the file reference number of a former communication file whose other end has been closed.

260 X3 parameter error: X3 is zero or points out of bounds.

360 X7 parameter error: X7 points out of bounds or points to a length greater than 2^18.

400 Recoverable error: an error occurred during the transmission of the message on a passed file/catalog or of the status information on the slave end of a communication file. The lower half of status word 1 contains the status from the device which generated the error.

MME 500135: RESET STATUS

X0 File reference number of file/catalog

X4 Flag bits

X6 Trap location

The file/catalog is forced into an idle state. Any file operation outstanding on the file/catalog is halted. The command which initiated the operation is trapped, generally with a status return of 20 (status was reset). Status cannot be reset on a file/catalog which has been passed to an infra job and which has not yet been returned. A file reference number of zero is invalid. Upon completion of the command, a trap occurs to the location specified in X6.

A RESET STATUS issued on a running job will suspend execution of that job and all jobs running below it until a CONTINUE command is issued.

A RESET STATUS issued on a communication file for which an operation is outstanding at both the master and a slave end will cause status to be reset at both ends. A RESET STATUS issued on the slave end of a communication file when no operation is outstanding at the master end of that file will generate a special interrupt number two (slave end issued RESET STATUS) at the master end and will cause the master end to be busy until a RESET STATUS is issued on it. A RESET STATUS issued at the slave end of a communication file will cause the file to be released (if it was reserved for that end).

The flag bits in X4 have the following meanings:

Bit	Function
0	Unassigned, must be zero.

1	Must be zero unless X0 contains the file reference number of the master end of a communication file. In that case, the file will be reserved for the master end if this bit is one (see Section 5.4). If this bit is zero, any prior reservation by the master end will be released.
2-17	Unassigned, must be zero.

A RESET STATUS issued on the source file for pure procedure will unbusy that file and declare the job to be impure; i.e. it has the same effect as a PURE command in which X7 is zero.

A RESET STATUS on a file opened in shared mode will cause the file to revert to an idle, unreserved (unlocked) state at the issuing end.

Status Returns

0	Successful: status was reset.
1	A RESET STATUS is already in progress on this file/ catalog.
120	Busy: the file/catalog has been passed to an infra job.
200	X0 parameter error: X0 does not contain a valid file reference number.
300	X4 parameter error: X4 contains invalid flag bits.

MME 500120: RUN

The RUN MME is very similar to the EXECUTE MME (500117) and differs only in the following:

(1) Upon initiation of the job, the source file is closed and its file reference number is assigned to the newly created job file. This remains open with all accesses possessed by the source file except Append permission until the job file is closed.

(2) The file reference number of the job file will not be placed in the lower half of status word 1 of the trap block specified by X6.

MME 500110: SCRATCH

X0 File reference number of file

X6 Trap location

The specified file, which must be open with Write and Append permissions, is erased and both its length and its read/write pointer are set to zero. Upon completion of the command, a trap occurs to the location specified in X6.

DTSS

A SCRATCH command issued at the slave end of a communication file will generate a special interrupt number 14 (slave end issued TRUNCATE) at the master end. The second word of the special interrupt will be zero (see TRUNCATE). If the communication file is busy, either because another end has an operation outstanding on it, or because another end has reserved it, the slave end issuing the SCRATCH command will be trapped with a status of 6. If the job at the master end of a communication file is not accepting special interrupts when a slave end issues a SCRATCH, then that slave end will be trapped with a status of 7. A SCRATCH command cannot be issued on the master end of a communication file.

A job which scratches a scratch (cataloged) file will have its scratch (catalog) word allotment incremented by the length of the file.

The observant reader will notice that the SCRATCH command is treated as a special case of the TRUNCATE command in which the A-register is assumed to contain a zero.

Status Returns

0 Successful: the file was scratched.

6 Communication file busy: the communication file at whose slave end this command was issued is busy because

another end has an operation outstanding on it or has reserved it.

7 The master end of the communication file at whose slave end this command was issued is not accepting special interrupts.

20 Status was reset on the TRUNCATE.

100 Access error: the file is not open with Append and Write permissions; bits 0-8 of status word 1 contain the missing access bits.

120 Busy: another command is outstanding on the file to be scratched.

200 X0 parameter error: X0 is zero and does not contain a valid file reference number, or contains the file reference number of a device, job, or master communication file.

MME 500113: SET POINTER

X0 File reference number of file/catalog

X6 Trap location

A Setting for read/write pointer

The read/write pointer of the file/catalog is set to the specified value, and a trap occurs to the location specified in X6. Zero is an invalid file reference number.

A SET POINTER command issued at the slave end of a communication file will generate a special interrupt number 15 (slave end issued SET POINTER) at the master end of the file. The second interrupt word will contain the specified setting for the read/write pointer. If the communication file is busy, either because another end has an operation outstanding on it, or because another end has reserved it, the slave end issuing the SET POINTER command will be trapped with a status of 6. If the job at the master end of a communication file is not accepting special interrupts when a slave end issues a SET POINTER, then that slave end will be trapped with a status of 7. A SET POINTER command may not be issued on the master end of a communication file.

Status Returns

0 Successful.

1 Pointer out of bounds. Register A specifies a pointer greater than the length of the file, or the file is a catalog or device file and register A is not less than 2^18.

6 Communication file busy: the communication file at whose slave end this command was issued is busy because another end has an operation outstanding on it or has reserved it.

7 The master end of the communication file at whose slave end this command was issued is not accepting special interrupts.

20 Status was reset on the SET POINTER.

120 Busy: another operation is outstanding on the file/ catalog.

200 X0 parameter error: X0 is zero or does not contain the file reference number of a file/catalog.

MME 500141: TALLY CATALOG

X0 File reference number of initial catalog to search

X1 Pointer to a tally word

X2 File reference number of file to be cataloged

X3 File reference number of alternate initial catalog

X4 Pointer to 2 words of permission and access information

X5 Pointer to two words for name of file/catalog cataloged by the Executive

X6 Pointer to trap block

X7 Pointer to usage and dates (Load-Dump enabled jobs only) or zero

The specified scratch file/catalog is cataloged in the file system in a catalog denoted by a standard treename.

First an attempt to open the specified file is made using TALLY OPEN (MME 500136). The trap bit mask for the search is in bits 9-17 of the first word pointed to by X4. If bit 4 (020000 [octal], the escape convention bit) is set in the upper half of this first word, then all special scanning conventions are disallowed, and the catalog will be trapped with a format error if the treename contains a special first name. The name and password of the file/catalog to be cataloged cannot be specified with the "/" convention, and the catalog will be trapped with a format error if that is attempted. For more information on tally operations, see Chapter 8.

If the Executive receives a file-not-found status for the last file/catalog then that file/catalog is cataloged if Append permission is available on the destination catalog. For example, if the treename is A:B:C, then C is cataloged in A:B if and only if A:B exists, Append permission is available on B, and A:B:C does not already exist. The second word pointed to by X4 is the access word with which the file will be saved. If the treename includes a password for the last file/catalog, then this password will be put on the cataloged file/catalog.

If the job is enabled with Load-Dump permission and X7 is non-zero, then it must point to two words of usage and dates information. The upper half of the first word contains an integer used to set the file's days-used count; the lower half contains type information or is zero. The second word contains a coded Date Last Used in the upper half and coded Date Last Modified in the lower half. See MME CATALOG or Chapter 9 for more details on usage and dates information.

Upon completion of the operation, the job's scratch word allotment is incremented by the length of the file or of the catalog header, and its catalog word allotment is decremented by this quantity plus the length of the entry created in the catalog. Finally a trap occurs to the location specified by X6.

If the command was unsuccessful, then status word 1 contains the reason for the failure. If the status is 12 or 40 or greater, the two words pointed to by X5 are undefined. If the status was 7 (fetch error) then the two words are cleared. Otherwise, the last file name scanned is placed in the two words. (These words need not be aligned on a even-word boundary.) If the status returned from the catalog is 12 (format error), then status word 2 will contain a substatus, and a tally count and character position pointing to the character where the format error was detected. See Chapter 8 for more information on format errors.

Status Returns

0 Successful: the file/catalog was cataloged.

1 Illegal trap protection request: a master trap protection bit which is not allowed to the issuing job has been specified.

2 Duplicate name: an entry with the given name is already in the catalog.

3 Not found: Search permission was available on some intermediate catalog and the next file/catalog was not found in it.

4 Protection violation: Search permission was available on some catalog, and the password specified for the next file/catalog was wrong.

6 Bad treename: Search permission was available on some intermediate catalog, and the next entry in the treename was not the last entry, but was the name of a file rather than a catalog.

7 Fetch error: an error of type 2, 3, 4, or 6 occurred and Search permission was not available on the catalog in which the search was being performed.

12 Format error: see Chapter 8 for possible types of errors.

13 Illegal usage and dates: the usage information pointed to by X7 was illegal; either the date last used was zero, or the preference or type information was wrong.

14 Preference too low: the specified file has a preference of 0, 1, 2, or 4 and may be cataloged only in a core catalog.

40 Quotas exceeded: either

(1) The job has exceeded its allotted state vector length; or

(2) Allocated storage for the next to last catalog is over twice the permissible MAX, and files must be erased from this catalog before any can be opened.

60 Out of storage: the Executive's table of opened files is full.

100 Access error: The initial catalog to search is not open with Search permission; or the last catalog before the file/catalog to be cataloged lacks Append permission. Bits 0-8 of status word 1 contain the missing permissions.

120 Busy: another command is outstanding on the catalog specified in X0 or X3.

200 X0 parameter error: X0 does not contain the file reference number of a catalog.

220 X1 parameter error: the pointer to the tally word is out of bounds.

240 X2 parameter error: X2 does not contain the file reference number of a scratch file/catalog.

260 X3 parameter error: X3 does not contain the file reference number of a catalog.

300 X4 parameter error: the two-word pair pointed to by X4 was out of bounds or an illegal bit was set.

320 X5 parameter error: X5 pointed to a two-word pair which was out of bounds.

360 X7 parameter error: the two-word pair pointed to by X7 was out of bounds, or the job was not Load-Dump enabled and X7 was nonzero.

400 Recoverable error: an error occurred in reading an intermediate catalog.

420 Unrecoverable error: information in an intermediate catalog has been destroyed.

MME 500137: TALLY ERASE

X0 File reference number of initial catalog to search

X1 Pointer to a tally word

X2 Reserved for future use (must be zero)

X3 File reference number of alternate initial catalog

X4 (bits 0-3,5-8) must be zero, (bit 4) escape special convention, (bits 9-17) trap bit mask

X5 Pointer to two words for name of file/catalog erased by the Executive

X6 Pointer to trap block

X7 Reserved for future use (must be zero)

AQ Passed to slave trap program if file is slave trapped

The TALLY ERASE command attempts to open the specified file/catalog with Read, Write, and Append permissions using TALLY OPEN (MME 500136). If the open is successful then the Executive destroys the file/catalog together with its catalog entry.

If bit 4 (020000 octal, the escape convention bit) is set in X4, then all special scanning conventions are disallowed, and the erase will be trapped with a format error if the treename contains a special first name.

For more information on tally operations, see Chapter 8.

Catalog quota checks are suspended for the TALLY ERASE command; hence, it can be used to destroy files in catalogs whose quotas have been exceeded. Conflict checks with accesses currently held by other jobs having the given file/catalog open are suspended for all accesses except Write. A TALLY ERASE command may therefore be used to destroy a file/satalog which is open, provided that it is not open with Write permission. With the provided that it is not open with Write permission. With the exception of this and the catalog quota check, a TALLY ERASE command is identical in effect to the sequence of commands TALLY OPEN, UNCATALOG, and CLOSE.

A job which issues a successful TALLY ERASE command will have its catalog word allotment incremented by the length of the file or of the header of the catalog which was erased and also by the length of the catalog entry.

On completion of the command, a trap occurs to the location specified in X6.

If the command was unsuccessful, then status word 1 contains the reason for the failure. If the status is 12 or 40 or greater, the two words pointed to by X5 are undefined. If the status was 7 (fetch error) then the two words are cleared. Otherwise, the last file name scanned is placed in the two words. (These words need not be aligned on a even-word boundary.) If the status returned from the open is 12 (format error), then status word 2 will contain a substatus, and a tally count and character position pointing to the character where the format error was detected. See Chapter 8 for more information on format errors.

Status Returns

0 Successful: the file/catalog was erased.

2 Lockout: the file/catalog is currently in use with Write permission.

3 Not found: Search permission was available on some intermediate catalog and the next file/catalog was not found in it.

4 Protection violation: Search permission was available on some intermediate catalog and the password specified for the next file/catalog was wrong.

5 Fail: Search permission was available on the next to last catalog and none of the requested accesses were available on the file/catalog.

6 Bad treename: Search permission was available on some intermediate catalog and the next entry in the treename was not the last entry, but was the name of a file rather than a catalog.

7 Fetch error: either

(1) An error of type 3, 4, 5, or 6 occurred and Search permission was not available on the catalog in which the search was being performed; or

(2) Search permission was not available on the next to last catalog and Fetch permission was not available on the file/catalog.

11 TALLY ERASE disallowed: the named file/catalog is a device file.

12 Format error: see Chapter 8 for possible types of errors.

100 Access error: the initial catalog to search is not open with Search permission; bits 0-8 of status word 1 contain the missing permission bit.

120 Busy: another command is outstanding on the catalog specified in X0 or X3.

200 X0 parameter error: X0 does not contain the file reference number of a catalog.

220 X1 parameter error: the pointer to the tally word is out of bounds.

240 X2 parameter error: X2 was nonzero.

260 X3 parameter error: X3 does not contain the file reference number of a catalog.

300 X4 parameter error: bits 0-3 or 5-8 were nonzero.

320 X5 parameter error: X5 pointed to a two-word pair which is out of bounds.

360 X7 parameter error: X7 was nonzero.

400 Recoverable error: an error occurred in reading an intermediate catalog.

420 Unrecoverable error: information in an intermediate catalog has been destroyed.

MME 500136: TALLY OPEN

X0 File reference number of initial catalog to search

X1 Pointer to a tally word

X2 Reserved for future use (must be zero)

X3 File reference number of alternate initial catalog

X4 Desired accesses on the file, trap bit mask and escape special convention bit

X5 Pointer to two words for name of last file/catalog accessed by the Executive

X6 Pointer to trap block

X7 Reserved for future use (must be zero)

AQ Passed to slave trap program if file is slave trapped

The specified file (denoted by a string of ASCII characters) is located in the file system and opened. A file reference number is returned, which is used on all subsequent operations on the file.

The tally word pointed to by X1 is a tally word for the treename of the file to be opened. A treename is defined to be a list of zero or more catalogs separated by colons (:), with a file name or catalog name at the end. Any catalog/file in the treename may have a password; the password immediately follows the catalog/file name and is set off from it by a comma (,).

If bit 4 (020000 octal, the escape convention bit) is set in X4, then all special scanning conventions are disallowed, and the open will be trapped with a format error if the treename contains a special first name. For more information on tally operations, see Chapter 8.

The procedure for opening the file/catalog specified by the treename is as follows. If a special first-name convention is not used, then the initial catalog specified by X0 is searched for the catalog specified by the first entry of the treename; if X0 is zero, then the Master File Directory (MFD) is searched. If one of the special first-name conventions is used, then the initial catalog specified by X3 is searched regardless of what is in X0. As each succeeding catalog in the treename is found, it is searched for the catalog specified by the next entry of the treename. Finally the next to last name in the treename is searched for the file/catalog specified by the last name.

For every file/catalog in the treename, the accesses available are determined in one of two manners:

(1) If Search permission was available on the previous catalog, then the accesses available on the current file/catalog are those allowed without a password (bits 3-8 of the current file/catalog's access word), or those allowed with a password if one is given (bits 21-26 of the current file/catalog's access word), or those returned by the master trap program if the catalog is so protected.

If Owner permission was available on the previous catalog and the current file/catalog is not protected by the master trap program, then all accesses are allowed on the current file/catalog.

Slave trapping programs are run only on the last file/catalog in the treename. If a supra catalog is slave trapped, the untrapped accesses are used.

(2) If Search permission was not available on the previous catalog, then the accesses as given above are examined for the presence of the fetch bit (bit 4). If the fetch bit is on, then accesses are granted on the current file/catalog just as if Search permission had been available on the previous catalog. If the fetch bit is not on, then no accesses are granted on the current file/catalog.

Next the access bits allowed on the last file/catalog are checked for use conflicts with accesses currently held by other jobs with the same file/catalog open. Access bits that are in conflict are masked off. In addition, Append permission is masked off if the catalog MAX of the next-to-last entry has been exceeded. All access bits are masked off if the allocated storage for any entry exceeds twice its permissible MAX. Accesses are ANDed with those requested, and, finally, bit 0 of the access is set to 1 if the file/catalog is a catalog. The file/catalog is opened if any accesses remain in bits 5-8.

If bit 1 of X4 (200000 octal) is set, then the Executive will open the file in shared mode. If the file is not already open in unshared mode and bit 3 (040000 octal) is set in its access word, then the available accesses will be reduced to a maximum of Read and Write, and those accesses, if any, will be returned.

On completion of the command, a trap occurs to the location specified in X6. If the file/catalog was opened successfully, then the access bits obtained are returned in bits 0-8 of status word 1, the new file reference number of the opened file/catalog in bits 18-35 of status word 1, and the length of the opened file or the MAX of the opened catalog in status word 2.

In addition to the access bits returned in status word 1 of the trap, nine user access bits can be obtained by issuing a REQUEST STATUS. These user access bits are obtained from bits 27-35 of the opened file/catalog's access word if the file/catalog was opened with a password. If the file/catalog was opened without a password or if it was protected by a trap program, the user access bits are obtained from bits 9-17 of the access word.

If the command was unsuccessful, then status word 1 contains the reason for the failure. If the status is 12 or 40 or greater, the two words pointed to by X5 are undefined. If the status was 7 (fetch error) then the two words are cleared. Otherwise, the last file name scanned is placed in the two words. (These words need not be aligned on a even-word boundary.) If the status returned from the open is 12 (format error), then status word 2 will contain a substatus, and a tally count and character position pointing to the character where the format error was detected. See Chapter 8 for more information on format errors.

The read/write pointer of the opened file/catalog is initialized to zero.

Status Returns

0 Successful: the accesses in bits 0-8 were obtained and are the same as those requested.

1 Partial success: the accesses in bits 0-8 were obtained but differ from those requested. This status return can occur if bit 0 of the requested access is 0 (is 1) and the opened file/catalog is a catalog (is a file).

2 Lockout: due to usage conflicts, none of the requested accesses was available on the file/catalog.

3 Not found: Search permission was available on some intermediate catalog and the next file/catalog was not found in it.

4 Protection violation: Search permission was available on some intermediate catalog and the password specified for the next file/catalog was wrong.

5 Fail: Search permission was available on the next to last catalog and none of the requested accesses was available on the file/catalog.

6 Bad treename: Search permission was available on some intermediate catalog and the next entry in the treename was not the last entry, but was the name of a file rather than a catalog.

7 Fetch error: either

(1) An error of type 3, 4, 5, or 6 occurred and Search permission was not available on the catalog in which the search was being performed, or

(2) Search permission was not available on the next to last catalog and Fetch permission was not available on the file/catalog.

10 Off-line file: the file to be opened was an off-line file. Bits 0-8 of status word 1 contain the accesses that would have been available were the file cataloged. Status word 2 contains the DLU in the upper half, the reel number in bits 21-26, and the preference and type in bits 29-35.

12 Format error: see Chapter 8 for possible types of errors.

40 Quotas exceeded: either

(1) The job has exceeded its alloted state vector length; or

(2) Allocated storage for the next to last catalog is over twice the permissible MAX, and files must be erased from this catalog before any can be opened.

60 Out of storage: the Executive's table of opened files is full.

100 Access error: the initial catalog to search is not open with Search permission; bits 0-8 of status word 1 contain the missing permission bit.

120 Busy: another command is outstanding on the catalog specified in X0.

200 X0 parameter error: X0 does not contain the file reference number of a catalog.

220 X1 parameter error: the pointer to the tally word is out of bounds.

240 X2 parameter error: X2 was nonzero.

260 X3 parameter error: X3 does not contain the file reference number of a catalog.

320 X5 parameter error: X5 pointed to a two-word pair which was out of bounds.

360 X7 parameter error: X7 was nonzero.

400 Recoverable error: an error occurred in reading an intermediate catalog.

420 Unrecoverable error: information in an intermediate catalog has been destroyed.

MME 500140: TALLY REPLACE

X0 File reference number of initial catalog to search

X1 Pointer to a tally word

X2 File reference number of scratch file/catalog

X3 File reference number of alternate initial catalog

X4 (bits 0-3,5-8) must be zero, (bit 4) escape special convention, (bits 9-17) trap bit mask

X5 Pointer to two words for name of file/catalog replaced by Executive

X6 Pointer to trap block

X7 Reserved for future use (must be zero)

AQ Passed to slave trap program if file is slave trapped

The specified scratch file/catalog replaces a file/catalog in the file system (denoted by a standard treename).

The TALLY REPLACE command attempts to open the file/catalog using TALLY OPEN. If Read, Write, and Append permissions are available in the catalog accesses and no other job has the file open with Write permission, then an attempt to replace the named file/catalog with the specified scratch file/catalog is made.

If bit 4 (020000 octal, the escape convention bit) is set in X4, then all special scanning conventions are disallowed, and the replace will be trapped with a format error if the treename contains a special first name. The name and password of the file/catalog to be replaced cannot be specified with the "/" convention, and the replace will be trapped with a format error if that is attempted. For more information on tally operations, see Chapter 8.

The replacement will be made unless either the replacement would cause the catalog's quotas to become exceeded, or the catalog's quotas are already exceeded. If the catalog's quotas are already exceeded, then the replacement will be made only if the length of the scratch file or MAX of the scratch catalog is less than or equal to the length or MAX of the file/catalog which it replaces. If the MAX of the catalog is still exceeded after replacement, then Append permission will be removed from the accesses available on the replaced file/catalog.

If the replacement is successful, then the file/catalog specified in X2 becomes a cataloged file/catalog open with the accesses which were determined to be available, and the catalog entry is correspondingly updated. If the replacement was unsuccessful, then the file/catalog remains a scratch file/catalog and the catalog entry remains unchanged.

A job which attempts to replace a file/catalog first has its catalog word allotment decremented by the length of the scratch file or of the scratch catalog header. If the replacement succeeds, then this allotment is incremented by the length of the file or header of the catalog which was replaced. If the replacement does not succeed, then this allotment is incremented by the amount of the original debit.

A file may not be replaced by a catalog, nor may a catalog be replaced by a file. Replacement of a cataloged file/catalog does not affect the name, password, or access word in the catalog entry. The replacement scratch file/catalog must not have been passed to the issuing job with the return bit set.

On completion of the command, a trap occurs to the location specified in X6. If the file/catalog was replaced successfully, then the access bits obtained are returned in bits 0-8 of status word 1, the new file reference number of the replaced file/ catalog in bits 18-35 of status word 1, and the length of the file or the MAX of the catalog in status word 2.

In addition to the access bits returned in status word 1 of the trap, nine user access bits can be obtained by issuing a REQUEST STATUS. These user access bits are obtained from bits 27-35 of the replaced file/catalog's access word if the file/catalog was opened with a password. If the file/catalog was opened without a password or if it was protected by a trap program, the user access bits are obtained from bits 9-17 of the access word.

If the command was unsuccessful, then status word 1 contains the reason for the failure. If the status is 12 or 40 or greater, the two words pointed to by X5 are undefined. If the status was 7 (fetch error) then the two words are cleared. Otherwise, the last file name scanned is placed in the two words. (These words need not be aligned on a even-word boundary.) If the status returned from the replace is 12 (format error), then status word 2 will contain a substatus, and a tally count and character position pointing to the character where the format error was detected. See Chapter 8 for more information on format errors.

Status Returns

0 Successful: the file/catalog was replaced and is now open with the accesses in bits 0-8.

1 Partial success: the file/catalog was replaced and is now open with the accesses in bits 0-8. Append permission has been masked off since the searched catalog's MAX is exceeded.

2 Lockout: the cataloged file/catalog is currently in use with Write permission.

3 The desired file/catalog was not found in the searched catalog.

4 Protection violation: Search permission was available on some catalog and the password specified for the next file/catalog was wrong.

5 Unsuccessful: one of the Read, Write, or Append permissions was not allowed.

6 Bad treename: Search permission was available on some intermediate catalog and the next entry in the treename was not the last entry, but was the name of a file rather than a catalog.

7 Fetch error: either

(1) An error of type 3, 4, 5, and 6 occurred and Search permission was not available on the catalog in which the search was being performed; or

(2) Search permission was not available on the (N-1)st catalog and Fetch permission was not available on the file/catalog.

11 REPLACE disallowed: the named file/catalog is a device file.

12 Format error: see Chapter 8 for possible types of errors.

40 Quotas exceeded: either

(1) The length of the scratch file or catalog header exceeds the job's remaining catalog word allotment, or

(2) The specified catalog's MAX is exceeded, and the length of the scratch file or MAX of the scratch catalog is greater than the length or MAX of the file/catalog to be replaced.

60 Out of storage: either

(1) The Executive's table of opened files is full, or

(2) A new entry had to be made in the catalog for the replacement and either

(a) The catalog already had 4095 entries;

(b) The Executive could not allocate sufficient storage for a required extension of the catalog; or

(c) It is not possible to append to the catalog since it either is a core catalog or is already allocated in twelve fragments.

100 Access error: the initial catalog to search is not open with Search permission; bits 0-8 of status word 1 contain the missing permission bit.

120 Busy: another operation is outstanding on either the catalog or the scratch file/catalog.

200 X0 parameter error: X0 does not contain the file reference number of a catalog.

220 X1 parameter error: the pointer to the tally word is out of bounds.

240 X2 parameter error: either

(1) X2 does not contain the file reference number of a scratch file/catalog;

(2) The scratch file/catalog was passed to the issuing job with the return bit set;

(3) The type of the scratch file/catalog does not match the type of the file/catalog it is to replace; or

(4) The scratch file had a preference of 0, 1, 2, or 4 and can only replace a file in a core catalog.

260 X3 parameter error: X3 is nonzero and does not contain the file reference number of a catalog.

300 X4 parameter error: bits 0-3 or 5-8 of X4 were nonzero.

320 X5 parameter error: X5 pointed to a two-word pair which was out of bounds.

360 X7 parameter error: X7 was nonzero.

400 Recoverable error: an error occurred in reading the catalog.

420 Unrecoverable error: information in an catalog has been destroyed.

MME 500107: TRUNCATE

X0 File reference number of file

X6 Trap location

A Length

The file specified in X0, which must be open with Write and Append permissions, is truncated to the length specified in the A-register. If the file's read/write pointer points beyond the new end of the file, it is reset to point to the end of the file. Upon completion of the command, a trap occurs to the location specified in X6.

A TRUNCATE command issued at the slave end of a communication file will generate a special interrupt number 14 (slave end issued TRUNCATE) at the master end. The second word of the special interrupt will contain the length specified in the A-register. If the communication file is busy, either because another end has an operation outstanding on it, or because another end has reserved it, the slave end issuing the TRUNCATE command will be trapped with a status of 6. If the job at the master end of a communication file is not accepting special interrupts when a slave end issues a TRUNCATE, then that slave end will be trapped with a status of 7. A TRUNCATE command cannot be issued on the master end of a communication file.

A job which truncates a scratch (cataloged) file will have its scratch (catalog) word allotment incremented by the number of words truncated from the file.

Status Returns

0 Successful: the file was truncated to the specified length.

1 Unsuccessful: the specified length is longer than the current length of the file.

6 Communication file busy: the communication file at whose slave end this command was issued is busy because another end has an operation outstanding on it or has reserved it.

7 The master end of the communication file at whose slave end this command was issued is not accepting special interrupts.

20 Status was reset on the TRUNCATE.

100 Access error: the file is not open with Append and Write permissions; bits 0-8 of status word 1 contain the missing access bits.

120 Busy: another command is outstanding on the file to be truncated.

200 X0 parameter error: X0 is zero and does not contain a valid file reference number, or contains the file reference number of a device, job, or master communication file.

MME 500104: UNCATALOG

X0 File reference number of cataloged file/catalog

X6 Trap location

The cataloged file/catalog, which must be open with Read, Write, and Append permissions, becomes a scratch file/catalog with Owner and eXecute/Search permissions added to its accesses if not already present. The catalog entry for the uncataloged file/catalog is destroyed. Upon completion of the command, a trap occurs to the location specified in X6.

A job which successfully uncatalogs a file/catalog has its scratch word allotment decremented by the length of the file or of the catalog header. At the same time its catalog word allotment is incremented by this amount plus the length of the catalog entry.

Status Returns

0 Successful.

1 The file/catalog is not cataloged.

40 Quotas exceeded: uncataloging the file/catalog would cause the job's remaining scratch word allotment to be exceeded.

100 Access error: the file/catalog is not open with Read, Write, and Append permissions; the missing permission bits are set in bits 0-8 of status word 1.

120 Busy: another command is outstanding on the file/catalog.

200 X0 parameter error: X0 does not contain the file reference number of a cataloged file/catalog, is zero, or is the file reference number of a device file.

400 Recoverable error: an error occurred in reading the catalog.

420 Unrecoverable error: information in the catalog has been destroyed.

MME 500134: WRITE

X1 Points to pointer to memory location M1 (may not be zero)

X2 File reference number of destination file (may not be zero)

X4 Flag bits

X6 Trap location

X7 Points to number of words N to write into destination file

N words starting at memory location M1 are written into the destination file, starting at the current location of that file's read/write pointer. Write permission is required on the destination file if the write will alter any existing data in the destination file, and append permission is required if the write will make the destination file longer. If the end of the issuing job's storage is reached, an end-of-file condition occurs and data transmission stops. If any of the words transferred to the destination file are outside the range of that file, and the file is not opened with Append permission, an end-of-file condition occurs and data transmission stops. Upon completion of the copy, a trap occurs to the location specified in X6, and the read/write pointer for the destination file is updated to point to the word beyond the last one written.

A WRITE issued at the slave end of a communication file will generate a special interrupt number 12 (slave end issued WRITE) at the master end. The second interrupt word will contain the length of the WRITE. No data will be transferred until the master end issues a corresponding READ or COPY. (See the description of communication files in Section 5.1.) If the communication file is busy, either because another end has an operation outstanding on it, or because another end has reserved it, the slave end issuing the WRITE command will be trapped with a status of 6. If the job at the master end of a communication file is not accepting special interrupts, the slave end issuing the WRITE command will be trapped with a status of 7.

The flag bits in X4 have the following meanings:

Bit	Function
0-8	Currently unassigned. Must be zero.
9	Must be zero unless X2 contains the file reference number of a master communication file. In that case, the corresponding slave end will be trapped only if a nonzero status return occurs or if this bit is zero.
10	Must be zero unless X2 contains the file reference number of a slave communication file or shared file. In that case, the communication file or shared file will be reserved for the end issuing the WRITE if this bit is one. Otherwise it will be released.
11-17	Must be zero.

Upon initiation of a WRITE command into a scratch (cataloged) file which is open with Append permission, the job's scratch (catalog) word allotment is decremented by the amount by which the destination file may be extended. Upon completion of the WRITE command the job's scratch (catalog) word allotment is incremented by the number of words charged to this allotment but not appended to the destination file.

If the status return in status word 1 is not between 100 and 360 on completion of the WRITE command, then status word 2 contains the difference between the number of words transferred and the number of words requested (i.e. it contains minus the number of words not transferred). If the status in word 1 is between 100 and 360, then status word 2 is zero. If the destination file is a device file, then the lower half of status word 1 contains the status return from that device.

The observant reader will notice that the WRITE command is treated as a special case of the COPY command in which both X0 and X3 are assumed to contain zeros.

Status Returns

0 Successful: all words were transferred.

1 Source file exhausted: some words were requested but not transferred from main memory, since the end of the issuing job's storage was reached.

2 Destination file exhausted: some words were not transferred because the destination file was full and was not open with Append permission.

3 Command inappropriate: the WRITE was issued at the master end of a communication file without having a corresponding READ command outstanding at a slave end of that file.

4 The source file pointer is out of bounds: either,

(1) The pointer was initially out of bounds;

(2) A MEMORY REQUEST released job storage involved in the write; or

(3) An attempt was made to write from the pure region of a job file which was swapped out of main memory.

5 The destination file pointer is out of bounds: either,

(1) The pointer was initially out of bounds;

(2) The other end of a communication file issued a MEMORY REQUEST which released job storage involved in the write; or

(3) An attempt was made to write into the pure region of a job file which was swapped out of main memory.

6 The communication file at whose slave end this command was issued is busy, either because another end has an operation outstanding, or because another end has reserved the communication file.

7 The master end of the communication file at whose slave end this command was issued is not accepting special interrupts.

20 Status was reset on the WRITE command.

40 Quotas exceeded: it was possible for the WRITE command to exceed the job's remaining scratch or catalog word allotment; no words were transferred.

60 Out of storage: the Executive could not allocate sufficient storage for a required extension of the destination file.

100 Access error: the destination file is not open with the necessary Write or Append permission. Bits 0-8 of status word 1 contain the missing permission bits.

120 Busy: another operation is outstanding on the destination file.

220 X1 parameter error: X1 is zero or points to an out-of-bounds location.

240 X2 parameter error: X2 does not contain the file reference number of a file.

300 X4 parameter error: X4 contains invalid flag bits.

360 X7 parameter error: X7 points out of bounds.

400 Recoverable error: an error occurred during transmission of data. The lower half of status word 1 contains the status from the device which generated the error. This return can occur if a device file is being written via a communications file and the issuing job is swapped out of main memory.

Chapter 5

Communication Files

A communication file is a pseudofile which allows direct communication between jobs. A communication file has a master end and one or more slave ends. Each end appears similar to an ordinary file. It has a file reference number which may be used as a parameter for certain Executive calls, principally the COPY command. Other calls which may be used with communication files are DRIVE, PASS, CLOSE, REQUEST STATUS, RESET STATUS, TRUNCATE (or SCRATCH), and SET POINTER.

This chapter describes how these operations affect communication files and explains other features related to communication files. An extended example can be found at the end of the chapter.

5.1 COPY Commands

The COPY command and its variants, READ and WRITE, are used to perform the primary function of communication files: the direct transfer of data between jobs. Generally speaking, data which is copied "into" one end can be copied "out of" the other end. Data transfer is always initiated at a slave end, and takes place between that slave end and the master end.

5.1.1 Slave End COPY Commands

A COPY command involving a slave end of a communication file may be issued at any time that the communication file is not busy (see Section 5.5), and uses the same format as a COPY command involving only ordinary files (see COPY MME). Besides the restrictions normally imposed on the COPY command, the following must be observed:

(1) If the COPY source file (specified in X0) is an end (either master or slave) of a communication file, the COPY destination file (specified in X2) may not be. In other words, only one communication file end may be specified as a parameter for a given COPY command.

(2) If the other file involved in the COPY is a device file (see Chapter 6), then the ultimate source or destination of the data must be main memory. Since this is determined by the job at the master end of the communication file independently of any action which the slave end job can take, it is not recommended that COPY commands be attempted between device and communication files.

When the slave end issues a COPY command, the job holding the master end is notified by a special interrupt of its action. (See "Special Interrupts", Section 1.4.2). Data transfer does not take place until the master end issues a corresponding COPY command. It should be noted, therefore, that COPY command operations issued at the slave end of a communication file may take a comparatively long time to complete. If the master end job is not accepting special interrupts, no COPY commands are possible on the communication file. Any attempts are rejected with the status of "other end not accepting specials" (7).

The total length of data transferred (barring other errors) is the lesser of that requested at the slave end and that requested at the master. The end requesting "too many" words is trapped with the status "source exhausted" (1) or "destination exhausted" (2), as appropriate (unless the master end job exercises the "no-trap" option — see below). This means that these statuses need not have their normal meaning on communication file COPY commands. On an ordinary file, these statuses would mean that further READ or WRITE commands are futile; however, they may represent normal and expected returns on communication file operations.

5.1.2 Master End COPY Commands

As noted above, a COPY command may only be issued at the master end after the job at that end has been notified by a special interrupt that a COPY has been issued at a slave end. It is imperative that the job holding the master end be accepting special interrupts (have a valid, nonzero, special interrupt tally word -- see Section 1.3). If it is not, all COPY commands at slave ends will be rejected with the status of "other end not accepting specials" (7).

When the master end job receives a special interrupt indicating "slave issued READ" (a job requested data transfer from a slave end of the communication file), it should issue a COPY (or WRITE) command transferring data to the master end. Similarly, in response to a "slave issued WRITE" special, it should issue a read-type command. (Types of special interrupts are described in Section 1.4.8). The same restrictions which apply to slave end copies also apply to the master end. That is, only one of the files specified in a COPY command may be an end of a communication file, and copies between a device file and communication file should not be attempted. The previous remarks on the source-exhausted and destination-exhausted statuses also apply to the master end. However, since the length requested by the slave end is contained in the special interrupt pair, it should always be possible for the master end job to predict the file-exhausted statuses.

When copying to or from the master end of a communication file, a file pointer should not be specified (X1 or X3 should be zero). This is because the slave end job has already completely specified the destination or source of the data to be copied through the communication file. If a file pointer is specified, it will be checked for validity and then ignored.

Once the master end job has received the special interrupt, it must eventually issue the corresponding COPY command, unless it resets status on the master end (see Section 5.5). Any other operation on the master end (except REQUEST STATUS) will be rejected as "inappropriate" until the COPY command is issued.

5.1.3 The "No-Trap" Option

When the master end job issues its COPY command, the usual action taken by the Executive is to transfer the requested data and then to trap the COPY at both ends (master and slave) involved. The communication file then reverts to an idle or reserved state, depending on the option selected by the slave end job (see Section 5.4). However, when the master end COPY command is issued, the "no-trap" option may be specified by setting the appropriate flag bit in X4 (see COPY MME). In this case, when the data has been transferred, and if the master end status is "successful" (0), the slave end job is not trapped. Instead, the slave end COPY remains outstanding and more master end COPY commands may be issued to satisfy any part of the slave end's request that was not fulfilled by a previous COPY. If the status presented at the master end is not "successful", the slave end is trapped as if the option had not been specified. In other words, the option has no effect in this case. The no-trap option allows the master end job to satisfy the slave end's request in a piecemeal fashion. For an example of the use of the no-trap option, see Section 5.6.

5.2 Non-COPY Commands

Communication files are temporary entities, existing only as long as the master end and at least one slave end exist. If either the master end or all slave ends are closed, the communication file disappears. Thus the file reference number of an end of a communication file is an invalid parameter for many Executive calls dealing with the catalog structure, such as OPEN, ERASE, CATALOG, or REPLACE. Intended for communication, the ends of communication files cannot be "run". An attempt to do so will result in a parameter error status. Many other Executive calls are similarly restricted to files other than communication files. Those commands which can be used with communication files are discussed here.

5.2.1 REQUEST STATUS (Master End)

A REQUEST STATUS can be issued to the master end of a communication file, and will produce the same four-word status block as for an ordinary file (see REQUEST STATUS MME). The type field of the status block will indicate master communication file. However, the other fields (access, preference, length, pointer) are used by the Executive to maintain the current state of the communication file and will frequently contain meaningless information. Under certain circumstances, however, useful information may be extracted from these fields.

The two high-order user access bits (bits 9 and 10) are used to indicate when a slave end has issued a COPY (to which the master end has not yet responded). B$CW (bit 9) will be on if (and only if) a slave end has issued a write-type command. B$CR (bit 10) will be on if a slave end has issued a read-type command. In either case, the preference, length, and pointer fields will reflect the state of the file at the far end of the communication file; i.e. the other file involved in the slave end job's COPY.

5.2.2 REQUEST STATUS (Slave End)

A REQUEST STATUS to the slave end is treated quite differently. It is essentially a variant of the READ command. When the slave end job issues it, a "slave issued request status" special interrupt is generated at the master end job to write complete status information to the master end. This information is presented to the slave end job as its status block, and no status information is provided by the Executive. No checks are made on the format of this information, and the master end job can write whatever it wishes. However, system convention dictates that the first three words of status information should correspond in format to the status block provided by the Executive for an ordinary file (see REQUEST STATUS MME) and should show a type of slave communication file. The master end job writes status information with a COPY or WRITE command, exactly as if the slave end job had issued a READ command. It may exercise the no-trap option if it desires.

5.2.3 DRIVE (Master End)

Data transfer may not be requested at the master end of a communication file without prior action at a slave end. It is sometimes desirable to allow the master end job to initiate a chain of events involving the communication file, and the DRIVE command can be used for this purpose. When the master end job issues a DRIVE, a special interrupt is generated at a slave end. (The selection of the slave end to receive the special, in cases where more than one slave end exists, is discussed below.) The type of special interrupt is determined by the drive type given in bits 0-17 of the A-register on the DRIVE command. The valid DRIVE types at the master end of a communication file are:

(1) "Read communication file" drive: A drive of type 1 (and the corresponding special interrupt) is conventially used by the master end job to signal a slave end job that the master end job has data which the slave end should read. The master end job may not simply write directly into the communication file, since only a slave end may initiate data transfers. When the drive is issued, a special interrupt is given to a job holding a slave end. If the communication file is not reserved to any slave end (see Section 5.4), the special is given to the lowest job which holds a slave end and which is accepting specials (see Section 1.4). If a slave end has the file reserved, the special must be given to the job holding that end, or not at all. If no job meets these criteria, the master end job has its DRIVE trapped with a status of "other end not accepting specials" (7).

(2) "Break" drive: A drive of type 3 (and the corresponding special interrupt) is conventionally used to notify a slave end job of some sort of "panic-stop" condition. Its most typical use is with a terminal communication file, to indicate the receipt of a break signal from the terminal. Since one of the intended purposes of this drive is to request the termination of runaway slave jobs by a higher job in the tree, this drive is treated specially by the Executive. If no slave end has reserved the communication file, the special is given to the lowest job in the job tree which holds a slave end, which is accepting specials, and which has eXecute ("break") permission on its end of the communication file. If a slave end has reserved the file,

the special is given to the lowest job meeting all these conditions, and which is also not lower than the job holding the reserving end. Thus if the job holding the reserving end meets all the other qualifications, it will receive the special. If no job can be found meeting these criteria, no special interrupt will be given, and the DRIVE command is trapped with the status of "other end not accepting specials" (7).

If a special is given to a job holding a slave end of a communication file due to a drive issued at the master end, then the communication file will automatically be reserved to that job.

5.2.4 DRIVE (Slave End)

Certain drive types may be issued at a slave end of a communication file to generate special interrupts at the master end job. In particular, a set mode drive (type 0) or a device drive (type 12) may be issued. (Only these types are legal.) The two drives will generate the corresponding special interrupts (set mode and slave-issued drive) at the job holding the master end. If the master end job is not accepting specials, the DRIVE will be trapped with the status of "other end not accepting specials" (7). A set mode drive is conventionally used to specify directions for the handling of data to be transferred through the communication file. For terminal communication files in particular, it is used to set input and output modes (see DRIVE MME). The exact interpretation of a "set mode" special interrupt among the jobs holding ends of the communication file is a matter of convention. Device drives are not normally used on communication files but are permitted by the Executive.

5.2.5 TRUNCATE, SCRATCH, and SET POINTER (Slave End Only)

These commands may be issued to a slave end and will generate special interrupts at the master end job, notifying it of the action at the slave end. When a slave end job issues a TRUNCATE to the slave end, a "slave-issued truncate" special is generated. When the slave end job issues a SET POINTER, a "slave-issued set pointer" special is given to the master end job. A SCRATCH issued to a slave end generates a "slave-issued truncate" special specifying a length of zero. As with DRIVE, the "other end not accepting specials" status (7) can occur. When issued to a slave end, these commands will release the reservation on the communication file, if it was reserved to the end at which the command was issued. (See also Section 5.4.) These commands have no other effect on the file, and it is the responsibility of the master end job to take any appropriate action upon receipt of the special interrupt.

5.2.6 PASS and CLOSE

These commands are the means by which communication files are created and destroyed, and additional slave ends are added and lost. A complete description of the use of these commands with communication files is given in Section 5.3.

5.2.7 RESET STATUS

Since several jobs can be affecting or attempting to affect the state of a communication file, the business of maintaining that state, and consequently of resetting it, becomes rather complicated. Section 5.5 is devoted to explaining the various possible states of a communication file and the role of RESET STATUS in modifying its state.

5.3 Creation and Destruction

A communication file is initially created by a create-type PASS command (or by the implicit PASS available in the RUN command). A communication file will be created for a job which issues a PASS specifying a file reference number of zero to be passed, and which turns on the communication file bit (bit 1) in the access to pass. The file reference number returned to the passing job in the trap of the PASS (or in the pass list of a RUN command) is the file reference of the master end. This end cannot be passed; it remains with the job which created it until the communication file is destroyed. The job to which the PASS is issued receives the file reference number of a slave end in a "passed file" special interrupt (see types of special interrupts in Section 1.4.8).

The master end is always allowed all significant access permissions. The slave end is allowed whatever accesses are specified in the PASS command. These accesses need not include "break" (eXecute) permission; if they do not, however, the master end job will never be able to issue a break drive successfully.

Once created, a slave end may be passed, subject to the restrictions which govern the passing of an ordinary file (see PASS MME). If a slave end is passed without the return bit set in the accesses, it becomes permanently unavailable to the passing job (as is the case with an ordinary file). Its file reference number at the passing job becomes invalid and may be reassigned by the Executive to the next file which the job opens or receives via a PASS. Unlike an ordinary file, if a slave end is passed with the return bit set, it does not become busy. Instead, the passing job may continue to use its slave end, and a new slave end is created for the job at the receiving end of the PASS. For this reason, a communication file may be passed more than once (via a PASS command or implicit PASS on a RUN) and have more than one slave end. Once a job has passed its slave end with the return bit set, it may not CLOSE or PASS it again until the newly created slave end is returned (i.e. closed by the receiving job). The ends of a communication file are linearly ordered from lowest (most recently created link in the chain of passes) to highest (master end).

Note that it follows that only the lowest slave end can be closed, since any higher slave ends have been passed, and cannot be closed until the passed end has been returned. When the lowest slave end is closed, if it has been passed to the closing job with the return bit (is not the highest slave end), that end disappears. The job holding the next lowest slave end is notified by a "returned file" special interrupt (if it is accepting specials) that the file has been returned. This process can take place even if other ends of the communication file are busy with data transfer or other operations (see Section 5.5). The creation or destruction of extra slave ends is a process local to the ends being created or destroyed and does not involve the communication file as a whole.

On the other hand, when the last remaining slave end has been closed, or when the master end is closed (if ever), the entire communication file is destroyed. Any ends that remain after the communication file is destroyed are changed to a special file type called a "non-file" (13), and the jobs holding these non-files are notified by file-closed special interrupts that the communication file has been closed. The file reference number of a non-file is an invalid parameter for any command except RESET STATUS or CLOSE. However, the file reference number is not freed for reassignment until the job acknowledges the receipt of the special interrupt by closing the non-file. It is possible for both ends of a two-ended communication file to be closed sufficiently close together in time to be considered simultaneous. In this case, no non-files are created, and no special interrupts are generated. Note that a job using communication files should be prepared to receive parameter errors on its commands, since the communication file can be destroyed by the jobs at the other ends at any time, leaving the job in question with a non-file. If the job is not accepting specials, this parameter error could be the only indication that the communication file no longer exists.

5.4 Reserving the File

It is sometimes desirable for a job at one end of a communication file to complete a sequence of operations on the file without danger of interference from the other ends. For example, one slave end job might wish to write a message to the master and then read a reply, without the possibility that another slave end might "sneak in" between its operations and read its reply from the master end. The process of reserving the communication file allows for uninterrupted sequences of operations. Either a slave end or the master end may reserve the file, but only one end can reserve it at any given time. Reservation by a slave end has the effect of locking out all other slave ends from the communication file. Only local operations (see Section 5.5) can be issued at the other slave ends. When the master end reserves the file, all slave ends are locked out.

Reservation is specified by setting the appropriate flag bit when issuing those commands which allow communication files to be reserved. A slave end may reserve a communication file on a

COPY, READ, WRITE, or DRIVE command. The master end may reserve the file only with a RESET STATUS command.

Once the file has been successfully reserved, it remains reserved until an operation is issued at the reserving end which does not call for continued reservation. There are two exceptions to this rule, however:

(1) If the master end resets status on the communication file, specifying that it wishes to reserve the file, then the slave end reservation, if any, is nullified.

(2) If a break special is given to a slave end other than the currently reserved slave end (because a break drive was issued at the master end), then the current reservation is nullified and the communications file is reserved to the end receiving the break special.

The actual reserving is specified by setting the appropriate flag bit in an index register. B$RSVS (bit 1) is used to reserve the file whose file reference number is in X0 (the source file on copy-type commands), while B$RSVD (bit 10) is used to reserve the file whose file reference number is given in X2 (the destination file in copy-type commands. (See the MME descriptions for more details on formats for individual commands.) Any operation which does not explicitly specify reservation will release an existing reservation (by the same end), except that a REQUEST STATUS issued to the master end will not release an existing reservation.

When a slave end has reserved the file, it appears busy to all other slave ends. Only local operations can be issued at the other slave ends. The master end is not restricted by a slave end reservation. When the master end has reserved the file, it appears busy to <u>all</u> slave ends. This gives the master end job the ability to "squelch" a runaway slave end job.

The routing of special interrupts is also affected by slave end reservations. This insures that slave ends which are locked out of the file do not receive special interrupts to which they are unable to respond. The routing algorithm is described in Section

5.2. The general rule is that if a slave end has reserved the communication file, special interrupts generated by the master end DRIVE commands will be given to the job holding that slave end, or not at all. The only exception is a break drive: in this case, if the reserving end does not have break permission, the break special "bounces" up the communication file until it finds a slave end which does have break permission. Thus a reserving slave end job is not only guaranteed that no other slave end is stealing its data, but also that its special interrupts are not being led astray.

5.5 Busy States and Reset Status

Since many jobs may attempt to use a communication file at the same time, some restrictions are necessary with respect to which may do what to the file and when. The general rule governing the use of the communication file is that only one operation which involves the file as a whole may be in progress at one time. The "communication file busy" status (6) is a special status returned to jobs which attempt to initiate operations on a communication file on which another "whole-file" operation is in progress. This status is also returned if the file is reserved to another end. The regular "busy" parameter error status (120) is used only when the end at which the operation is issued is busy; that is, another operation has been issued at the same end and has not yet been trapped.

Thus there are two different concepts involved here: that of the communication file's being busy, and that of one of its ends' being busy. Most operations involve both the file and at least one of its ends; however, there are certain local operations which tie up only the end at which they are issued. There are cases in which the file can be busy without any of its ends being busy. Local operations may be issued even if the file itself is busy.

The state of the communication file, as it appears at any one end, depends on the operations which have been issued at that and at other ends. The function of the RESET STATUS command is always to force the end at which it is issued to be idle (not busy). It may have additional effects, depending on the state of the communication file. These effects are discussed below, state by state, beginning with those presented to a slave end.

5.5.1 Slave End States

First, the file and the slave end may both be completely idle. In this state, any valid operation may be issued at the slave end. If a RESET STATUS is issued, it functions as a purely local operation. It has no effect and traps with good status.

Second, the file may be reserved to this slave end, with the end's being idle. The file will thus appear busy ("communication file busy") to all other slave ends. Operations may be issued, however, at the master end. If no operations are issued at the master end, the file will apear idle to the reserving slave end, and any valid operation may be issued. However, if the master end has an operation pending or has reserved the file, the file will appear busy to the slave end, and only local operations may be issued. In either case, RESET STATUS functions as a local operation, with the additional effect of releasing the slave end reservation. It will not, however, change the file's busy state if it is busy due to the action of the master end.

Third, a local operation may have been issued at the slave end. Local operations are RESET STATUS, PASS, and, if the end was passed with the return bit set, CLOSE. If the return bit is not set and if CLOSE is legal, then this is the only slave end. In this case, CLOSE causes the entire communication file to be destroyed. Thus it is not a local operation, and can only be issued when the file is not busy. (There is one exception to this rule — see below.)

When a local operation has been issued, the slave end will be busy. The state of the file as a whole is immaterial to the state perceived by this slave end. Local operations may be issued even if the file is busy or is reserved to another end, and do not cause the file to become busy. Neither PASS nor CLOSE can be reset. If a RESET STATUS is issued while a PASS is in progress, it will have no effect and will trap after the PASS command is complete. Once a CLOSE command has been issued, the file reference number becomes invalid, so that no RESET STATUS can be issued.

Fourth, the slave end may be idle, but the file may be reserved to another end or may have an outstanding operation initiated from another end. In this case, the file appears busy ("communications file busy") to this end. Only local operations may be issued, except that a CLOSE may be issued at the last remaining slave end whenever both it and the master end are idle, even if the file is reserved to the master end. A RESET STATUS is a purely local operation in this fourth case, and does not change the file's busy state, since this end is not responsible for its being busy.

Fifth, the slave end may be busy because a drive-type operation (DRIVE, TRUNCATE, SCRATCH, or SET POINTER) has been issued at this end. In this case, the file itself will also be busy, and no operations except local ones may be issued at any other end. The operation may be reset by a RESET STATUS if the special interrupt has not already been given to the master end job. In any case the RESET STATUS will trap after the DRIVE has trapped. The status on the DRIVE will indicate if the special interrupt was given. Note that the DRIVE may also be reset by a RESET STATUS issued at the master end. If a RESET STATUS is issued at the slave end, any reservation specified in the DRIVE will be released.

Finally, the slave end may be busy because of a copy-type operation (COPY, READ, WRITE, or REQUEST STATUS) issued at this end. This also has the effect of locking out all other slave ends. The master end, of course, must issue the corresponding COPY command, unless it issues a RESET STATUS. The action of a RESET STATUS issued at the slave end depends upon whether the master end job has already issued that COPY. If it has, then both COPY commands (master and slave) are trapped with the "status was reset" status (20). Both ends become idle, as does the communication file itself. Any reservation by the slave end is released. On the other hand, if the COPY command has not yet been issued at the master end, the master end job is notified by a special interrupt ("slave issued reset") that its COPY is no longer required. The slave end COPY is trapped, both ends become idle, and any reservation to the slave end is released; however, the file itself remains busy until an echoing RESET STATUS is issued at the master end. No operations except local operations or CLOSE can be issued at any end until the master echoes the slave's RESET STATUS. This restriction is imposed to prevent the master end job from being confused by a barrage of special interrupts caused by slave end jobs issuing COPY and RESET STATUS commands in rapid succession. Thus, after a COPY has been issued, some positive action is always necessary at the master end before the file is idle again.

5.5.2 Master End States

The situation is complicated slightly at the master end by the fact that the master end can be reserved with a RESET STATUS command; however, it is correspondingly simplified by the fact that there is always exactly one master end of any given communication file. The states visible at the master end are discussed below.

First, the master end and the file may both be idle. This will be the case even if a slave end has issued a local operation. In this state, the master end job may issue DRIVE, CLOSE, or REQUEST STATUS commands. A RESET STATUS may be issued but will have no effect unless it causes the file to be reserved to the master end.

Second, the file may already be reserved to the master end. Unless the last remaining slave end has been closed, any operation may be issued at the master end which could be issued if the file were completely idle. A RESET STATUS will have no effect other than to release the reservation, unless it specifies to continue it.

A third case occurs when a local operation has been issued at the master end. The only local operation which can be issued at the master end is REQUEST STATUS. The situation is similar to a local operation at a slave end; the master end becomes busy, but the state of the file itself is not affected. REQUEST STATUS at the master end cannot be reset. A RESET STATUS will trap after the REQUEST STATUS is complete with no effect other than reserving or releasing the file. A REQUEST STATUS issued at the master end will not release a reserved communication file.

Fourth, the master end and the communication file are both idle (from the master's point of view) but a slave end has reserved the file. From the master end, this is indistinguishable from the case in which the file is completely idle. A RESET STATUS at the master end has no effect on slave reservations unless it specifies to reserve the file to the master end, in which case the slave reservation is nullified.

The fifth case arises when a slave end has issued a drive-type command but the special interrupt has not yet been given to the master end job. This is clearly a rather transitory state. In this case, the master end is idle, but the file is busy. No operation can be issued at the master end except REQUEST STATUS or RESET STATUS. If issued in time, a RESET STATUS will prevent the special interrupt from occurring, and will cause the slave end DRIVE to be trapped with the "status was reset" status (20). Other than that, it has no effect other than possibly reserving the file. The file will be idle after the special interrupt is given or the RESET STATUS traps, unless another operation is issued.

A sixth posible state occurs when a slave end COPY command has been issued, but the master end is idle. In this case, all operations other than the correct COPY response (except REQUEST STATUS, RESET STATUS) will be rejected as inappropriate. A RESET STATUS will cause the slave COPY to be aborted and will return the file to an idle (or possibly reserved) state. (The slave end COPY will be trapped with the "status was reset" status.)

Seventh, the master end may have issued an operation: either DRIVE, on its own initiative, or COPY, in response to a slave end copy-type operation. In both cases, both the file and its master end are busy. A RESET STATUS will abort a COPY command and return the master end and the file to an idle (or reserved) state. A master end DRIVE cannot be reset; a RESET STATUS will trap after the DRIVE completes, reserving the file if specified.

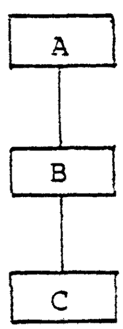
The case which has no real parallel to the slave end situation occurs when the master end job has received a slave-issued reset special interrupt and has not yet echoed it. Although the master end is idle, no operation except CLOSE or REQUEST STATUS can be issued to that end before a RESET STATUS is issued. Other commands will be rejected as being inappropriate. The RESET STATUS will return the file to an idle (or possibly reserved) state.

Finally, when the master end issues a CLOSE, both it and the file become busy until all slave ends have been destroyed. The CLOSE then traps and the master end is destroyed. A RESET STATUS issued before the CLOSE traps will be rejected.

5.6 Usage Example

5.6.1 Creating a Communication File

Suppose that job A has run job B, which has in turn has run job C. Our portion of the job tree thus looks like:



Job A now issues a PASS command to job B, specifying a file reference number of zero (create-type PASS) and access bits for Read, Write, Append, eXecute ("break"), and communication file. A communication file is thus created. Job A receives the file reference number for the master end (suppose it is 5) in the trap of the PASS, and job B receives the file reference number of the slave end (suppose it is 7) in a "passed file" special interrupt. The file is now idle.

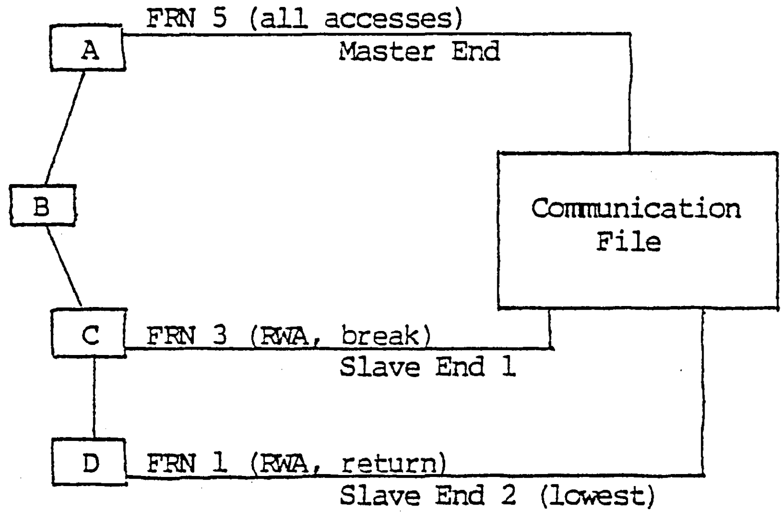
5.6.2 REQUEST STATUS from a Slave End

Upon receipt of the "passed file" special interrupt, suppose job B attempts to get more information about its new file by issuing a REQUEST STATUS. It therefore issues one to its file reference 7 (the slave end), requesting ten words. A special interrupt (slave-issued read) is generated at job A, specifying a length of ten words and file reference number 5 (job A's end). (See Section 1.4 for a description of the format of a special interrupt.) Job A then issues a WRITE to file reference number 5 (the master end) for four words, placing a standard status block in the first three words and a special identifying code in the last word. These four words are presented to job B as its status block. Job A's WRITE operation is trapped with a status of "successful" (0), since all the data that job A requested were transferred. But job B has its REQUEST STATUS trapped with a status of "source exhausted" (1), since only four of the ten words it requested were available.

5.6.3 PASS at a Slave End

Suppose now that upon examining the data received on its REQUEST STATUS, job B decides that job C should have the slave end. It therefore issues a PASS to job C, specifying file reference number 7 (its slave end) and access bits for Read, Write, Append, and break. By not setting the return bit, job B effectively puts itself out of the picture for the duration of the communication file. Its file reference number 7 becomes invalid, and job C receives a passed file special interrupt specifying its file reference number for the slave end (suppose it is 3).

Suppose job C immediately issues a RUN on a new job (job D, of course), specifying file reference number 3 as the first file in the pass list, with access bits for Read, Write, Append, and return. Job D is thus run with a slave end as its file reference number 1. Since job C set the return bit, it retains its slave end (still its file reference number 3). The diagram now looks like:



5.6.4 COPY Command with No-Trap Option

Job D now issues a COPY command from its storage to file reference number 1 for twenty words. A "slave-issued write" special interrupt is generated at job A specifying file reference number 5 and a length of twenty. Job A issues a READ for ten words, specifying the no-trap option by setting B$NTPS in the flag bits. The first ten words are then transferred from job D's memory to job A's memory, and job A is trapped with the "successful" status (0). Job D is not trapped, since the master end job (job A) specified the no-trap option.

Suppose job A now issues another ten-word READ, not specifying the no-trap option. The last ten words are copied from job D to job A, and both jobs are trapped with the "successful" status. The communication file is now idle again.

Suppose job C should now happen to issue a COPY from its file reference number 3 (a slave end of the communication file) to some other file (its file reference number 4, for example), for five words. Job A receives a "slave-issued read" special interrupt. If job A then issues a WRITE for ten words, specifying the no-trap option by setting B$NTPD in the flag bits, the first five words will be transferred from job A's storage to job C's file reference number 4. Job C will then be trapped with the "successful" status, while job A is trapped with the status "destination exhausted" (2). Note that the no-trap option had no effect, since the status presented at the master end was not "successful" (0).

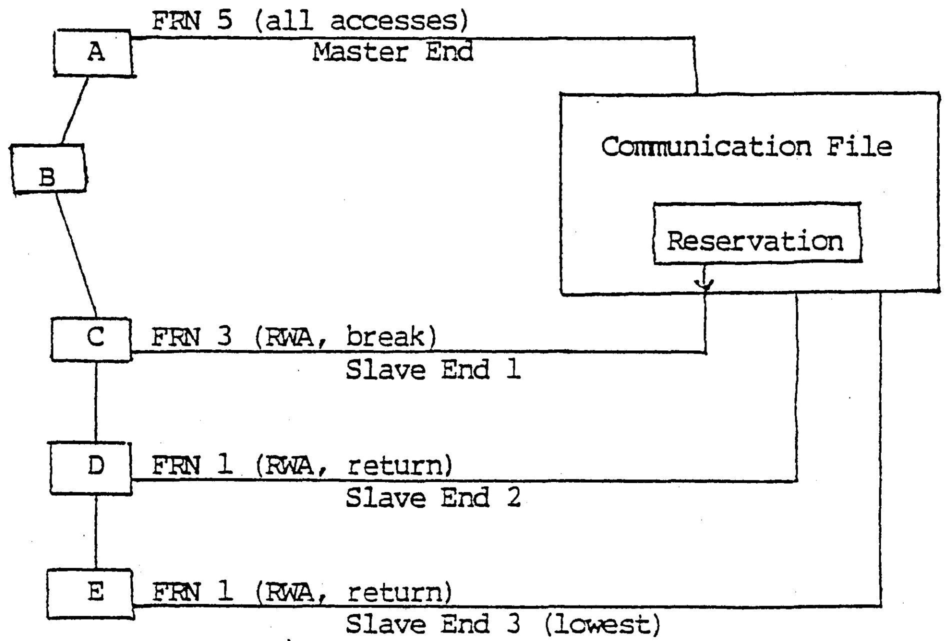
5.6.5 Aborted COPY and Break Drive

Suppose job D now issues a READ to file reference number 1. Job A is notified with a "slave-issued read" special interrupt. However, job A decides that a break drive should be given. It issues the DRIVE command. The drive is rejected with the "inappropriate" status (3) because the slave end has issued a copy-type command. Job A therefore issues a RESET STATUS requesting the communication file to be reserved. Job D's READ is trapped with the "status was reset" status (20), and the file becomes reserved to the master end. If job D attempts to reissue the READ, it will receive the "communication file busy" status (6). Job A now issues the break drive. Upon checking, the Executive discovers that job D does not have break permission on its end of the communication file. If we suppose that job C is not accepting special interrupts, job A's DRIVE will be trapped with the status of "other end not accepting specials" (7). The file becomes idle, since job A's DRIVE had the effect of releasing its reservation.

5.6.6 Slave End Reservation; Local Operations

Now consider what happens if job C issues a set mode drive requesting reservation of the file. Job A receives the special interrupt (set mode, type 0). The file is now reserved to the first slave end (at job C).

Suppose job D now runs a new job (job E), passing it the slave end with the return bit set. It may not pass break permission, since it does not have break permission to pass. The diagram is now:



Job D's implicit PASS is legal, since it is a local operation. Any nonlocal operation attempted by job D would be rejected, since the file is reserved to job C's end.

Similarly, if job E now attempts to issue a REQUEST STATUS on the file it was just passed, it will receive the "communication file busy" status.

If job A now issues a "read communication file" drive, the special interrupt will be given to job C (assuming it is accepting specials). This will be true even if job E or D is accepting specials, since the file is reserved to job C's end. Suppose that job C, on receipt of the "read communication file" special, issues a READ command, not specifying reservation. The reservation is released, and the file becomes busy until job A issues the corresponding WRITE command. When it does so, the data is transferred, and the file becomes idle.

5.6.7 Master End DRIVE

Job A now issues another "read communication file" drive. Since no slave end has reserved the file, the Executive attempts to give the special interrupt to the lowest slave end (at job E). However, let's suppose that job E is not accepting specials. Therefore, the "read communication file" special interrupt will "bounce" up to job D. Assuming job D is accepting specials, it will receive the special interrupt. Note that the communication file is automatically reserved to job D now. If job C were to issue a READ command at this point, it would fail with the status "communication file busy" (6).

5.6.8 Illegal Actions

Suppose job C now attempts to run a new job (this would be job F, presumably), passing it an end of the file (passing its file reference number 3). The RUN is rejected because, although job C's end (file reference number 3) is not busy, it has already been passed and cannot be passed again until it is returned.

Job A now issues a WRITE command to the master end. The WRITE is rejected with the "inappropriate" status, since no slave end has a READ outstanding.

Job D now issues a READ command. This is legal. On receipt of the special interrupt (slave-issued read), job A issues a COPY from the slave end of a different communication file to file reference number 5 (our master end). This is illegal, since only one of the files involved in a COPY may be the end of a communication file. The COPY is thus rejected, without affecting the state of the communication file. If job A now issues a CLOSE on the master end, it too will be rejected, since job D's READ is still outstanding.

5.6.9 Destroying the File

With the file in this fouled-up state, job A issues a RESET STATUS to clear out job D's READ and to reserve the file. It then issues a CLOSE. Before the Executive can process job A's CLOSE, suppose that job E also closes its end of the communication file. Now things start happening. Job D receives a "returned file" special interrupt due to the closing of job E's end. This special is closely followed by a "file closed" special as its file reference number 1 is changed into a non-file by the master end CLOSE. Job C still holds a slave end of the disappearing communication file. This end is changed to a non-file, and job C receives a "file closed" special interrupt. No more slave ends exist, so job A's CLOSE is trapped. Job A's file reference number 5 becomes invalid.

Jobs D and C are still left with non-files. Job D issues a CLOSE to its file reference number 1, and its non-file disappears. Job C, though, being a little slow to process its file-closed special, attempts to issue a set mode drive to what used to be its slave end (file reference number 3). The DRIVE is rejected with a parameter error status, since drives are not legal on non-files. Job C finally closes its file reference number 3, and the last trace of our communication file disappears from the system.

5.6.10 Passing Back a Communication File

Not included in this example is the capability to create a communication file for which the highest slave end is passed to the creating job's immediate supra job (PASSBACK). There is no operational difference, however, between this type of communication file and communication files created in the normal manner.

Note that since passing upward must be done when the communication file is created, the file may backtrack at most one level up the job tree. This and any subsequent passing of the created slave end in the immediate supra job are otherwise subject to the rules associated with the PASS MME regarding normal communication files.

Chapter 6

Device Files

When a slave job wishes to access a peripheral it makes use of a device file. Device files exist for such peripherals as the terminal controllers and the high-speed printer. Device files are always of type 6 (special cataloged file). They are associated with nonallocatable devices. Since many devices have specialized functions which a slave mode program may wish to use, the DRIVE command with function code of 12 (single device action) or 24 (drive with DCW's) is provided for this purpose. A typical example is "rewind" on a magnetic tape handler. This chapter describes the special features and restrictions of device files, and special programming techniques used with them.

6.1 General Information and Restrictions

Currently, all device files are cataloged in :PERCAT, a direct subcatalog of the master file directory. A slave job opens a device file in the same way as it would any other file in the catalog tree (see OPEN MME). This file reference number thus obtained can then be used as a parameter for a copy-type command in order to transfer data to or from the device, or for the DRIVE command to activate special functions of the device.

Since device files reflect configured features of the hardware, they cannot be created, destroyed or altered as can normal (allocated) files. Any attempt to use a special cataloged file as a parameter for a RUN, ERASE, UNCATALOG, TRUNCATE, SCRATCH, or REPLACE type command will be rejected.

Since all devices currently accessible as device files transfer data to or from a medium which is essentially external to the main computer system, the "length" of a device file is meaningless. It will always appear as zero to a slave user.

Because certain peripherals depend on the transfer of large records in a single physical operation (magnetic tape), or of records with a particular length (card equipment), the Executive is unable to guarantee that appropriate buffering will be available in master mode storage to copy between a device file and another noncore file. Thus, copy-type commands are restricted so that if one file in the copy is a nonallocatable device, the other must be a core file or a job file. In general a slave job will only transfer data between its own core (file reference number 0 on a copy-type command) and a device file. If a copy between a device and a noncore file is attempted, it will be rejected with a status of unrecoverable I/O error (420).

In addition, if a job copying between its own core and a device file via a communication file should be swapped out of core before the copy can be initiated, the copy will not be able to proceed, and will be trapped with a status of recoverable I/O error (400). The copy must then be reissued. Note that in this case, the lower half of status return return word 1 will be zero. This serves to distinguish from the case in which an error occurred in the actual transfer of data.

If a copy or drive with DCW's is successfully initiated between core and a device file, a section of the Executive concerned with physical device operations is called. This section, called Physical I/O (PIO) will attempt to transfer exactly one physical record of data (whenever such a quantity is defined) for each COPY command issued on a device file. Physical I/O can also be called directly by the DRIVE command in order to request special functions or transfer more than one record to or from the device. Physical I/O produces a status word based on the queue word generated by the input-output multiplexer (IOM) in the hardware-defined input-output sequence. The format of this status word is as follows:

SYNC	POWER	MAJOR STATUS	SUB-STATUS	QUEUE ADDRESS	IOM CHANNEL STATUS	IOM CENTRAL STATUS	RETURN CODE	RECORD COUNT RESIDUE
0	1	2 5	6 11	12 17	18 20	21 23	24 29	30 35

On a single device action DRIVE command this word is returned to the user in status return word 2 of the trap. On a copy-type command which involves device files or which results in a recoverable error (status return 400), and on multirecord drives, the upper half of this word is returned in the lower half of status return word 1. The lower half is not returned in this case.

The various fields of this word have the following meanings:

SYNC BIT (Bit 0) - This bit is 1 whenever the status originated in the IOM. If it is 0, the status was simulated by Physical I/O. Statuses simulated by Physical I/O are discussed below.

POWER BIT (Bit 1) - If this is 1, the device in question is powered off.

MAJOR STATUS (Bits 2-5) - This field contains the major device status provided by the hardware. Major status 12, 13 and 15 will occur only on devices controlled by a Microprogrammed Peripheral Controller (MPC). The general meanings of these statuses are:

0 - Subsystem ready (good status)

1 - Device busy

2 - Device attention (This normally indicates that the device is in an inoperable condition which must be corrected by the computer operator.)

3 - Data alert (This normally indicates that some abnormal condition occurred during the data transfer.)

4 - End-of-file condition

5 - Command rejected (The peripheral subsystem did not accept the command.)

6 - Intermediate condition (specific to the type of device)

7 - Timeout status (This status is simulated by Physical I/O when an operation has not completed in a reasonable amount of time.)

10 - Channel busy. A hardware or Physical I/O error has occurred.

12 - MPC Device attention (The MPC detected an inoperable device. This normally requires operator intervention to clear.)

13 - MPC Data alert (The MPC detected an error in the data.)

15 - MPC Command reject (The MPC could not process the command.)

SUBSTATUS (Bits 6-11) - This field contains additional information about the operation just completed. It is specific both to the type of device and the major status. For formats and more details, consult the Honeywell peripheral subsystems manual for the specific type of peripheral or the Honeywell GCOS I/O Programming manual, DB82.

QUEUE ADDRESS (Bits 12-17) - This field contains the relative address of the interrupt queue word for this operation, as it would be generated by an IOC (input-output controller). This field is normally not of interest to the slave user, with the following exception: A queue address between 40 and 57 (octal) represents a terminate interrupt, which is the normal way for most operations to complete. A queue address between 20 and 37 (octal) represents an initiate interrupt, which usually indicates an error, except on operations such as rewind tape or console alarm which do not require the peripheral subsystem to become busy. A queue address of 0 should only occur when a status is simulated by Physical I/O; other queue addresses should never occur.

IOM CHANNEL STATUS (Bits 18-20) - This field represents the internal status of the IOM upon termination of the operation. A nonzero IOM Channel Status usually indicates an Executive or channel failure. The meaning of these codes is as follows:

0 - No error
1 - connect while busy
2 - illegal instruction
3 - incorrect DCW
4 - incomplete command sequence
5 - not used
6 - parity error (peripheral to channel)
7 - parity error (IOM central to channel)

IOM CENTRAL STATUS (Bits 21-23) - This field contains the status returned by IOM Central, which performs list service for the channels. A nonzero status in this field indicates an Executive failure. The status codes are:

0 - No error
1 - LCW tally runout
2 - Two TDCW's in a row
3 - LPWX boundary error
4 - TDCW attempted to change address extension from restricted mode
5 - IDCW encountered in restricted mode
6 - Character position incompatible with size
7 - Parity error from channel

RETURN CODE (Bits 24-29) - This code is generated by Physical I/O as a general summary of the result of the operation. Codes are:

0 - Good; no error
1 - Incomplete data transfer, but no error
2 - End-of-file condition. This can occur on magnetic tape or card reader.
3 - Recoverable error
4 - Unrecoverable error. Do not reissue the command.
5 - Timeout status. This is always a simulated status.

RECORD COUNT RESIDUE (Bits 30-35) - This field has the number of records requested minus the number actually transferred.

6.2 Device Drives

A number of device operations do not fall into the READ/WRITE categories. These are invoked by the DRIVE command (see Chapter 4). There are two kinds of device drives. Single action drives change the state of the device. Drives with DCW's transfer data to or from the device. Since drives are valid only for particular device types; the abbreviations below are used to indicate the device types for which a given operation is valid.

Abbreviation	Device
CR	Card Reader
CP	Card Punch
PR	Line Printer
MT	Magnetic Tape Handler
CN	Console Typewriter
DN	Datanet-30
MP	Microprogrammed Peripheral Controller
DS	Mass Storage Unit (Disk)
H7	Honeywell 716 Front End Processor

6.2.1 Single Action Drives

A single action drive has a code of 12 (octal) in the upper half of register A. The lower half contains an 18-bit MODE code, which identifies the operation to perform. These codes are:

Mode	Description
070000	(MP) Reset MPC. Puts the MPC into an idle state. This should be done before it is reloaded.
070000	(CN) Reset console. Any pending interrupts from the console are cleared.
100000	(CR,CP,PR,MT) Await-Ready. Waits for the device to be in an operable condition, unless the device power is off. Traps with a bad status if power is off.
110000	(CR,CP,PR,MP) Await-Special. Waits for a hardware special interrupt to occur on the channel to which the device is connected. As opposed to the await-ready drive, this command will wait even if the device power if off. (Hardware special interrupts should not be confused with the software special interrupts returned to a slave job at location 20.)
140000	(ALL) Enable Error Recovery. Sets normal error recovery mode so that any appropriate recovery procedures are attempted by the Executive when an error is detected.
150000	(ALL) Supress Error Recovery. Causes Physical I/O to treat any future statuses generated by the device as normal, and to pass them back to the caller immediately, until an enable error recovery drive is set. This mode is intended only for the use of hardware test and diagnostic programs, and should be used with caution. Await-ready drives issued when this mode is set will trap whether the device is ready or not, since Physical I/O treats all statuses as ready. An ICM/MEM interface error detected while this mode is set will be returned to the caller with a return code of 3 (recoverable I/O error).

200000	(CR,CP,MT) Set Binary Mode. All subsequent copy-type commands will be done in binary mode. Status on this command is always simulated by Physical I/O.
200000	(H7) Set Normal Mode. I/O to the H716 will be done with normal Read and Write commands.
210000	(CR,CP,MT) Set Decimal or Mixed Mode. The inverse of set binary mode.
210000	(PR) Set BCD Mode. BCD data is to be written to the printer.
210000	(H7) Set Dump Mode. I/O to the H716 will be done with Forced Read and Forced Write commands.
220000	(MT) Set ASCII to EBCDIC Conversion Mode. On output data will be converted from ASCII to EBCDIC. On input data will be converted from EBCDIC to ASCII.
231000	(MT) Set 200 BPI. This can only be done on 7-track tape handlers.
232000	(MT) Set 556 BPI.
233000	(MT) Set 800 BPI.
234000	(MT) Set 1600 BPI. This can only be done on 9-track tape handlers.
235000	(MT) Set 6250 BPI. This can only be done on certain 9-track handlers.
240000	(MT) Set Default High Density. This is usually 800 BPI. The value is set when the system is initialized.
250000	(MT) Set Default Low Density. This is usually 556 BPI. The value is set when the system is initialized.
260000	(PR) Set ASCII Mode. ASCII data is to be written to the printer.
270000	(MT) Set File Protect
300000	(MT) Forward Space Record

310000 (MT) Backspace Record

320000 (MT) Forward Space File. An end-of-file return is, of course, normal on this drive.

330000 (MT) Backspace File. The next read after this operation normally results in an end-of-file return.

340000 (MT) Erase. This erases the tape. It should not be confused with the Executive ERASE command.

340000 (CN) Alarm. This starts the console alarm ringing.

350000 (MT) Write End-of-File. Writes a standard file mark character and end-of-file gap. On a 7-track tape this is 17(8). On a 9-track tape this is 23(8).

3600cc (MT) Write Single Character. Writes a single-character record consisting of character cc. (This end-of-file record may be treated improperly by other computer systems.)

370000 (MT) Rewind

371000 (MT) Rewind and Unload. Both this and the rewind drive trap immediately on initiate interrupts. An await-ready drive can be issued if it is desired to wait until the tape finishes rewinding.

6.2.2 Drive with DCW's

24

A Drive with DCWs has a drive type of 29 in AU. AL has the MODE. The record count (number of records to transfer) is in bits 30-35 of AL. This must be from 1 to 63, or zero which signifies a record count of 64. The number of IOTD's in the DCW list must be at least as large as the record count.

The data control words (DCW's) pointed to by register X1 have the following format:

Bits	Meaning
0-17	Starting memory address to/from which to write/read
18-21	Must be zero
22-23	Action code
29-35	Word count

The possible action codes are:

Bits 22-23	Name	Action
00	IOTD	I/O Transfer and Disconnect.Transfer the number of words specified in the word count from/to main memory to/from the next physical record of the device.
01	IOTP	I/O Transfer and Proceed.Transfer the number of words specified in the word count from/to main memory, then continue to the next DCW using the same physical record of the device.
10	TDCW	Transfer to DCW.The Executive will not allow use of this code.
11	IONTP	I/O No Transfer and Proceed.Do not transfer the number of words specified. Proceed to the next DCW. Data read will be discarded, data written will be zero.

The valid drives of this type are:

4100nn (CR) Multi-record Read. Reads nn records from the device.

420001 (DS) Read Track Header. Reads the track header of the current track.

43x001	(DS) Format Track. Formats the current track and writes the track header. The TI (track indicator) is set to x, which must be 0, 1, 2, or 3.
6100nn	(PR) Multi-record Write. Writes nn records to the device.
620001	(PR) Write Image. Loads the print train image.
630001	(PR) Write VFC. Loads the vertical format control.
640001	(MP) Load Control Store. Loads a microprogram into the MPC writeable control store.
650001	(MP) Load Main Memory. Writes the MPC main memory.
660001	(MP) Load Personality. Writes the MPC's personality overlay for a set of ports.
670001	(MP) Read Detail Status. Reads the detail status registers.

In addition, the following three codes are used internally by the Executive and may not be issued by a slave job, except indirectly by COPY commands:

400000	Read (all devices except CP,PR,MP)
600000	Write (all except CR,MP)
700000	Diagnostics (all devices)

6.3 Device Status Returns

This section describes the status returns a slave job can expect to get from a device file, on either COPY-type or DRIVE commands.

The Major Status, which is four bits, numerically coded, is given in octal. The Minor Status (Substatus) is six bits, bit coded. Thus it is possible to get several Minor Status returns in one status. The Minor Status is given here in binary, where "x" means either 0 or 1.

More detailed explanations of device status returns can be found in the Honeywell Information Systems publication I/O Programming, order number DB86.

6.3.1 Console

Major Status	Minor Status	Meaning
00		channel ready
02		device attention
03		data alert
	000001	transfer timing alert
	0x0010	transmission parity alert
	000100	operator input alert
	001000	operator distracted
	0100x0	incorrect format
	100000	message length alert
05		command rejected

6.3.2 Card Reader

Major Status	Minor Status	Meaning
00		channel ready
	000001	51 column cards
02		device attention
	000000	offline
	xxx0x1	hopper/stacker alert
	xxx01x	manual halt
	0x10xx	feed alert
	x1x0xx	card jam
	1x00xx	read alert
	1x1xxx	sneak feed
03		data alert
	000001	transfer timing alert
	000x10	validity alert
	0001x0	dual read failure
	001000	no read command
04		end of file
	000000	logical end of file
05		command rejected
12		MPC device attention
	001000	invalid punch
13		MPC device data alert
15		MPC command reject
	001000	device reserved

6.3.3 Card Punch

Major Status	Minor Status	Meaning
00		channel ready
02		device attention
	000000	offline
	0xxxx1	hopper/stacker alert
	0xxx1x	manual halt
	0xx1xx	chad box full
	0x1xxx	feed failure
	01xxxx	card jam
03		data alert
	000000	transfer timing alert
	000xx1	transfer timing alert
	000x1x	transmission parity alert
	0001xx	punch alert
05		command rejected
12		MPC device attention
13		MPC device data alert
	000110	PSI data overflow
15		MPC command reject
	001001	device reserved

6.3.4 Printer

Major Status	Minor Status	Meaning
00		channel ready
	000000	no button
	000001	print one line
	000010	forward space
	000011	forward to top of page
	000100	invalid line
	000101	reverse/rewind
	000110	backspace
	000111	backspace to top of page
02		device attention
	000000	power fault
	000001	out of paper
	000010	manual halt
	000100	VFU tape alert
	001000	check
03		data alert
	000000	invalid character code/image buffer alert
	000001	transfer timing alert
	000010	alert before printing
	000100	alert after printing
	001000	paper low
	010000	slew alert
	100000	top of page echo
05		command reject
	000000	VFC not loaded
	000001	invalid op code
	000010	invalid device code
	001000	train image not loaded
	010000	feed alert on last slew
	100000	top of page echo on last slew
12		MPC device attention
13		MPC device data alert
15		MPC command reject

6.3.5 Tape

Major Status	Minor Status	Meaning
00		channel ready
	xx0xx1	no write ring
	000x1x	at load point
	xxx1xx	nine track
	001100	ASCII alert
	010x0x	two-bit fill
	100x0x	four-bit fill
	110x0x	six-bit fill
01		device busy
	000001	in rewind
	000010	alternate channel in control
	000100	loading
	100000	device reserved
02		device attention
	00xx01	no write ring
	000010	no such handler
	0xx10x	handler in standby
	0x1x0x	handler in check
	01xx00	blank tape on write
03		data alert
	000001	transfer timing alert
	000010	blank tape on read
	xxxx11	bit detected during erase
	xxx1xx	transmission parity alert
	xx1xxx	lateral parity alert
	x1xxxx	longitudinal parity alert
	1xxxxx	end of tape foil
04		end of file
	001111	seven-track EOF
	010011	nine-track EOF
	111111	data alert
	xxxxxx	EOF character
05		command reject
	000000	invalid density
	xxxxx1	invalid operation code

	xxxx1x	invalid device code
	xxx1xx	parity error on operation/device code
	001000	tape at load point
	010000	attempted read after write
	100000	nine-track alert
07		channel timeout
10		channel busy
12		MPC device attention
	000001	configuration switch error
	000010	multiple devices
	000011	illegal device ID number
	001000	incompatible mode
	0011xx	TCA malfunction
	010000	MTH malfunction
	010001	multiple BOT
13		MPC device data alert
	000001	transmission parity alert
	000010	inconsistent command
	000011	sum check error
	000100	byte locked out
	001000	PE-burst write error
	001001	preamble error
	010000	multiple track error
	010001	skew error
	010010	postamble error
	010011	NRZI CCC error
	010100	code alert
	100000	marginal capstan speed
15		MPC command reject

6.3.6 Disk

Major Status	Minor Status	Meaning
00		channel ready
	0000xx	automatic retries
	0010xx	in T&D
	010000	EDAC correction
01		device busy
	000000	file positioning
	100000	alternate channel in control
02		device attention
	0000x1	write inhibit
	00001x	seek incomplete
	001000	inoperable
	010000	in standby
	100000	offline
03		data alert
	000001	transfer timing alert
	000010	transmission parity alert
	000100	invalid seek address
	0x1000	header verification failure
	x1x000	check character alert
	1x0000	data compare alert
04		end of file
	000000	good track detected
	0000x1	last consecutive block
	00001x	sector count limit
	000100	defective track - alternate assigned
	001000	defective track - no alternate assigned
	010000	alternate track detected
05		command reject
	001000	invalid command sequence
10		channel busy

13		MPC device data alert
	000001	transmission parity alert
	000010	inconsistent instruction
	000011	sum check error
	000100	byte locked out
	001110	EDAC parity error
	010010	nonstandard sector size
	010001	sector size error
	010011	search alert on first search
	010100	cyclic code error - not first search
	010101	search alert - not first search
	010110	sync byte error
	010111	error in automatic alternate track
	011001	EDAC - last sector
	011010	EDAC - not last sector
	011011	EDAC - block count limit
	011100	EDAC - uncorrectable error
	011101	EDAC - short block

6.3.7 Microprogrammed Peripheral Controller

Major Status	Minor Status	Meaning
00		channel ready
12		MPC device attention
	000001	configuration error
	000011	device number error
	000010	multiple devices
	001011	CA OPI down
	001100	alert EN1 unexpected interrupt
	001101	CA EN1 error
	001110	CA alert - no interrupt
13		MPC device data alert
	000001	transmission parity
	000010	inconsistent command
	000011	checksum error
	000100	byte locked out
15		MPC command reject
	000001	illegal procedure
	000010	illegal logical channel number
	000011	illegal logical channel to suspend controller
	000100	continue bit not set

6.3.8 716 Front End Processor

Major Status	Minor Status	Meaning
00		channel ready
03		data alert
	000001	transfer timing error
	000010	improper buffer size
	000100	transmission parity error
05		command reject

6.4 Simulated Status Returns

In addition to status returns generated by the IOM (sync bit equals 1), the following statuses will be simulated by Physical I/O.

Status	Word	Status
000000	000000	Ready. Simulate after a successful set decimal or set binary drive.
000B00	000300	Button status. Occurs only on line printer. B is the button number. See Section 6.5.
010000	000400	Device Busy. Occurs only on diagnostic drives with queuing bypass option specified.
030400	000400	Invalid address. The device address for the file is invalid. Usually indicates a hardware or Executive error.
050100	000400	Invalid operation. The code is invalid for the device. Note that all codes are invalid for nonexistent devices.
051000	000300	Invalid sequence. Occurs only on write commands to the Datanet-30. The Datanet-30 has issued a write-initiate, and the 635 slave program programmer must respond with a read-type command.
070000	000500	Channel Timeout. An operation did not complete in a reasonable amount of time. This indicates a hardware or Executive failure.
070100	000500	Special Timeout. Occurs on await-special and await-ready drives, and on reads to the Datanet-30. Indicates an excessive amount of time spent waiting for appropriate conditions (device

		ready, special interrupt, or write-initiate from the Datanet-30)
100000	000400	Channel Busy. All channels to the device have been released.
200000	000400	Power off. The device is not configured.

6.5 Special Programming Considerations

Datanet-30

A write on a Datanet-30 file causes a write to be issued immediately. However, if a read is issued, Physical I/O delays until a special interrupt is received from the Datanet-30 channel, indicating the the Datanet-30 has issued a write, before issuing the read. If no special interrupt occurs in a reasonable amount of time, the read command is rejected with the simulated special interrupt timeout status.

High-Speed Printer

The user of the printer should be aware of the fact that top-of-page echo statuses are returned directly from Physical I/O — the Executive does not retry operations with such statuses. Await-special, await-ready, and multirecord write drives may be issued to the printer.

If the operator presses one of the printer buttons, the printer returns device-attention/manual-halt status to subsequent write commands. When the OPERATE button is pressed, the printer rejects the next command with button status (cf. previous section). Thus upon encountering a device-attention/manual-halt status, the preferred programming technique is to issue an await-ready command and inspect the status on the subsequent trap to determine the button number. The button numbers and labels are:

1 - PRINT ONE LINE
2 - FORWARD SPACE
3 - FORWARD SPACE TOP OF PAGE
4 - INVALID LINE
5 - REVERSE/REWIND
6 - BACKSPACE
7 - BACKSPACE TOP OF PAGE

A copy-type command generates a device command for the printer designed to print edited and slew one line after printing. Note that the slewing may be overridden by the appropriate use of escape and formatting characters.

Disk

The file pointer on a disk is the logical record number, not a physical record (sector) number. On a DSS190-type disk there are two physical records in a logical record. On a MSU450-type disk there are four physical records in a logical record.

The FORMAT TRACK commands (MODE 43x001) format the track containing the current sector. The Track Indicator bits are specified in the MODE and not in the file pointer.

Card Punch

The card punch has the property that a card is not checked for punch errors until the subsequent card is being punched. Thus, a slave job using the card punch should have the ability to repunch not only the current card, but also the previous card for error recovery in case of punch alerts.

In decimal mode, cards are punched in the nonedit mode.

Card Reader

The card reader can be operated in either binary or mixed mode. Each read issued on the device file causes one card to be read, except in the case of multirecord read drive (see DRIVE MME). An end-of-file return (return code of 2) will be generated by Physical I/O if an attempt to read a card is made when the hopper is empty and the LAST BATCH light is turned on. (Note that the last batch substatus is also returned on the trap of the READ command for the last card.)

Magnetic Tape Handler

DTSS

Note that it is the user's responsibility to detect the end-of-tape foil status when writing tape. Note also that any single character record will be treated as a file mark, but that octal 17 is the standard end-of-file mark on a 7-track tape, and octal 23 is on a 9-track tape.

Await-ready and await-special drives may be issued to tape handlers. Note, however, that since all tape handlers may be connected to the same ICM channel, an await-special drive may trap whenever a hardware special interrupt is generated by <u>any</u> tape handler. Thus the recommended procedure for mounting a tape is to issue a rewind-standby drive, type a message to mount the proper tape, and then issue an await-ready drive.

Console Typewriter

Note that unlike the printer, the typewriter does not halt data transfer on receiving an end-of-line sequence (escape character followed by a character of the form 0XXXXX). Thus, more than one line can be written to the typewriter in a single WRITE. A carriage return is issued automatically after successful input; the user does not need to take this responsibility.

MPC

The Reset operation on an MPC (MODE 070000) normally results in a timeout. In this case a zero status is returned by Physical I/O.

6.6 Executive Error Recovery

If Executive error recovery is enabled, Physical I/O will attempt standard error recovery. The device status will be returned to the user only when the operation has succeeded, or if error recovery has failed.

If a device status indicates a condition of interest to the user, such as Top-of-page Echo on a printer, then no error recovery is attempted and the status is returned to the user.

Standard error recovery includes retrying timeout status and all ICM errors (nonzero ICM Channel Staus or ICM Central Status).

Error recovery for the console includes printing "DELETED" and retrying operator input alert and operator distracted status. "LINE TOO LONG—RETYPE IT" is printed in response to a message length alert, and the operation is retried.

A data alert on tape is retried with backspace-reread or backspace-erase-write operations. The reread operations are each tried up to eight times by the MPC. Up to seven reread operations are issued, with varying capstan speeds and thresholds. Thus, a read operation may be tried up to 64 times.

Tape movement errors are retried only if they returned an initiate status, that is, if the tape did not move.

Chapter 7

Shared Files

Files may be open in one of two modes, normal and shared access. In normal mode, permissions are granted to various jobs such that only one job may have a file open with append permission; or if a job has a file open with write permission, then only that job may access the file at all.

In shared access mode, however, any number of jobs may simultaneously hold any combination of read and write permission. This mode allows separate jobs to update a common data base without closing and reopening the file between accesses. However, if any job has a file open in one mode (normal or shared), it may not be opened by another job in the other mode.

To open a file in shared mode, see TALLY OPEN.

A file which is to be accessed in shared mode must be saved with the 'shared' bit turned on in the unpassworded accesses. (The shared bit is '040000' which is the same as the Owner bit for catalogs.) The bit will be ignored if it appears in the passworded accesses; when it appears on the unpassworded side it applies to opening the file with or without a password. This means there is no way to protect the permission to open a file in shared mode differently than the permission to open it in normal mode. Anyone who can open a file in shared mode can also open it in normal mode. Having Owner permission on a catalog does not override the lack of the shared bit on a file in that catalog.

COPY-type operations to a file open in shared mode are queued and executed in the order in which they are received, except that reads may overlap each other. A job may lock the file specifying that no other jobs' operations may complete until the file is unlocked, allowing that job to be sure that no changes will be made to the file for a certain interval and that no one will read inconsistent data. If another job has the file locked, the operation locking the file will not complete until the file is unlocked by the other job.

Also note that shared mode files may act as a general inter-job semaphore. For example, a job may open a certain file in share mode, lock it to seize the semaphore and unlock it to release the semaphore.

All COPY operations will proceed normally in the order in which they are issued unless the Reserve bit is set in the flag bits in X4 ('200000' for source, '000200' for destination) of one of the operations. When this bit is specified, the file will be 'locked' by or reserved to that job until the same job issues another copy operation without the bit set, or a RESET STATUS. While the file is reserved, other jobs' operations will be queued and their execution will be delayed indefinitely until after the file is released. A COPY operation involving a file open in shared mode may specify only the job's core as the other end of the COPY. Thus file-to-file or file-to-communications file copies are not allowed.

Chapter 8

Tally Operations

The "Fetching" catalog operations (OPEN, ERASE, REPLACE) provide two important utilities:

(1) Allowing a job to access files in a catalog not accessible to the job, and

(2) Providing a method to OPEN (ERASE, REPLACE) a file anywhere in the file system in one, indivisible, operation.

These operations also unfortunately have two disadvantages:

(1) The format of a treename is very inflexible, and not amenable to simple user interfaces, and

(2) There are several "special conventions" which all jobs must therefore know about and (hopefully) handle identically.

The "Tally" operations have been created to provide all the power of fetching operations with as much flexibility as possible. These operations use a standard treename format (defined in Section 8.2) so that all jobs can take advantage of this format without having to know anything about it. By creating these tally operations, all scanning and parsing is centralized. A job simply passes a tally (hence the name) which describes a string of ASCII characters to the Executive, and the Executive performs all necessary parsing on the treename.

8.1 General Parameters

The Tally operations are all built around one set of scanning routines in the Executive; because of this, they all have similar parameters. The only difference in calling parameters is in the specification of accesses.

The general formats for the tally operations are:

X0 Contains the file reference number of the initial catalog to search

X1 Contains a pointer to the tally word for the treename

X3 Contains the file reference number of the "alternate initial catalog".

This catalog is used as the initial catalog to search when one of the three special conventions is used for the first name in the treename (see Section 8.2).

X4 Has one of three formats:

(1) For TALLY ERASE and TALLY REPLACE:

Bit 4 is the escape convention bit (see below). Bits 9-17 are the trap bit mask to be used during the Open phase of the operation.

(2) For TALLY OPEN:

Bits 0-3, 5-8 are the accesses with which to open the file/catalog. Bit 4 is the escape convention bit (see below). Bits 9-17 are the trap bit mask.

(3) For TALLY CATALOG:

The register points to two words. The upper half of the first word is the same as (1) above, the lower half is zero. The second word is the access word with which to save the file/catalog.

X5 Is a pointer to two words (not neccessarily on an even-word boundary). This two word pair will contain the name of the file/catalog last scanned by the Executive (e.g. if the operation was successful, the last name in the treename), except:

(1) If the status was 12 (format error) or 40 or greater, the two words are undefined; or

(2) If the status was 7 (fetch error) the two words are zero.

The "Escape Convention" bit (020000 octal) is provided to control the use of the three special conventions explained in Section 8.2. If this bit is zero, the Open phase of the operation will proceed as defined in Section 8.2. If this bit is one, then any attempt to use one of the special conventions will trap the operation with a 12.7 status (see Section 8.3).

8.2 Scanner Conventions

The conventions to be used by the scanner are the following: A treename consists of one or more file/catalog names seperated by colons. Any of the names may have a password specified; the password is located after the name, separated from the name by a comma. For instance,

CATALOG1:CATALOG2,PASSWORD:FILENAME

is a legal treename. A name or password consists of 8 characters. Any name/password less than 8 characters will be left justified and space filled to form a full 8 character quantity. In the past it was possible to catalog files with non-standard names or passwords; a name/password in this case may be specified by a slash ("/") followed by up to 24 octal digits. These digits are then right justified and null filled to form a 72 bit quantity. To avoid future problems with non-standard treenames, any attempt to TALLY CATALOG or TALLY REPLACE a file/catalog with a name/password specified in octal will trap the operation with a 12.10 status (see Section 8.3).

Legal file/catalog name characters are:

(1) all uppercase characters

(2) all lowercase characters, which shall be converted to uppercase characters

(3) all digits

(4) and "-" and "." as special characters.

Legal password characters are:

(1) all legal filename characters

(2) all control characters (oct 1-37).

Any character which is not legal where found delimits the tree name string (all nulls are ignored), and will trap the operation with a 12.5 status (see Section 8.3).

There are three special conventions for the first name of a treename. Use of a special convention when the escape special convention bit is on in the access word (see Section 8.1) will trap the operation with a 12.7 status (see Section 8.3).

The special conventions are:

(1) If the first name begins with a colon (:), the file or catalog is to be found in the MFD catalog specified by X3. A first name of ":MFD" is equivalent to a first word octal name of "/777777777777777777777777" in the catalog specified by X3.

(2) If the first name begins with a star ("*"), and is a legal user number format the scanner will expand the name to open a user catalog in the catalog "MUD" in the catalog specified by X3. If it is not a legal user number format, the operation will trap with a 12.2 status (see Section 8.3).

Legal user number formats are:

```
*C          (*K — super group name)
*LLC        (*HDK — super group name)
*CDD        (*KOO — group name)
*LLCDD      (*HDKOO — group name)
*CDDDDC     (*40849L — individual user number)
*LLCDDDDC   (*HDKOOOOO — individual user number)
```

where:

C = character which is a letter or digit
L = a letter
D = a digit

(3) If the first name contains a trailing "***", the file or catalog is to be found in "DLIBRARY" in the catalog specified by X3.

Treenames are limited to 36 names. Any attempt to use a treename consisting of more than 36 catalog names (including any names prefixed by the scanner) will trap the operation with a 12.4 status (see Section 8.3).

The following is a formal BNF definition of a treename:

```
<treename>  ::=  <first name>:<names>
              |  <first name>[<space> ... ]

<first name>  ::=  <name>
                |  :<name>
                |  <filename> *** [,<password>]
                |  *<:MUD name> [,<password>]

<names>  ::=  <name>:<names>
           |  <name>[<space> ... ]

<name>  ::=  <filename>[,<password>]

<filename>  ::=  {<filename character>}
              |  /{<octal digit>}

<filename character>  ::=  <C>
                        |  <special>

<password>  ::=  {<password character>}
              |  /{<octal digit>}

<password character>  ::=  <C>
                        |  <special>
                        |  <control character>

<:MUD name>  ::=  <super group>
               |  <super group><D><D>
               |  <super group><D><D><D><D><C>

<super group>  ::=  <L><L><C>
                 |  <C>

<C>  ::=  <L>
       |  <D>
<L>  ::=  {A|B|C|D|E|F|G|H|I|J|K|L|M|N|
              O|P|Q|R|S|T|U|V|W|X|Y|Z}
       |  {a|b|c|d|e|f|g|h|i|j|k|l|m|n|
              o|p|q|r|s|t|u|v|w|x|y|z}
<D>  ::=  {0|1|2|3|4|5|6|7|8|9}

<space>  ::=  {b̸}
```

<special> ::= {.|-}

<octal digit> ::= {0|1|2|3|4|5|6|7}

<control character> ::= {<SOH>|<STX>|<ETX>|<EOT>|<ENQ>|
<ACK>|<BEL>
| <BS>|<HT>|<LF>|<VT>|<FF>|<CR>|
<SO>|<SI>|<DLE>
| <DC1>|<DC2>|<DC3>|<DC4>|<NAK>|
<SYN>|<ETB>
| <CAN>|<EM>|<SUB>|<ESC>|<FS>|
<GS>|<RS>|<US>|}

8.3 Status Returns

If the operation was successful, then bits 9-17 of status word 1 will be zero. The two words pointed to by X5 will contain the name of the file/catalog on which the operation was performed.

If the operation was unsuccessful, then status word 1 contains the reason for the failure. If the status is 12 or 40 or greater, the two words pointed to by X5 are undefined. If the status was 7 (fetch error) then the two words are cleared. Otherwise, the last file name scanned is placed in the two words.

If the status returned from the operation is 12 (format error), then status word 2 will contain a tally count and character position in the lower half. When this lower half quantity is concatenated with the starting address of the string (the upper half of the initial tally word), the resulting tally will run out at the character where the format error was detected. In the upper half of status word 2 one of the following substatuses will be returned:

(1) Name too long: cannot be more than 8 characters nor more than 24 digits following a / in the "/" octal convention.

(2) Illegal place for asterisk: an asterisk was found, and it was not the first character of the

treename nor a set of 3 at the end of the first name in the treename.

(3) Illegal format for name in ":MUD": the name following the leading "*" convention was not one of the proper formats.

(4) Too many levels: the treename (perhaps expanded by the scanner) contains more than 36 catalog names, and is therefore too long.

(5) Illegal character: a delimiter was found before the tally ran out. A delimiter is any character which is not legal where it is found (e.g. a non-octal digit in a "/" octal specification).

(6) Illegal format: every character in the treename was null, there was a null name, or there were no digits following a "/".

(7) Illegal convention: a special convention was used, but the escape convention bit was set disallowing all special conventions.

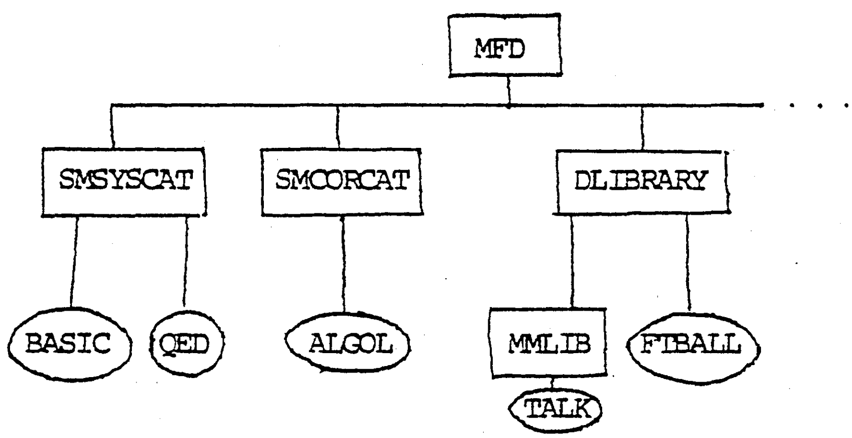
(10) Illegal use of "/" format: the final name or password in a TALLY CATALOG or a TALLY REPLACE cannot be specified using the "/" convention.

Chapter 9

Catalog Structure

9.1 Introduction

This chapter describes some of the basic notions and subtle points of the catalog system. Catalogs are essentially ways of organizing access to information which is external to any running slave program. The catalog structure is generally drawn as a tree, with the Master File Directory (MFD) as the root, with files and catalogs contained in the MFD, with files and subcatalogs in turn saved in these subcatalogs and so on. Thus we see:



This chapter answers such questions as: what kinds of things can be files? What kind of information is kept about files? What do catalogs look like? How do storage quotas work?

This chapter is divided into three parts:

(1) The physical organization of files

(2) Catalog information accessible to slave jobs

(3) Quotas and storage limitations

9.2 Definitions

A catalog is a collection of descriptors. Each descriptor describes some sort of object; the object may be a disk file, a catalog, a peripheral device such as a card reader, a core file, or one of several other things.

Each descriptor may be broken into two halves: one half contains logical (device-independent) information, and the other specifies where the object is stored by a list of device addresses (DA's). In general, the logical information may be viewed and/or directly modified by slave programs, while the device addresses may not be.

There are only two types of descriptors: catalog descriptors and "file" descriptors. They are pictured below:

Catalog Descriptor

logical	8 words
DA's	1 word

File Descriptor

logical	8 words
DA's	0 to 27 words

Both types have 8 words of logical information. Catalog descriptors include exactly 1 word for DA's, while file descriptors may have from 0 to 27 DA's apiece.

The DA in a catalog descriptor points to the first part of the catalog it describes; this first part contains a catalog header which describes storage for that catalog. A catalog header looks like this:

Catalog Header

logical	8 words
DA's	12 words

Objects may be either cataloged or scratch. An object is cataloged if a descriptor for it is included in some catalog; otherwise it is a scratch object. Once an object is cataloged, it may not be cataloged again, so objects created by slave programs are guaranteed to have only one descriptor apiece.

9.3 Physical Organization

9.3.1 Device Addresses (DA's)

A device address is a full word quantity specifying where part or all of an object can be found. The format of a normal DA is:

0	1 7	8 11	12 17	18 35
0	EXP	X	DC	RN

EXP - log base 2 of the number of contiguous records in DA

X - (not used)

DC - logical Device Code (logical device number in ENV deck)

RN - Record Number of first record in DA

(Although we refer to the DA as containing physical addresses, in fact the device code and record number are internally mapped by the Executive into the true physical addresses; but this mapping does not concern us and we can consider these fields to be "true" physical addresses.)

The device code (DC) is the most important part of a DA, for it specifies what kind of machinery the object is. Most objects are disk files or catalogs, so the device code reflects which disk drive their data is stored on. Peripherals such as printers, tape drives, card readers, and so on, all have their own device codes.

The device code is the only part of an object's description which specifies the physical type of the file. Since most slave jobs cannot look at DA's they cannot determine an object's physical type (whether it is a peripheral device, etc.) from its description. They must open it and perform operations on it to deduce its type.

The EXP (size) field is only used for disk files and catalogs. It contains the log base 2 of the number of words in this DA, some or all of which are used to hold a chunk of the file or catalog. When the Executive allocates room for a file/catalog on a disk, it finds chunks of storage big enough to hold all the data. Each of these chunks is some power of 2 words long. The DA list, if read in order, contains the addresses of each chunk of the file in the proper order.

If data is appended to the end of the file/catalog, the remaining unused space in the last DA is filled. When the last DA is completely filled, a new DA is allocated and appended to the end of the descriptor's DA list. Then data is written into that DA's space.

Note that since file descriptors may hold no more than 27 DA's, files' sizes may be limited even though there is enough disk space to hold more data. The Executive's algorithm for allocating various sized DA's is designed to minimize this problem, but often has the side effect of reserving too much space for files.

This space is periodically regained by the Shuffle module which opens non-optimally allocated files, moves them into preallocated scratch files, and then replaces the old files with the scratch files. When Shuffle opens each scratch file, it can tell the Executive how long the scratch file will be so that the Executive may optimally allocate DA's for it and thus avoid wasted space.

The DA for a mass storage device file has the format:

0	1 7	8 11	12 17	18 35
1	AT	X	DC	0

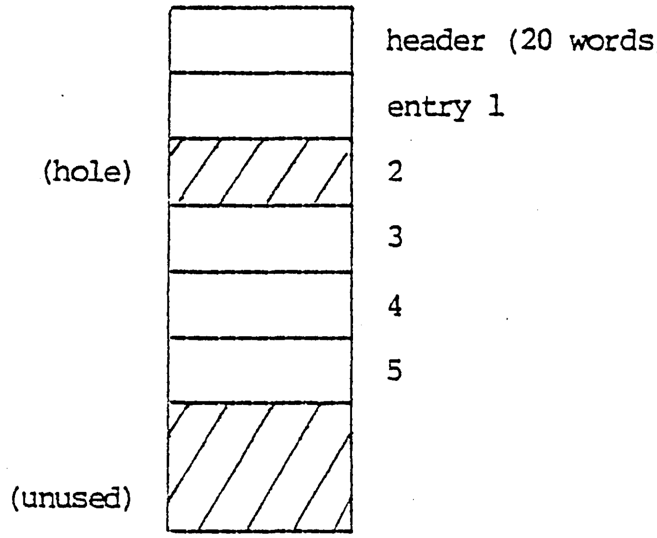
AT - Allocation Type (specifies logical to physical record mapping scheme)

DC - Logical Device Code (specifies logical to physical record mapping scheme)

9.3.2 Catalogs

A catalog is, as we have said, a list of descriptors of objects. Catalogs are always allocated on disks, and the 12 DA's in catalog descriptors serve as addresses to the various chunks of the catalog. (Naturally, not all 12 of the DA's need be used.) Two special catalogs, the MFD and SMCORCAT, are allocated in core.

Storage management within catalogs works as follows. Assume the following picture of a catalog:



"Holes" are created whenever objects are unsaved from catalogs. Holes are simple unused descriptors. When a new descriptor is added to a catalog, the Executive searches from the beginning of the catalog to find the first hole big enough to put the descriptor in. If there are no holes big enough, the descriptor is added to the end of the catalog and the catalog's length is updated.

Roughly the same thing happens when a file expands and needs a new DA in its descriptor. The descriptor is moved to the first hole big enough, or is added to the end of the catalog.

Adjacent holes are never coalesced in a new, bigger hole. This is because the Executive maintains a table of open files, and this table contains the <u>entry number</u> of each file in its catalog. Entry numbers must include holes, or removing a descriptor from a catalog would change the entry numbers of an unknown number of open files in this table. For the same reason, holes may not be coalesced, or the entry numbers for all following descriptors would change.

Note that entries (holes and descriptors) are essentially maintained on a sequential list. Thus searches for a descriptor must proceed linearly through all entries until it is found.

Finally, catalogs may contain no more than 4095 entries (holes and descriptors), due to the field size allocated for entry numbers in the Executive's table of all open files internal file table.

Subcatalogs may be nested within catalogs an arbitrary number of levels deep (subject only to rules on storage quotas).

9.4 Slave Accessible Information

9.4.1 Catalog Headers

Slave jobs may view a portion of each descriptor in a catalog, and may obtain information about the catalog itself. The information they may see includes information which governs access to the file, file identifiers, and so forth. This information is summarized below, in the forms in which it can be obtained by the "READ CATALOG" MME.

The zeroth entry returned on a READ CATALOG MME contains information about the catalog itself. This information is as below:

0	MAX		
1	ALOC		
2	Ø		
3	Ø		
4	ACC	PREF	TYPE
5	ENTRIES		
6	Ø		
7	LEN		

PREF: bits 29-31;
TYPE: bits 32-35

MAX

The maximum length which the catalog may hold. This maximum and its effects are described in detail in the section on catalog max's.

ALOC

The current length of all objects contained in the catalog plus 20 words for the catalog header, plus 12 words for each file descriptor in the catalog. (Twelve is somewhat arbitrary, since file descriptors may range from 8 to 35 words long. It is chosen as a good average length.)

Every catalog contains a 20 word header describing the catalog, so the minimum ALOC for a catalog is 20 words.

The length of each file in the catalog is added into the ALOC; but each subcatalog has its MAX added into the ALOC. This means that changes to the lengths of files in subcatalogs do not affect the lengths of all catalogs containing the subcatalog.

ACC

The accesses with which the slave job has this catalog open.

PREF

The catalog's preference. See the discussion of preferences below, under file descriptors.

TYPE

The file's type. Described below for file Descriptors.

ENTRIES

The number of entries (descriptors and holes) in the catalog. Jobs which want to read all the descriptors in a catalog should issue a Read Catalog MME for the number of entries, plus one. This guarantees that they will get them all.

LEN

The length of the catalog itself: 20 word header, descriptors, and holes.

9.4.2 Logical Descriptors

Information about all objects saved in a catalog may be gained by applying a READ CATALOG MME to the catalog. The MME returns an 8 word block for each descriptor; this descriptor is not in the same format as the Executive maintains it internally, and does not provide the DA's in each descriptor.

The DA's may be obtained by opening each object and applying a "PROVIDE DEVICE ADDRESS" MME to the object. Jobs may only issue this MME if they are Load-Dump enabled, and so most jobs may not see an object's DA's. Since DA's are the only way to determine peripheral types, etc., most jobs may not readily discover whether or not a given catalog contains peripherals.

The descriptors (as returned by the READ CATALOG MME) look like this:

0	NAME		
1			
2	PASSWORD		
3			
4	ACC		
5	DAYS-USED	PREF	TYPE
6	DLU	DLM	
7	LEN or MAX		

PREF: bits 29-31;
TYPE: bits 32-35

NAME

The name is a unique identifier for this descriptor, and hence for the object it represents. No two descriptors in a catalog may have the same name, and the name is the only identification for a descriptor.

PASSWORD

If a job supplies a password when trying to open (or erase, etc.) a file, the password is compared with this field, and no accesses are allowed if the password doesn't match. If the access word has the slave-trapped bit set, then this field contains the filename of a slave trap program (in the same catalog) to be run whenever some job attempts to open (erase, etc.) this file.

If the file is master-trapped, the password field is irrelevant and may be used for arbitrary data. In any case, this field is returned as zero unless the job issuing the Read Catalog MME has owner permission on the catalog. See the publication "File Access and Protection" for help.

ACC

This word contains access bits, which are used to determine how various jobs, may handle the object associated with this descriptor. See TM102 ("File Accesses") for an exhausting discussion.

DAYS-USED

The DAYS-USED field contains a count of how many different days the object associated with the descriptor has been used since it was last modified. A file is "used" when it is opened. The day when the object was modified is not included in this count.

Catalogs are never "used" but are modified by changing their MAX's.

PREF

An object's preference determines which storage class its storage is allocated from. In general, lower$preference files are saved so they are faster to access, but this is not an inherent property of preferences. They are simply a means of partitioning storage. The various preferences and how they are used are:

0 - allocated in core

1 - swap storage

2 - monitor scratch files, and files in :SMCORCAT

3 - catalogs

4 - normal scratch files

5 - special data base

6 - normal saved files

7 - permanent data bases

In general, if a file needs storage of some preference and none is available, it will be allocated storage from the next higher preference which has room.

Special permissions are required for jobs to open scratch files with preferences other than 4, or cataloged files with preferences other than 5 or 6. These permissions are given to a job by the job which runs it, which must in turn have the permissions.

All catalogs created by slave jobs are preference 3, no matter how the jobs create them. The Executive creates some core catalogs with preference 0.

An object's type to a large extent determines what can be done to it. Only the types marked with a star below are possible for cataloged files:

0	regular scratch file
1	regular scratch catalog
*2	regular cataloged file
*3	regular cataloged catalog
4	special scratch file
5	special scratch catalog
*6	special cataloged file (device file)
*7	special cataloged catalog
10	communications file (slave)
11	communications file (master)
12	running job
13	nonexistant file (closed communications file)
*14	migrated file
16	shared mode file

The difference between regular and special files/catalogs is that the DA's for regular files/catalogs are drawn from a system-wide pool, and storage for these files forms a part of the file system. Special files/catalogs have their own pre-allocated DA's which point to storage which is dedicated to these files or catalogs. Only the Executive can create special files/catalogs. Many operations such as ERASE, REPLACE, SCRATCH are not allowed on special files/catalogs.

At present, the MFD is the only specially-cataloged catalog. This, and the fact that its DLM is 00/00/00, serve as its only identification.

Peripherals are saved as special cataloged files, since their DA's point to the devices themselves and are not suitable for use as file storage. The Executive makes a special check for special cataloged files, and does not update their lengths when data is written to them. (Thus tape drives do not get immensely long, for example.) Also, special cataloged files may not be unsaved or erased. Off-line descriptions contain no DA's and are fairly intractable, since the s they represent have moved to tape.

DLU

This field contains a coded form of the date last used. An object is "used" when it is opened, replaced, etc. The coding format is the same as for the DLM. Catalogs are never "used" in this sense.

DLM

This field contains the coded form of the date the object was last modified. An object is modified when it is cataloged, its length changes, or when data is written to it. The DLM of a catalog is modified only when its max changes.

```
bits  0 -  8 - year modulo 100
      9 - 12 - month (January = 1, December = 12)
     13 - 17 - day of the month
```

LENGTH (or MAX)

If the descriptor is for a file, this field contains the file's length in words. If for a catalog, this field contains the catalog's MAX as described in the previous section.

9.5 Quotas and Storage Limitations

9.5.1 Catalog quotas

A catalog's quotas are determined by the MAX word in its header. A catalog's MAX does not place any upper bound on its ALOC (length of its included files and subcatalogs); rather it limits the accesses which jobs may obtain upon opening a file when the ALOC>MAX. That is, if a job can open a file with append permission, it can write as much data into the file as it wants, limited only by the job's quotas (discussed in the next section).

Three cases are notable for the relations between ALOC and MAX.

(1) ALOC<MAX. A file may not be replaced if its new length would push the allocated length over the catalog's maximum. But if a job can open a file with append permission, the amount of data which it can append is limited only by its job quota.

(2) ALOC>MAX. Jobs may never get append permission on any objects within the catalog. Objects, however, may be replaced with objects of smaller lengths, or erased. (But not scratched, since scratching requires append permission!)

(3) ALOC>(2*MAX). Files may not be opened. They may, however, be erased or replaced with smaller files.

Jobs with special permissions may open scratch catalogs with infinite max's. Infinite max's are flagged with the sign bit in the MAX field set; quota checks are suspended for infinite max catalogs. Such catalogs may only be cataloged within other infinite max catalogs, which then have the sign bit set in the ALOC field to signal an infinite length catalog.

Note that scratch catalogs with infinite maxes may be cataloged within themselves, but may not then be cataloged elsewhere (since already cataloged).

9.5.2 Job quotas

To prevent jobs from running amok and filling all system storage with unnecessary data by appending to files, each job has a quota of how many words it may add to cataloged or scratch storage.

Index

Indexed words are given in their immediate context, i.e. as much to either side of the word as will fit on a line. Underlining has been omitted, all series of three or more spaces have been reduced to two, and each end-of-line has been made a space. The line number is calculated from the top of the page: the header line is always line 7, the trailer is always line 60, and line 11 is usually the first line of text.

Brackets are used to enclose entries for which the same wording will not be found in the text; no line number is given for these entries.

page	line	word
47	23	ter. SQUEEZE causes the BAR for the issuing job to be
10	26	e base address register (BAR) limits memory references
24	31	relative to the squeezed BAR, and has bit 35 (the squee
10		[BAR = base address register]
10	23	as noted here. 1.1.1 Base Address Register The ba
47	22	gram by setting a pseudo base address register. SQUEEZ
156	23	ape special convention bit X5 Pointer to two words
231	33	The "Escape Convention" bit (020000 octal) is provided
182	50	n the communication file bit (bit 1) in the access to p
87	14	d. (4) If the return bit (bit 2) is set in the acce
100	39	g to open. If the fetch bit (bit 4) of X4 is 1, then X
101	47	presence of the fetch bit (bit 4). If the fetch bit
157	47	he presence of the fetch bit (bit 4). If the fetch bit
230	42	is the escape convention bit (see below). Bits 9-17 a
185	33	ing the appropriate flag bit in an index register. B$R
178	21	ing the appropriate flag bit in X4 (see COPY MME). In
102	35	catalog, and the fetch bit is on in the access word o
64	25	d to it. If the return bit is set in the access word
48	19	losed and, if the return bit is set in their access, re
152	23	ention, (bits 9-17) trap bit mask X5 Pointer to two w
162	23	ention, (bits 9-17) trap bit mask X5 Pointer to two w
81	23	zero (bits 9-17) trap bit mask X5 Reserved for fut
131	22	e zero; (bits 9-17) trap bit mask X5 Reserved for fut
124	22	or OPEN (bits 0-8); Trap bit mask (bits 9-17) X5 Poin
156	22	cesses on the file, trap bit mask and escape special
148	38	(MME 500136). The trap bit mask for the search is in
101	11	l be limited by the trap bit mask in bits 9-17 of X4.
230	43	Bits 9-17 are the trap bit mask to be used during t
92	48	t possess the Log enable bit may not issue this MME. A
55	36	must not have the return bit set in its access word.
87	28	log which has the return bit set in its access word mus
183	24	assed without the return bit set in the accesses, it be
163	43	uing job with the return bit set. On completion of th
132	45	uing job with the return bit set. Upon completion of
227	41	saved with the 'shared' bit turned on in the unpasswor
112	11	atch file and the return bit was not set in the access
184	54	ing the appropriate flag bit when issuing those command
148	40	], the escape convention bit) is set in the upper half
152	41	1, the escape convention bit) is set in X4, then all sp
156	49	1, the escape convention bit) is set in X4, then all sp
162	46	1, the escape convention bit) is set in X4, then all sp
24	32	has bit 35 (the squeeze bit) set on to indicate this f
11	46	y the use of the inhibit bit. This hardware feature pr
68	24	of device file X4 Flag bits X6 Trap location X7 P

page	line		word
74			[break special]
183	18	cesses need not include	"break" (eXecute) permission; i
180	44	g specials" (7). (2)	"Break" drive: A drive of type
180	55	nd which has eXecute (	"break") permission on its end
175	54	ommunication file is not	busy (see Section 5.5), and us
44	39	. This file will remain	busy as long as the PURE comma
188	41	ly, the slave end may be	busy because of a copy-type op
87	23	y or which are currently	busy cannot be passed to the n
75	19	mmunication file that is	busy due to the action of anot
186	24	eing led astray. 5.5	Busy States and Reset Status
187	31	le, the file will appear	busy to the slave end, and onl
141	42	use the master end to be	busy until a RESET STATUS is i
88	28	ully spawned job remains	busy until a trap occurs to th
112	26	he return bit set become	busy until they are closed and
186	32	The "communication file	busy" status (6) is a special
187	26	usy ("communication file	busy") to all other slave ends
188	18	("communi- cations file	busy") to this end. Only loca
189	51	; the master end becomes	busy, but the state of the fil
69	23	he communication file is	busy, either because another e
116	45	he communication file is	busy, either because another e
138	34	he communication file is	busy, either because another e
144	29	he communication file is	busy, either because another e
146	30	he communication file is	busy, either because another e
167	33	he communication file is	busy, either because another e
171	47	com- munication file is	busy, either because another e
187	44	ued when the file is not	busy. (There is one exception
183	30	set, it does not become	busy. Instead, the passing job
13	15	de for a valid Executive	call are always 101 to reduce
12	17	3 MME Faults; Executive	Calls A slave job may reques
12	38	nt interrupt. Trapping	calls also function as transpa
14	49	mally indicate Executive	calls. However, if the job ge
223	40	t in the file pointer.	Card Punch The card punch ha
224	11	the nonedit mode.	Card Reader The card reader
55	11	MME 500103:	CATALOG X0 File reference n
120	11	MME 500114: READ	CATALOG X0 File reference n
148	11	MME 500141: TALLY	CATALOG X0 File reference n
96	11	MME 500125: OLD READ	CATALOG AND OPEN FILES X0 F
96	32	command is like the READ	CATALOG AND OPEN FILES command
114	37	n them by issuing a READ	CATALOG AND OPEN FILES command
124	11	MME 500127: READ	CATALOG AND OPEN FILES X0 F
124	33	ication words The READ	CATALOG AND OPEN FILES command
86	15	(quotas exceeded). The	catalog and scratch word allot
59	11	MME 500111: CHANGE	CATALOG ENTRY X0 File refer
62	11	MME 500124: CHANGE	CATALOG MAX X0 File referen

page	line		word
141	32	running below it until a	CONTINUE command is issued.
109	45	d the supra job's RUN or	CONTINUE command is trapped wi
42	42	job in X4 of the RUN or	CONTINUE command which initiat
48	29	the trap for the RUN or	CONTINUE command which initiat
66	56	be restarted by another	CONTINUE command. If the job
88	40	be continued by using a	CONTINUE command. If, however
88	43	f a recoverable error, a	CONTINUE may not be issued. I
29	35	(3) On RUN, EXECUTE and	CONTINUE, gives termination
14	28	b then has the option of	continuing or terminating the
156	23	ask and escape special	convention bit X5 Pointer to
230	42	: Bit 4 is the escape	convention bit (see below).
148	40	0000 [octal], the escape	convention bit) is set in the
152	41	020000 octal, the escape	convention bit) is set in X4,
156	49	020000 octal, the escape	convention bit) is set in X4,
162	46	020000 octal, the escape	convention bit) is set in X4,
231	33	are zero. The "Escape	Convention" bit (020000 octal)
152	23	(bit 4) escape special	convention, (bits 9-17) trap b
162	23	(bit 4) escape special	convention, (bits 9-17) trap b
148	45	e specified with the "/"	convention, and the catalog wi
162	51	e specified with the "/"	convention, and the replace wi
231	42	on 8.3). 8.2 Scanner	Conventions The conventions
152	42	hen all special scanning	conventions are disallowed, an
156	50	hen all special scanning	conventions are disallowed, an
162	47	hen all special scanning	conventions are disallowed, an
232	48	There are three special	conventions for the first name
230	34	one of the three special	conventions is used for the fi
229	40	ere are several "special	conventions" which all jobs
232		ial first name = special	convention] [spec
44	52	le for pure procedure.	Copies to and from the pure re
22	14	length. (See slave end	COPY commands, Section 5.1.)
68	11	permission. MME 500131:	COPY X0 File reference numb
194	17	1" (0). 5.6.5 Aborted	COPY and Break Drive Suppose
193	32	ion Job D now issues a	COPY command from its storage
190	34	occurs when a slave end	COPY command has been issued,
117	42	as a special case of the	COPY command in which X1 and X
172	55	as a special case of the	COPY command in which both X0
193	29	End 2 (lowest) 5.6.4	COPY Command with No-Trap Opti
177	51	issue the corresponding	COPY command, unless it resets
138	30	slave end via a WRITE or	COPY command. In other words,
175	26	e calls, principally the	COPY command. Other calls whi
175	49	end. 5.1.1 Slave End	COPY Commands A COPY comman
177	11	s. 5.1.2 Master End	COPY Commands As noted above
178	36	Section 5.6. 5.2 Non-	COPY Commands Communication
22	42	location pointer (see	COPY MME). 1.4.9 Simultaneo

page	line		word	
180	11	if it desires. 5.2.3	DRIVE	(Master End) Data tran
181	27	ed to that job. 5.2.4	DRIVE	(Slave End) Certain dr
181	32	ive (type 0) or a device	drive	(type 12) may be issued.
21	50	. 12 D Slave issued	drive	- notifies the master
74		[set mode	drive	= set mode special]
19	29	which are generated by a	DRIVE	command at the master en
180	17	munication file, and the	DRIVE	command can be used for
200	47	e called directly by the	DRIVE	command in order to requ
201	22	n a single device action	DRIVE	command this word is ret
199	44	m the device, or for the	DRIVE	command to activate spec
199	28	ram may wish to use, the	DRIVE	command with function co
194	23	be given. It issues the	DRIVE	command. The drive is r
185	11	a COPY, READ, WRITE, or	DRIVE	command. The master end
181	41	ut and output modes (see	DRIVE	MME). The exact interpr
20	13	pecials generated by a	DRIVE	on a communication file,
20	23	ecial was generated by a	DRIVE	or CLOSE on a communic
194	22	b A decides that a break	drive	should be given. It iss
181		[slave-issued	drive	special]
183	19	be able to issue a break	drive	successfully. Once cre
207	41	shes rewinding. 6.2.2	Drive	with DCW's A Drive wit
77	31	ce was timed out. If a	DRIVE	with DCW's fails with a
200	41	of data. If a copy or	drive	with DCW's is successful
74	16	r to Data Control Words	(Drive	with DCW's only) X4 Fl
199	29	le device action) or 24	(drive	with DCW's) is provided
76	20	e device action) and 24	(drive	with DCW's). A drive of
29	38	of infra job. (4) On	DRIVE	with DCW's, gives record
181	35	et mode and slave-issued	drive)	at the job holding the
189	34	master end job may issue	DRIVE,	CLOSE, or REQUEST STATU
190	44	ued an operation: either	DRIVE,	on its own initiative,
175	28	communication files are	DRIVE,	PASS, CLOSE, REQUEST ST
195	43	read communication file"	drive,	the special interrupt w
188	28	a drive-type operation	(DRIVE,	TRUNCATE, SCRATCH, or S
20	56	ed a break drive. (See	DRIVE-master	end, Section 5.2.
21	54	command (MODE). (See	DRIVE-slave	end.) 13 R S
190	20	a slave end has issued a	drive-type	command but the spe
180	27	Read communication file"	drive:	A drive of type 1 (and
180	44	als" (7). (2) "Break"	drive:	A drive of type 3 (and
186	14	nly exception is a break	drive:	in this case, if the r
204	31	nsferred. 6.2 Device	Drives	A number of device op
205	11	6.2.1 Single Action	Drives	A single action drive
204	36	e drives. Single action	drives	change the state of the
204	37	he state of the device.	Drives	with DCW's transfer dat
74		[set mode	drive]	
74		[read comm file	drive]	

page	line	word
167		[slave end issued TRUNCATE special]
144	26	terrupt number 14 (slave end issued TRUNCATE) at the ma
167	30	terrupt number 14 (slave end issued TRUNCATE) at the ma
69		[slave end issued WRITE special]
171		[slave end issued WRITE special]
69	19	ssued READ) or 12 (slave end issued WRITE) at the maste
171	42	terrupt number 12 (slave end issued WRITE) at the maste
69	25	s reserved it, the slave end issuing the COPY command w
22	12	- notifies the master end job that a read-type comma
22	29	- notifies the master end job that a TRUNCATE or SCR
144	35	be issued on the master end of a communi- cation file.
19	30	VE command at the master end of a communication file ar
64	34	ppeared. If the master end of a communication file is
69	17	A COPY issued at a slave end of a communication file wi
74	43	rive types at the master end of a communication file ar
116	39	READ issued at the slave end of a communication file wi
137	32	a job file or the slave end of a communication file, t
138	23	ence number of the slave end of a communication file, t
144	25	mand issued at the slave end of a communication file wi
144	32	If the job at the master end of a communication file is
146	26	mand issued at the slave end of a communication file wi
167	29	mand issued at the slave end of a communication file wi
171	41	RITE issued at the slave end of a communication file wi
75	44	y be issued on the slave end of a terminal communicatio
74	38	pectively, at the master end of the communication file.
138	28	s the duty of the master end of the communication file
146	28	T POINTER) at the master end of the file. The second i
22	33	. (See TRUNCATE-slave end only, Section 5.2.) 17
87	12	be created and its slave end passed. (4) If the ret
194	39	ervation. 5.6.6 Slave End Reservation; Local Operati
184		[slave end reserve]
184		[master end reserve]
185	21	: (1) If the master end resets status on the commu
187	15	lave end. 5.5.1 Slave End States First, the file a
189	21	e again. 5.5.2 Master End States The situation is
64	35	is closed, or if a slave end to which the communication
172	21	the corresponding slave end will be trapped only if
141	37	h the master and a slave end will cause status to be re
179	11	REQUEST STATUS (Master End) A REQUEST STATUS can be
179	35	2 REQUEST STATUS (Slave End) A REQUEST STATUS to the
181	27	b. 5.2.4 DRIVE (Slave End) Certain drive types may
181		[truncate (slave end)]
181		[scratch (slave end)]
181		[set pointer (slave end)]

page	line	word
231	33	words are zero. The "Escape Convention" bit (020000
152	22	8) must be zero, (bit 4) escape special convention, (
162	22	8) must be zero, (bit 4) escape special convention, (
84	11	MME 500117: EXECUTE X0 File reference n
29	35	opened. (3) On RUN, EXECUTE and CONTINUE, gives te
64	46	e the description of the EXECUTE command for a descript
106	50	he supra job on a RUN or EXECUTE command). A job can a
36	17	bles a job to receive an execute fault (real-time timer
15	23	r counts to zero, and an execute fault is given if the
15	48	1.3.5 Parity Faults; Execute Faults When a job ge
15	27	ardware timer runout nor execute faults are returned as
143	14	E is very similar to the EXECUTE MME (500117) and diffe
13	15	of the code for a valid Executive call are always 101
12	17	em. 1.2.3 MME Faults; Executive Calls A slave job
14	49	faults normally indicate Executive calls. However, if
225	19	by Physical I/O. 6.6 Executive Error Recovery If
36	17	ob to receive an execute fault (real-time timer runout
24	18	curred, then the connect fault (special interrupt) if a
46	18	o receive a timer runout fault after a specified amount
15	43	ection 1.4. The parity fault cell is not currently us
15	44	future use. The startup fault cell is not used by the
18	46	ord of the slave connect fault cell) is zero, a slave
15	34	The shutdown and connect fault cells have special uses
13	25	is divided into two-word fault cells, one for each of t
43	18	e real-time timer runout fault has occurred. If an N g
11	54	for too long, a lockup fault is generated. This faul
15	23	to zero, and an execute fault is given if the real-tim
15	22	s given. A timer runout fault is returned if the run-t
13	14	r out of bounds, the MME fault is returned to the slave
24	19	any, and then any slave fault other than simulated sla
14	37	lts Whenever a memory, fault tag, DRL, lockup, illega
22	49	cial interrupt, or slave fault to be given to a job eac
13	21	tive call. 1.3 Slave Fault Vector The first 40 (o
18	48	t takes place. No dirty fault vector aborts are gene
14	26	ro, the job had a "dirty fault vector" and is aborted;
15	25	o be aborted for a dirty fault vector. The simulated f
15	35	pecial uses in the slave fault vector. Word 0 (the fir
42	36	released, then a memory fault will occur. The amount
36	18	(real-time timer runout fault) after a specified amoun
24	29	cial interrupt, or slave fault, it is unsqueezed. The
14	15	faults. To give a slave fault, the Executive stores th
25	53	MME instruction unless a fault, trap, or special interr
47	33	e specified IC/IR. If a fault, trap, or special interr
25	39	and will result in a MME fault. (A job receives a MME

page	line	word
141	36	ssued on a communication file for which an operation is
144	28	). If the communication file is busy, either because a
64	34	r end of a communication file is closed, or if a slave
64	45	US. If a suspended job file is closed, then that job
84	40	or, which is on. A job file is created for the new jo
74	36	are 0 (set communication file mode) and 12 (single devi
200	24	the other must be a core file or a job file. In genera
74	27	only on a communication file or on a device file. Dri
200	32	file via a communication file should be swapped out of
192	42	job C receives a passed file special interrupt specify
48		[returned file special]
64		[returned file special]
74		[read comm file special]
184	14	are changed to a special file type called a "non-file"
199		[file type]
69	17	e end of a communication file will generate a special i
116	39	e end of a communication file will generate a special i
146	27	e end of a communication file will generate a special i
167	30	e end of a communication file will generate a special i
171	41	e end of a communication file will generate a special i
184	15	file type called a "non-file" (13), and the jobs holdi
195	43	es a "read communication file" drive, the special inter
180	27	(1) "Read communication file" drive: A drive of type
182	56	a slave end in a "passed file" special interrupt (see t
183	49	notified by a "returned file" special interrupt (if it
191	51	se it is 7) in a "passed file" special interrupt. The f
192	14	n receipt of the "passed file" special interrupt, suppo
197	19	b D receives a "returned file" special interrupt due to
246	33	escriptors. TYPE The file's type. Described below
111	31	terrupt number 4 (passed file) if that job is accepting
48	21	terrupt type 5 (returned file) is generated for each re
64	27	rrupt number 5 (returned file). If it is a scratch fil
199	25	ype 6 (special cataloged file). They are associated wi
64	52	rrupt number 5 (returned file). The job issuing the CLO
117	13	s a master communication file, the corresponding slav
69	44	d of a communica- tion file, and X1 is nonzero. The
66	49	tus are reset on the job file, or when the job terminat
117	20	munication or a shared file, that file is reserved fo
70	44	rce) file must be a core file, the issuing job's core,
20	24	CSE on a communication file, the lower half of the DA
138	24	e end of a communication file, then a special interrupt
172	50	ination file is a device file, then the lower half of s
136	54	pended or terminated job file, then the status informat
137	32	e end of a communication file, then the status informat

page	line		word
184	54	setting the appropriate	flag bit when issuing those co
68	24	mber of device file X4	Flag bits X6 Trap location
74	18	ve with DCW's only) X4	Flag bits X6 Trap location
116	18	memory location M2 X4	Flag bits X6 Trap location
141	16	ber of file/catalog X4	Flag bits X6 Trap location
171	18	e (may not be zero) X4	Flag bits X6 Trap location
228	25	eserve bit is set in the	flag bits in X4 ('200000' for
69	53	he master end job. The	flag bits in X4 are divided in
75	26	ith a status of 7. The	flag bits in X4 have the follo
116	53	ith a status of 7. The	flag bits in X4 have the follo
141	49	ed for that end). The	flag bits in X4 have the follo
172	11	h a status of 7. The	flag bits in X4 have the follo
193	54	by setting B$NTPD in the	flag bits, the first five word
193	37	by setting B$NTPS in the	flag bits. The first ten word
148	42	g will be trapped with a	format error if the treename c
152	43	e will be trapped with a	format error if the treename c
156	51	n will be trapped with a	format error if the treename c
162	48	e will be trapped with a	format error if the treename c
235	41	rom the operation is 12	(format error), then status wor
232			[format error]
224	24	t card.) Magnetic Tape	Handler Note that it is the
18	53	the end of its special	handling routine. 1.4.5 Sa
222	37	errupt timeout status.	High-Speed Printer The user
120	39	ferred. Note that since	holes can occur in catalogs an
125	32	ferred. Note that since	holes can occur in catalogs, a
243	34	unused) ————————	"Holes" are created whenever ob
246	39	entries (descriptors and	holes) in the catalog. Jobs wh
121	29	er of entries, including	holes, in catalog 6 Zero
126	20	er of entries, including	holes, in catalog 6 Zero
200	44	section, called Physical	I/O (PIO) will attempt to tran
225	22	ery is enabled, Physical	I/O will attempt standard erro
137	53	(q.v.). Lower: Unique	Identifier. This field can be
190	11	munication file are both	idle (from the master's point
193	46	ommunication file is now	idle again. Suppose job C sh
188	49	(20). Both ends become	idle, as does the communicatio
188	15	th, the slave end may be	idle, but the file may be rese
190	55	though the master end is	idle, no operation except CLOS
191	52	terrupt. The file is now	idle. 5.6.2 REQUEST STATU
187	18	d may both be completely	idle. In this state, any valid
196	26	file busy" (6). 5.6.8	Illegal Actions Suppose job
14	37	fault tag, DRL, lockup,	illegal procedure (IPR), incom
14	38	illegal procedure (IPR),	incomplete op (ONC), overflow/
60	19	contain the new usage	information for the file. The
55	45	mation or is zero. Type	information has the following

page	line	word
16	15	using traps and special interrupts. 1.4.1 Traps A
18		[bounced special interrupt]
18		[saved special interrupt]
14	38	ckup, illegal procedure (IPR), incomplete op (ONC), ove
21	50	ure use. 12 D Slave issued drive - notifies the ma
181		[slave-issued drive special]
181	35	upts (set mode and slave-issued drive) at the job holdi
22	11	e end.) 13 R Slave issued read - notifies the mas
69		[slave end issued READ special]
177		[slave issued READ special]
177	25	errupt indicating "slave issued READ" (a job requested
193	52	Job A receives a "slave-issued read" special interrupt
194	21	s notified with a "slave-issued read" special interrupt
116	40	upt number 11 (slave end issued READ) at the master end
192	18	special interrupt (slave-issued read) is generated at j
69	18	upt number 11 (slave end issued READ) or 12 (slave end
22	23	n 5.1.) 15 RS Slave issued request status - notifi
138		[slave end issued REQUEST STATUS special]
138	25	upt number 13 (slave end issued REQUEST STATUS) is gene
179	40	job issues it, a "slave issued request status" special
20	48	n 5.2.) 2 SRS Slave issued reset - notifies the ma
190	53	job has received a slave-issued reset special interrupt
188		[slave issued reset special]
141		[slave end issued RESET STATUS special]
141	41	pt number two (slave end issued RESET STATUS) at the ma
188	53	pecial interrupt ("slave issued reset") that its COPY i
22	35	n 5.2.) 17 SP Slave issued set pointer - notifies
146		[slave end issued SET POINTER special]
181	55	a SET POINTER, a "slave-issued set pointer" special is
146	28	upt number 15 (slave end issued SET POINTER) at the mas
144		[slave end issued TRUNCATE special]
167		[slave end issued TRUNCATE special]
181	54	the slave end, a "slave-issued truncate" special is ge
182	11	e end generates a "slave-issued truncate" special speci
144	26	upt number 14 (slave end issued TRUNCATE) at the master
167	31	upt number 14 (slave end issued TRUNCATE) at the master
22	17	on 5.1.) 14 W Slave issued write - notifies the ma
69		[slave end issued WRITE special]
171		[slave end issued WRITE special]
177	28	in response to a "slave issued WRITE" special, it shou
193	33	twenty words. A "slave-issued write" special interrup
69	19	d READ) or 12 (slave end issued WRITE) at the master en
171	42	upt number 12 (slave end issued WRITE) at the master en
84	30	t (in 1/64 milliCRU) Q Job access mask The source f

page	line	word
103	27	f the opened file or the MAX of the opened catalog in s
158	32	f the opened file or the MAX of the opened catalog in s
163	15	h of the scratch file or MAX of the scratch catalog is
106	20	rap location A Catalog MAX or preallocation length N
62	32	rrent ALOC. ALOC's and MAX's are described in Chapter
251	49	f the month LENGTH (or MAX) If the descriptor is fo
103	17	ds twice its permissible MAX. Finally, bit 0 of the ac
127	44	twice its permissible MAX; (6) Any remaining bi
42	36	mory is released, then a memory fault will occur. The
42	11	MME 500006: MEMORY REQUEST X5 Upper mem
10	32	ly when the job issues a MEMORY REQUEST call (on a MME
21	45	ock is released with a MEMORY REQUEST command before
84	52	spawned job can issue a MEMORY REQUEST to change the s
14	37	lave Faults Whenever a memory, fault tag, DRL, lockup
138	14	cataloged files. 4-N Message specified by PASS comm
111	40	a nonzero word, then the message specified can be read
87	32	by the PASS command, no messages accompany files passe
233		[MFD = Master File Directory]
242	34	wo special catalogs, the MFD and SMCORCAT, are allocate
233	12	og is to be found in the MFD catalog specified by X3.
251	11	ogs. At present, the MFD is the only specially-cata
124		] [migrated files = offline files
124	47	g read. Information for migrated files is provided onl
210	20	is given in octal. The Minor Status (Substatus) is si
76	50	r transferred). 6-11 Minor status. 12-17 Queue
202		[substatus = minor status]
202		[minor status]
143	14	y similar to the EXECUTE MME (500117) and differs only
12	21	xecutive by generating a MME (master mode entry) fault.
25	36	e command by executing a MME (master mode entry) instru
149	21	in the lower half. See MME CATALOG or Chapter 9 for m
13	13	e, or out of bounds, the MME fault is returned to the s
25	39	command will result in a MME fault. (A job receives a
14	49	xecuted in slave mode. MME faults normally indicate E
47	38	the Executive since all MME faults will be returned to
13	11	the address field of the MME instruction does not speci
25	45	d immediately by a PAUSE MME with a pause count of one.
25	46	d 200 series, or pausing MME's. The contents of the sl
181	42	output modes (see DRIVE MME). The exact interpretatio
14	42	ntire system (see ENABLE MME). This feature is to allow
60	21	to by X5 of the CATALOG MME. Upon completion of the
62	23	ses via the ALTER ACCESS MME. The previous MAX for the
10	16	1.1 Running in Slave Mode A job actually running
14	50	g the MME is in squeezed mode (entered by the Executive

page	line	word
233	19	), and is a legal user number format the scanner will
243	50	table contains the entry number of each file in its cat
120	36	used to specify an entry number. Upon completion of th
125	30	used to specify an entry number. Upon completion of th
232		[octal]
58	27	cifying a file type of off-line but the file has devi
56	12	ble file type, which is "off-line" (14 octal). Files t
251		[off-line]
124		[migrated files = offline files]
94	11	n. MME 500102: OLD ERASE X0 File reference
96	11	MME 500125: OLD READ CATALOG AND OPEN FILE
97	11	MME 500130: OLD REPLACE X0 File referen
14	38	re (IPR), incomplete op (CNC), overflow/truncate, or di
100	11	MME 500101: OPEN X0 File reference numb
156	11	MME 500136: TALLY OPEN XO File reference numb
152	36	permissions using TALLY OPEN (MME 500136). If the ope
148	38	file is made using TALLY OPEN (MME 500136). The trap b
96	11	5: OLD READ CATALOG AND OPEN FILES X0 File referenc
124	11	00127: READ CATALOG AND OPEN FILES X0 File referenc
124	33	s The READ CATALOG AND OPEN FILES command combines th
96	32	ike the READ CATALOG AND OPEN FILES command with two ex
38	50	in the READ CATALOG AND OPEN FILES command. When a j
114	37	suing a READ CATALOG AND OPEN FILES command. Status R
199	42	in the catalog tree (see OPEN MME). This file referenc
106	11	MME 500100: OPEN SCRATCH X4 File/catalo
88	25	tch file/catalogs (see OPEN SCRATCH) The job file f
29	31	the command. (2) On OPEN, ALTER ACCESSES, DUPLICAT
178	44	talog structure, such as OPEN, ERASE, CATALOG, or REPLA
229	22	ing" catalog operations (OPEN, ERASE, REPLACE) provide
81	49	the sequence of commands OPEN, UNCATALOG, and CLOSE.
153	13	quence of commands TALLY OPEN, UNCATALOG, and CLOSE.
227	37	n shared mode, see TALLY OPEN. A file which is to be
162	40	file/catalog using TALLY OPEN. If Read, Write, and App
230	46	ation. (2) For TALLY OPEN: Bits 0-3, 5-8 are the
53	24	ime the file/catalog was opened, and if they do not con
227		[TALLY OPEN]
189	49	ter end. The only local operation which can be issued
187	33	TUS functions as a local operation, with the additional
195	33	gal, since it is a local operation. Any nonlocal operat
194	39	e End Reservation; Local Operations Now consider what
184	49	cation file. Only local operations (see Section 5.5) c
185	46	slave ends. Only local operations can be issued at th
186	48	ends being busy. Local operations may be issued even
189	11	operations except local operations or CLOSE can be iss

page	line		word
85		] [scratch	quota = scratch word allotment
81	41	catalog entry. Catalog	quota checks are suspended for
152	50	see Chapter 8. Catalog	quota checks are suspended for
102		[catalog	quota checks]
131		[catalog	quota checks]
157		[catalog	quota checks]
162		[catalog	quota checks]
252	14	tations 9.5.1 Catalog	quotas A catalog's quotas ar
253	11	cataloged). 9.5.2 Job	quotas To prevent jobs from
132	36	ucceeds then his scratch	quotas are incremented by the
116	11	MME 500133:	READ X0 File reference numb
22	11	13 R Slave issued	read - notifies the master e
175	41	ommand and its variants,	READ and WRITE, are used to pe
120	11	MME 500114:	READ CATALOG X0 File refere
96	11	MME 500125: OLD	READ CATALOG AND OPEN FILES
96	31	ILES command is like the	READ CATALOG AND OPEN FILES co
114	37	obtain them by issuing a	READ CATALOG AND OPEN FILES co
124	11	MME 500127:	READ CATALOG AND OPEN FILES
124	33	entification words The	READ CATALOG AND OPEN FILES co
247	15	be gained by applying a	READ CATALOG MME to the catalo
244	38	can be obtained by the "	READ CATALOG" MME.
74		[	read comm file drive]
74		[	read comm file special]
138	32	in the same manner as a	READ command, except that the
179	52	ave end job had issued a	READ command. It may exercise
179	39	ntially a variant of the	READ command. When the slave
20	41	s the mode. 1 RCF	Read communication file - used
180	27	cation file are: (1) "	Read communication file" drive
195	43	If job A now issues a "	read communication file" drive
41	21	pausing after issuing a	READ on a terminal communicati
171	45	d issues a corresponding	READ or COPY. (See the descri
69	49	precede the slave issued	READ or slave issued WRITE spe
69		[slave end issued	READ special]
177		[slave issued	READ special]
194	20	ppose job D now issues a	READ to file reference number
177	25	indicating "slave issued	READ" (a job requested data tr
193	52	receives a "slave-issued	read" special interrupt. If j
194	21	ied with a "slave-issued	read" special interrupt. Howev
116	40	ber 11 (slave end issued	READ) at the master end. The
192	18	interrupt (slave-issued	read) is generated at job A, s
69	18	ber 11 (slave end issued	READ) or 12 (slave end issued
193	43	issues another ten-word	READ, not specifying the no-tr
185	11	nication file on a COPY,	READ, WRITE, or DRIVE command.
188	42	py-type operation (COPY,	READ, WRITE, or REQUEST STATUS

page	line	word
16	29	d of the trap block (the return word), and its IC is re
23	22	lustrates the use of the return word. EAX6 TRAP p
64		[returned closed special]
21	21	le was passed. 5 RF Returned file - notifies the j
48		[returned file special]
64		[returned file special]
183	49	ve end is notified by a "returned file" special interru
197	19	ning. Job D receives a "returned file" special interru
48	21	pecial interrupt type 5 (returned file) is generated fo
64	27	cial interrupt number 5 (returned file). If it is a sc
64	52	cial interrupt number 5 (returned file). The job issuin
94	31	eed to be zero. Status Returns 0 Successful: th
221	11	6.4 Simulated Status Returns In addition to statu
210	11	6.3 Device Status Returns This section describ
76		[device status returns]
199	30	. A typical example is "rewind" on a magnetic tape han
36	24	ssuing job happens to be road- blocked at this time it
43	22	is zero, the job will be roadblocked and will restart a
18	29	Otherwise, any pending roadblocks on the job are rele
29	20	rn word IC IR Trap Routine Access (bits 0-8
17	12	fourth word. This trap routine should end with an RET
18	53	f its special handling routine. 1.4.5 Saved Speci
143	11	MME 500120: RUN The RUN MME is very simi
38	53	index register X3 on the RUN command are ANDed with the
182	47	it PASS available in the RUN command). A communication
21	15	it PASS command in the RUN command. If DATA is nonze
84	22	allowed X5 Pointer to run list X6 Trap location X
86	22	. The remainder of the run list contains a list of fi
84	35	nd with the files in the run list open for it. The new
85	42	------------------+ The run list pointed to by X5 must
42	41	e supra job in X4 of the RUN or CONTINUE command which
109	45	rted and the supra job's RUN or CONTINUE command is tra
106	49	et by the supra job on a RUN or EXECUTE command). A jo
178	46	ication files cannot be "run". An attempt to do so wil
199	51	ile as a parameter for a RUN, ERASE, UNCATALOG, TRUNCAT
29	35	been opened. (3) On RUN, EXECUTE and CONTINUE, giv
15	15	lable to a slave job. A run-time timer is set by the E
45	11	memory. MME 500002: RUNNING TIME No arguments
46	17	a job to receive a timer runout fault after a specified
43	18	, or one real-time timer runout fault has occurred. If
15	22	fault is given. A timer runout fault is returned if th
36	18	e fault (real-time timer runout fault) after a specifie
11	35	ll connect faults, timer runout faults, and startup and
18		[saved special interrupt]

page	line	word
24	19	pt) if any, and then any slave fault other than simulat
22	49	p, special interrupt, or slave fault to be given to a j
13	21	Executive call. 1.3 Slave Fault Vector The first
15	35	have special uses in the slave fault vector. Word 0 (t
24	29	p, special interrupt, or slave fault, it is unsqueezed.
14	15	slave faults. To give a slave fault, the Executive sto
24	19	ult other than simulated slave faults. The job will th
13	29	hile others are used for slave faults. The position of
21	50	or future use. 12 D Slave issued drive - notifies
22	11	E-slave end.) 13 R Slave issued read - notifies t
177		[slave issued READ special]
177	25	al interrupt indicating "slave issued READ" (a job requ
22	23	Section 5.1.) 15 RS Slave issued request status -
179	40	ve end job issues it, a "slave issued request status" s
20	48	Section 5.2.) 2 SRS Slave issued reset - notifies
188		[slave issued reset special]
188	53	by a special interrupt ("slave issued reset") that its
22	35	Section 5.2.) 17 SP Slave issued set pointer - not
22	17	Section 5.1.) 14 W Slave issued write - notifies
177	28	larly, in response to a "slave issued WRITE" special, i
14	29	terminating the aborted slave job. This feature is in
10	16	tail. 1.1 Running in Slave Mode A job actually ru
9	23	ontrol of a processor in slave mode and may perform any
14	46	struction is executed in slave mode. MME faults norma
81	31	then 1 is assumed) AQ Slave trap identification word
94	22	X6 Trap location AQ Slave trap identification word
96	28	of words N to copy AQ Slave trap identification word
97	24	X6 Trap location AQ Slave trap identification word
100	28	f bit 4 of X4 is 1) AQ Slave trap identification word
124	30	of words N to copy AQ Slave trap identification word
131	30	then 1 is assumed) AQ Slave trap identification word
152	32	be zero) AQ Passed to slave trap program if file is
156	32	be zero) AQ Passed to slave trap program if file is
162	32	be zero) AQ Passed to slave trap program if file is
102	31	N-1)st catalog, then any slave trap program is bypassed
101		[slave trap program]
132		[slave trap program]
158		[slave trap program]
163		[slave trap program]
127	20	ile/catalog is master or slave trapped, then the acce
157	41	rrent file/catalog. Slave trapping programs are ru
181		[slave-issued drive special]
181	35	interrupts (set mode and slave-issued drive) at the job
193	52	ords. Job A receives a "slave-issued read" special int

page	line	word
201	43	is powered off. MAJOR STATUS (Bits 2-5) - This field
179	11	re. 5.2.1 REQUEST STATUS (Master End) A REQUES
179	35	s COPY. 5.2.2 REQUEST STATUS (Slave End) A REQUEST
210	20	ven in octal. The Minor Status (Substatus) is six bits
22	23	RS Slave issued request status - notifies the master
111	42	hed by issuing a REQUEST STATUS as the first file opera
190	55	except CLOSE or REQUEST STATUS can be issued to that e
16	34	us word always contain a status code indicating whether
186	54	he function of the RESET STATUS command is always to fo
69	33	copy by issuing a RESET STATUS command on the communic
88	51	ilable through a REQUEST STATUS command) is always zero
67	13	be verified by a REQUEST STATUS command), whereas the l
185	12	e file only with a RESET STATUS command. Once the fil
189	35	DRIVE, CLOSE, or REQUEST STATUS commands. A RESET STAT
192	11	idle. 5.6.2 REQUEST STATUS from a Slave End Upon
187	32	. In either case, RESET STATUS functions as a local op
190	56	that end before a RESET STATUS is issued. Other comma
138	19	mmand and if the REQUEST STATUS is the first file opera
191	18	d is destroyed. A RESET STATUS issued before the CLOSE
185	40	), except that a REQUEST STATUS issued to the master en
179	16	dinary file (see REQUEST STATUS MME). The type field o
20	50	end job issued a RESET STATUS on a COPY to which the
85	48	it by issuing a REQUEST STATUS on file reference numbe
185	21	If the master end resets status on the communication
177	52	ommand, unless it resets status on the master end (see
44	46	ro or by issuing a RESET STATUS on the source file for
184	19	any command except RESET STATUS or CLOSE. However, the
194	25	therefore issues a RESET STATUS requesting the communic
221	11	6.4 Simulated Status Returns In addition t
210	11	6.3 Device Status Returns This section
76		[device status returns]
138		slave end issued REQUEST STATUS special] [
141		[slave end issued RESET STATUS special]
197	15	te, job A issues a RESET STATUS to clear out job D's RE
22	24	r end job that a REQUEST STATUS was issued at a slave
190	25	issued in time, a RESET STATUS will prevent the specia
16	37	e description of the two status words is given in Secti
200		[device status word]
179	40	a "slave issued request status" special interrupt is g
141	41	(slave end issued RESET STATUS) at the master end and
138	25	slave end issued REQUEST STATUS) is generated at the ma
177	53	ster end (except REQUEST STATUS) will be rejected as "i
187	39	cal operations are RESET STATUS, PASS, and, if the end
20	52	responded. (See RESET STATUS, Section 5.2.) 3 BR

page	line		word
225	44	ried. A data alert on	tape is retried with backspace
75	44	ed on the slave end of a	terminal communication file.
180	47	typical use is with a	terminal communication file, t
181	40	communication file. For	terminal communication files i
48	11	. MME 500000:	TERMINATE X4 Termination ac
25	51	e case of a SQUEEZE or a	TERMINATE command, a slave job
65	12	e the description of the	TERMINATE command.) If a ter
64	46	nd all jobs below it are	terminated (see the descriptio
137	28	esources consumed by all	terminated infra jobs. If X0
88	49	CLOSE. (See CLOSE.) A	terminated job file can always
136	54	number of a suspended or	terminated job file, then the
88	41	d. If, however, the job	terminated, either successfull
66	56	NUE command. If the job	terminated, however, another C
66	49	ob file, or when the job	terminates or aborts. Status w
88	34	the job file, or the job	terminates or is aborted. If
14	29	option of continuing or	terminating the aborted slave
86	17	the spawning job. Upon	termination of the spawned job
40	11	MME 500012: JOB	TIME No arguments The tota
45	11	MME 500002: RUNNING	TIME No arguments The tota
49	11	MME 500003:	TIME OF DAY No arguments T
50	11	MME 500010:	TIME SINCE BOOTLOAD No argum
36	18	e an execute fault (real-time	timer runout fault) after
43	18	l interrupt, or one real-time	timer runout fault has oc
46	11	MME 500001: SET	TIMER A Time The SET TIME
15	15	a slave job. A run-time	timer is set by the Executive
15	16	real-time (elapsed time)	timer is set by the Executive
10	36	scribed below). 1.1.2	Timer Register The timer reg
46	17	ables a job to receive a	timer runout fault after a spe
43	18	errupt, or one real-time	timer runout fault has occurre
15	22	slave fault is given. A	timer runout fault is returned
36	18	execute fault (real-time	timer runout fault) after a sp
11	35	are all connect faults,	timer runout faults, and start
15	15	y the Executive call SET	TIMER, and a real-time (elapse
15	14	Two software-simulated	timers are available to a slav
36			[timer]
46			[timer]
11	27	e fault occurs. 1.2.2	Transparent Interrupts Certa
157	33	e returned by the master	trap program if the catalog
12	42	he action is complete, a	trap (described in Section 1.4
12	41	ping call must specify a	trap address in X6 at the time
16	23	or invalid parameter. A	trap address is specified when
152	23	convention, (bits 9-17)	trap bit mask X5 Pointer to
162	23	convention, (bits 9-17)	trap bit mask X5 Pointer to
81	23	st be zero (bits 9-17)	trap bit mask X5 Reserved fo

page	line	word
176	47	nd job exercises the "no-trap" option — see below). T
25	53	truction unless a fault, trap, or special interrupt was
47	34	fied IC/IR. If a fault, trap, or special interrupt occ
22	48	ssible for more than one trap, special interrupt, or sl
24	29	in Chapter 3) receives a trap, special interrupt, or sl
102	27	successfully run master trap, then all requested acc
23	17	do so. 1.4.10 Typical Trap-Handling Code (Non-Multip
127	20	talog is master or slave trapped, then the access bit
12	38	transparent interrupt. Trapping calls also function a
157	41	file/catalog. Slave trapping programs are run only
12	28	hree types: nontrapping, trapping, and trapping with au
9	46	truction that faulted traps indicating that asynchr
16	15	h the slave job by using traps and special interrupts.
24	14	e will link simultaneous traps and specials in such a w
17	23	will also encounter all traps before encountering any
43	17	o be roadblocked until N traps, one special interrupt,
81	29	X7 Number of entries in treename (if zero, then 1 is a
131	28	X7 Number of entries in treename (if zero, then 1 is a
233	45	s. Any attempt to use a treename consisting of more th
231	46	er are the following: A treename consists of one or mo
229	47	perations use a standard treename format (defined in Se
157	18	ucceeding catalog in the treename is found, it is searc
229	37	(1) The format of a treename is very inflexible, a
156	41	is a tally word for the treename of the file to be ope
100	32	filenames which form a "treename") is located in the f
162	36	m (denoted by a standard treename). The TALLY REPLACE
148	34	og denoted by a standard treename. First an attempt t
148		[standard treename]
152		[standard treename]
153		[treename]
156		[standard treename]
162		[treename]
233		[treename]
167	11	estroyed. MME 500107: TRUNCATE X0 File reference
175	28	ST STATUS, RESET STATUS, TRUNCATE (or SCRATCH), and SET
181		[truncate (slave end)]
144	45	as a special case of the TRUNCATE command in which the
22	29	master end job that a TRUNCATE or SCRATCH command
144		[slave end issued TRUNCATE special]
167		[slave end issued TRUNCATE special]
181	54	ave end, a "slave-issued truncate" special is generated
182	11	enerates a "slave-issued truncate" special specifying a
144	27	ber 14 (slave end issued TRUNCATE) at the master end.
167	31	ber 14 (slave end issued TRUNCATE) at the master end.

page	line	word
163	56	a REQUEST STATUS. These user access bits are obtained
163	56	a REQUEST STATUS. These user access bits are obtained
103	34	word 1 of the trap, nine user access bits can be obtain
132	22	word 1 of the trap, nine user access bits can be obtain
158	36	word 1 of the trap, nine user access bits can be obtain
163	55	word 1 of the trap, nine user access bits can be obtain
127	54	k. In addition, the nine user access bits from 9-17 of
233	19	("*"), and is a legal user number format the scanner
121		[DLU = date of last use]
13	21	all. 1.3 Slave Fault Vector The first 40 (octal)
18	48	s place. No dirty fault vector aborts are generated.
38	30	ission 2 Large State Vector permission (permits a s
14	26	e job had a "dirty fault vector" and is aborted; that i
85	15	enabled are Large State Vector, Priority Scheduling, C
15	25	borted for a dirty fault vector. The simulated faults
15	35	uses in the slave fault vector. Word 0 (the first wor
128	29	date, and the lower half-word contains the reel numbe
136	35	Job's remaining catalog word allotment 1 Job's rem
136	37	Job's remaining scratch word allotment 2 CRU limit
132	34	og first has its catalog word allotment decremented by
163	31	og first has its catalog word allotment decremented by
169	28	/catalog has its scratch word allotment decremented by
56	38	/catalog has its scratch word allotment incremented by
144	40	ve its scratch (catalog) word allotment incremented by
153	17	nd will have its catalog word allotment incremented by
56	39	he same time its catalog word allotment is decremented
70	52	job's scratch (catalog) word allotment is decremented
107	35	job's remaining scratch word allotment is decremented
149	27	header, and its catalog word allotment is decremented
149	25	ation, the job's scratch word allotment is incremented
169	29	he same time its catalog word allotment is incremented
48	22	ning scratch and catalog word allotments are added to t
86	15	The catalog and scratch word allotments for the spawne
64	54	its scratch and catalog word allotments incremented by
48		[scratch word allotment]
70		[catalog word allotment]
70		[scratch word allotment]
85		[catalog quota = catalog word allotment]
85		[scratch quota = scratch word allotment]
144		[scratch word allotment]
144		[catalog word allotment]
167		[scratch word allotment]
167		[catalog word allotment]
172		[scratch word allotment]

DTSS

Please use tape to seal this form.

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 21 HANOVER, NH

POSTAGE WILL BE PAID BY ADDRESSEE

Editor
DTSS Incorporated
10 Allen Street
Hanover, NH 03755

READER'S COMMENT FORM

We would appreciate receiving corrections, criticisms, and comments concerning this publication. Please include specific page and line references whenever possible. Your comments will help us improve future editions. DTSS Incorporated will be free to use or distribute information you supply without incurring any obligation whatever. If the title at top of page is not for the manual you are commenting on, please give the following information:

Title ____________________

Publication Number ____________ Date ____________

COMMENTS:

DTSS

Thank you for your cooperation. Please fold on dotted lines, tape, and mail. NO POSTAGE NECESSARY IF MAILED IN THE USA.

Optional — Name: ____________________

Phone: ____________________

Mailing Address:

Computer
Sharing
Services
Inc.

National Headquarters
2498 West Second Ave.
Denver, Colorado 80223
(303) 934-2381

Control and Non-Control Executive

Multiple Processor Support

- o In single processor configurations, the CPU executes both the Control and Non-Control EXEC.

- o If there are master mode tasks, the Control EXEC is executed. If not, the Non-Control EXEC is executed (next slave job is run).

- o In multiple processor configurations, one CPU is designated the Master CPU. All others are Slave CPUs.

- o The Master CPU operates as in single processor configurations, executing the Control or Non-Control EXEC when appropriate.

- o Slave CPUs only execute the Non-Control EXEC.

- o The Master CPU provides almost all master mode services. Note that I/O operations are handled only by the Master CPU.

- o The Control and Non-Control EXEC communicate through the Run Queue and Job Service Queue. The Control EXEC places jobs to be run by the Non-Control EXEC on the Run Queue. The Non-Control EXEC places jobs which require Control EXEC processing on the Job Service Queue.

- o Access to both queues is serialized by a semaphore. This allows asynchronous access by multiple, independent CPUs.

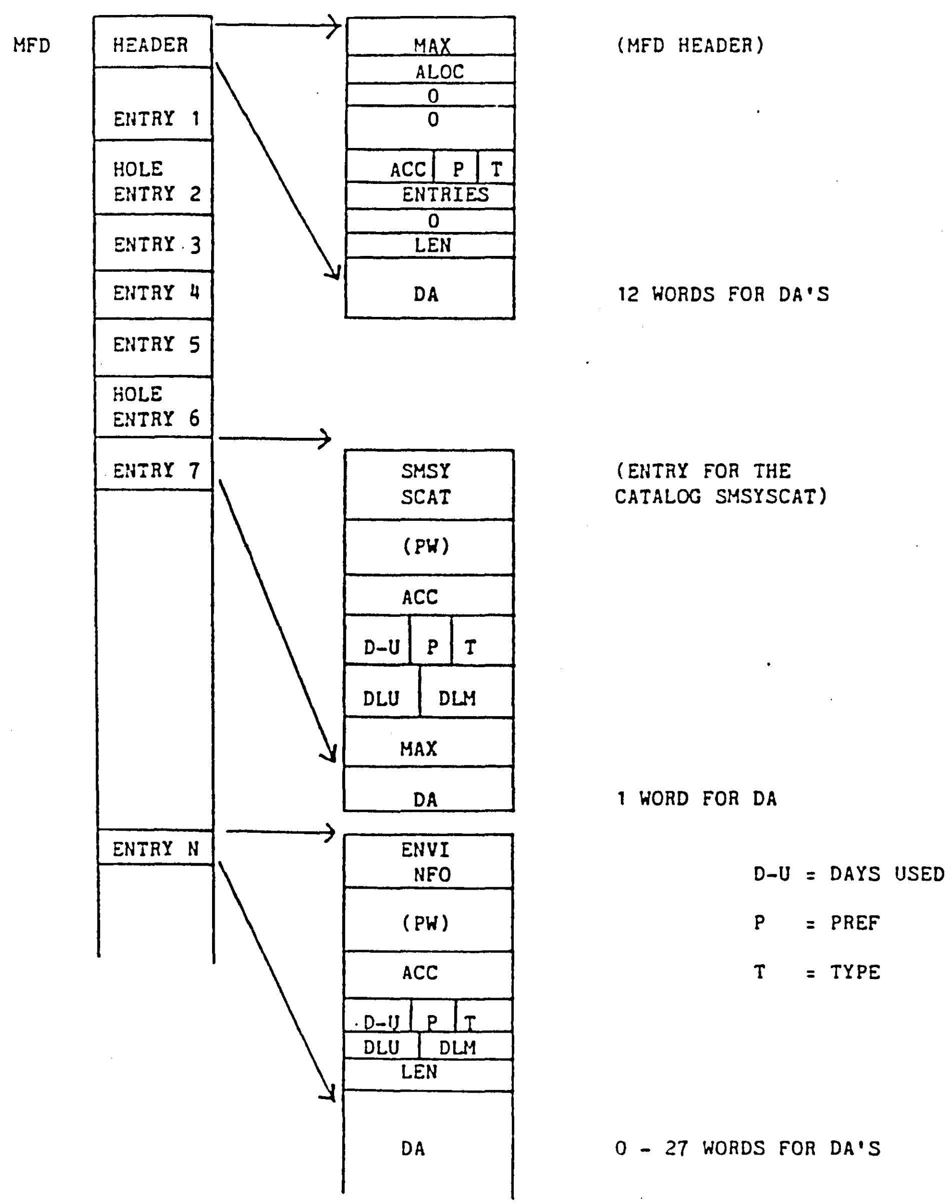
- o Since the Non-Control EXEC is re-entrant, it can be executed simultaneously by multiple CPUs.

Executive Segments

- INSERT
- LCORE
- CATALOG
- CONSOLE
- COPY
- DISKALC
- GLOBAL
- LISTALC
- LOG
- LOGFILE
- MMES
- NCONTROL
- PIO
- SWAPSKED
- UTIL
- DSTART
- TSTART
- BOOTCARD

Catalog Search Algorithm

Physical Layout of a Catalog



CONTROL AND NON-CONTROL EXECUTIVE MULTIPLE PROCESSOR SUPPORT