Honeywell

LEVEL 66

SOFTWARE

FILE MANAGEMENT SUPERVISOR

Honeywell

FILE MANAGEMENT SUPERVISOR ADDENDUM A

SERIES 60 (LEVEL 66)/6000

SOFTWARE

SUBJECT:

Additions and changes to the File Management Supervisor manual

SPECIAL INSTRUCTIONS:

This update, Order Number DD45A, is the first addendum to DD45, Revision 1, dated August 1976.

The attached pages are to be inserted into DD45, Revision 1, as indicated in the collating instructions on the back of this cover. Change bars in the margins indicate technical additions and changes, and asterisks indicate deletions. These changes will be incorporated into the next revision to this manual.

NOTE: This cover should be placed following the manual cover to indicate the updating of the document with Addendum A.

SOFTWARE SUPPORTED:

Series 60 Level 66 Software Release 4
Series 6000 Software Release J

DATE:

April 1977

ORDER NUMBER:

DD45A, Rev. 1

21187
3778
Printed in U.S.A.

COLLATING INSTRUCTIONS

To update this manual, remove old pages and insert new pages as follows:

Remove	Insert
vii, viii	vii, viii
5-3, 5-4	5-3, 5-4
5-13, 5-14	5-13, 5-14
8-1, 8-2	8-1, blank
	8-1.1, 8-2
9-1 through 9-6	9-1 through 9-6
9-11, 9-12	9-11, 9-12
9-17, 9-18	9-17, 9-18
10-1, 10-2	10-1, 10-2
10-9 through 10-12	10-9, through 10-12
10-15 through 10-22	10-15 through 10-22
10-25, 10-26	10-25, 10-26
11-21, 11-22	11-21, 11-22
11-25 through 11-28	11-25 through 11-28
12-3, 12-4	12-3, 12-4
12-13, 12-14	12-13, 12-14
12-21 through 12-24	12-21 through 12-24
B-5, B-6	B-5, B-6
B-9, B-10	B-9, blank
	B-9.1, B-10
B-13, B-14	B-13, blank
	B-13.1, B-14
B-19 through B-22	B-19, B-20
	B-21, blank
	B-21.1, B-22
B-27, B-28	B-27, B-28
E-1, E-2	E-1, E-2
F-1, F-2	F-1, F-2

File No.: 1733,1P33

SERIES 60 (LEVEL 66)/6000

SOFTWARE

FILE MANAGEMENT SUPERVISOR

SUBJECT

Description and Use of the File Management Supervisor

SPECIAL INSTRUCTIONS

This manual, Order No. DD45, Rev. 1, supersedes the previous edition dated May 1974, Order No. DD45, Rev. 0, and Addendum DD45A, dated May 1975. Change bars in the margins indicate new and changed information; asterisks denote deletions.

SOFTWARE SUPPORTED

Series 60 Level 66 Software Release 3
Series 6000 Software Release I

ORDER NUMBER

DD45, Rev. 1

August 1976

Honeywell

PREFACE

The facilities of the File Management Supervisor (FMS), a subsystem of the General Comprehensive Operating Supervisor (GCOS) are described in this manual. All of the services provided for in the subsystem known as the File System are included in the FMS, plus new services.

Use of the FMS involves other GCOS subsystems and numerous software programs. Users inexperienced in filing of data should accumulate manuals listed for non-FMS services in Section I for ready reference to topics that are peripheral to the FMS itself.

Series 60 Level 66 is hereafter referred to as the Series 60. Information in this manual refers to both the Series 6000 and Series 60, unless otherwise specifically stated.

File No: 1733,1P33

FUNCTIONAL LISTING OF PUBLICATIONS
for
SERIES 60 (LEVEL 66) and SERIES 6000 SYSTEMS

FUNCTION	APPLICABLE REFERENCE MANUAL	
	TITLE	ORDER NO.
	Series 60 (Level 66)/Series 6000:	
Hardware reference:		
Series 60 Level 66 System	Series 60 Level 66 Summary Description	DC64
Series 6000 System	Series 6000 Summary Description	DA48
DATANET 355 Processor	DATANET 355 Systems Manual	BS03
DATANET 6600 Processor	DATANET 6600 Systems Manual	DC88
Operating system:		
Basic Operating System	General Comprehensive Operating Supervisor (GCOS)	DD19
Job Control Language	Control Cards Reference Manual	DD31
Table Definitions	System Tables	DD14
Table Definitions	NPS Tables	DE34
I/O Via MME GEINOS	I/O Programming	DB82
System initialization:		
System Startup	System Startup	DD33
System Operation	System Operation Techniques	DD50
Communications System	GRTS/355 and GRTS/6600 Startup Procedures	DD05
Communications System	NPS Startup	DD51
Communications System	NPS Configuration Examples	DE76
DSS180 Subsystem Startup	DSS180 Startup (Series 6000 only)	DD34
Program Recovery	Program Recovery/Restart	DC98
Data management:		
File System	File Management Supervisor	DD45
Integrated Data Store (I-D-S)	I-D-S/I Programmer's Guide	DC52
Integrated Data Store (I-D-S)	I-D-S/I User's Guide	DC53
File Processing	Indexed Sequential Processor	DD38
File Input/Output	File and Record Control	DD07
File Input/Output	Unified File Access System (UFAS) (Series 60 only)	DC89
I-D-S Data Query System	I-D-S Data Query System Installation	DD47
I-D-S Data Query System	I-D-S Data Query System User's Guide	DD46
Coexistent I-D-S	Coexistent I-D-S Overview	DE60
Program maintenance:		
Object Program	Source and Object Library Editor	DD06
System Editing	System Library Editor	DD30
Test system:		
Online Test Program	Total Online Test System (TOLTS)	DD39
Test Descriptions	Total Online Test System (TOLTS) Test Pages	DD49
Error Analysis and Logging	Honeywell Error Analysis and Logging System (HEALS)	DD44
Language processors:		
Macro Assembly Language	Macro Assembler Program (GMAP)	DD08
COBOL-68 Language	COBOL Reference Manual	DD25
COBOL-68 Usage	COBOL User's Guide	DD26
Standard COBOL-68 Language	Standard COBOL-68 Reference Manual	DE17
Standard COBOL-68 Usage	Standard COBOL-68 User's Guide	DE18
JOVIAL Language	JOVIAL	DD23
FORTRAN Language	FORTRAN	DD02
Macro Assembly Language	DATANET 355/6600 Macro Assembler Program	DD01

FUNCTION	APPLICABLE REFERENCE MANUAL	
	TITLE	ORDER NO.
	Series 60 (Level 66)/Series 6000:	
Generators:		
Sorting	Sort/Merge Program	DD09
Merging	Sort/Merge Program	DD09
Simulators:		
DATANET 355/6600 Simulation	DATANET 355/6600 Simulator	DD32
Service and utility routines:		
Loader	General Loader	DD10
Utility Programs	Utility	DD12
Utility Programs	UTL2 Utility Routine (Series 60 only)	DC91
Media Conversion	Bulk Media Conversion	DD11
System Accounting	Summary Edit Program	DD24
FORTRAN	FORTRAN Subroutine Libraries	DD20
FNP Loader	DATANET 355/6600 Relocatable Loader	DD35
Service Routines	Service Routines	DD42
Software Debugging	Debug and Trace Routines	DD43
Time Sharing systems:		
Operating System	TSS General Information	DD22
System Programming	TSS Terminal/Batch Interface	DD21
System Programming	TSS System Programmer's Reference Manual	DD17
BASIC Language	Time Sharing BASIC	DD16
FORTRAN Language	FORTRAN	DD02
Text Editing	Time Sharing Text Editor	DD18
dataBASIC Language	dataBASIC System Language Manual	DD95
dataBASIC Loading	dataBASIC Load/Unload System	DD96
Remote communications:		
DATANET 30/305/355/6600 FNP	Remote Terminal Supervisor (GRTS)	DD40
DATANET 355/6600 FNP	Network Processing Supervisor (NPS)	DD48
DATANET 355/6600 FNP	NPS Micro-Ops Programming	DE35
DATANET 355/6600 FNP	Application Guidelines for NPS	DE77
DATANET 700 RNP	RNP/FNP Interface	DB92
Communication facilities:		
COBOL-74 Communications	Message Control System Site Manual	DC99
Transaction processing:		
User's Procedures	Transaction Processing System User's Guide	DD41
Handbooks:		
System-operator communication	System Console Messages	DD13
Error Messages, Abort Codes	Error Messages and Abort Codes	DC97
NPS Error Messages	NPS Error Messages	DE75
Pocket guides:		
Control Card Formats	Control Cards and Abort Codes	DD04
FORTRAN	FORTRAN Pocket Guide	DD82

CONTENTS

CONTENTS (cont)

TABLES

SECTION I

INTRODUCTION

FMS CLASSES OF SERVICES

The File Management Supervisor (FMS) performs a range of services for files of any type or size. Six classes of services are provided:

1. cataloging
2. control of mass storage space
3. prevention of unauthorized access
4. protection against device failure
5. protection against incomplete or incorrect update
6. protection against concurrent usage

FMS SERVICES

Cataloging is the method by which information concerning the file is kept and thus is the basis for all other services. Cataloging services allow files to be created and deleted and their descriptions to be grouped and modified.

The mass storage space control services allow space to be confined within prescribed limits for the user, for the user on a specific disk pack, or for file growth. Assignment of all files for a user, for a group of files, or of a specific file can be directed to a specific device or class of devices. No limits need be expressed nor device assignments made. The space used for each user's files and catalogs is measured and reported.

Prevention of unauthorized access depends on controlling access to a file by names and optional passwords, either of which can be changed. Moreover, access can be limited to a specific type of access (read, for example) or to a specific user (read by a specific user, for example). In addition, a file may be locked and unlocked to temporarily restrict access, with the locking being performed on command or automatically by the passage of time. Requests for access can be recorded, either for all requests or for only those denied for security reasons.

The three classes of protection services prevent damage to a file that may be due to four common causes of trouble: partial or complete failure of a device containing all or part of the file, abrupt termination of a job changing the file because of job or system failure, change to the file by a test job, and change to the file by several jobs running concurrently. These protection services are provided without any actions on the part of the operator running the jobs. Several options are provided for each of these classes to allow selection based on speed and sureness of protection versus space and time required to provide the protection.

FMS PROTECTION

Protection against device failure or change to the file by an aborted or testing job is provided by FMS in two fundamentally different ways. One way is to copy the entire file from time to time and to restore from the copy when the file is damaged by either device failure, test job change, or incomplete change by an aborted job. (In the time sharing environment, a file can be saved twice by use of two names in a SAVE or RESAVE statement.)

When the file is large, however, it is not practical to copy the entire file often enough to keep the file copy from being uselessly out-of-date. FMS can intervene on every file change, though, to copy the change either to an online duplicate or to a journal maintained offline, to protect against device failure, and to buffer the changes so they can be cancelled in case of job abort or test update.

Protection against concurrent change is similarly provided by FMS in both a way based on the entire file and one based on each change to the file. When the file contains records of interest to only a few users, it is practical for FMS to restrict use of the file so that only one job changes the file at a time and then when there are no users reading from the file. When the file contains records of interest to many users, however, it is sometimes more expedient to allow several jobs to change the file at the same time as other jobs are reading from the file and to restrict access only to those parts of the file in use by another job. Again, FMS intervention on each access to the file is the basis of this way of protecting against concurrent change.

The form of protection based on the entire file and not on intervention on each read or write should be used whenever it provides adequate protection. Files that are small or used only by a few users at a time rarely require protection based on FMS intervention.

NON-FMS PROTECTION

The three classes of protection are always provided in ways independent of file format, content, patterns of usage, etc. Only because of this independence can they be offered for any file. But this may limit their practical applicability for some files, and it is recognized that services by data management supervisors (like I-D-S for specially formatted files) or by application managers (for specially used files) may be used to provide more efficient and/or applicable protection against some kinds of damage to the file.

Sometimes it is possible to use one non-FMS service and one FMS service. For example, files can be saved by a program that uses format information to reduce the time and space required to save the file. Then this non-FMS save may be used in conjunction with FMS journaling to obtain an up-to-date copy of the file. On the other hand, when concurrent update is allowed but FMS control of concurrency is not used, the use of FMS journaling may lead to unwanted anomalies.

UNCATALOGED FILES

The services provided by the FMS apply only to files that are cataloged by the FMS. In addition to such files, there are uncataloged files - temporary in the sense that they are released as soon as the job using them terminates - or files that may be cataloged by means other than FMS. None of the services described herein apply to such uncataloged files, although similar services are sometimes provided by other operating system subsystems, software, or applications programs.

Temporary files are used for input, working storage, and output to another activity of the same job. Examples of these are the input file with source language for a compiler, compiler intermediate files, and output files with object or assembler programs.

Permanent uncataloged files are used for recovery by General Comprehensive Operating Supervisor (GCOS) subsystems in case of system failure. Because they are not cataloged, they are not vulnerable to FMS failure and because the files are assigned permanently assigned space, system failure does not cause release and subsequent reassignment of the file space.

NON-FMS SERVICES

The services that FMS provides depend on supporting services performed by other operating system subsystems and the Time Sharing Executive. Although occasionally referenced in passing, the operation of these other programs is not described herein.

The following application programs perform services on files. The interface between many of these programs and the FMS is the Peripheral Assignment Table (PAT) in which information about a cataloged file is placed by the FMS.

- File and Record Control (GFRC)
  (See File and Record Control manual.)

- Integrated Data Store/I (I-D-S/I)
  (See I-D-S/I User's Guide and I-D-S/I Programmer's Guide.)

- Indexed Sequential Processor (ISP)
  (See Indexed Sequential Processor manual.)

- COBOL
  (See COBOL manual and COBOL User's Guide.)

- FORTRAN
  (See FORTRAN manual.)

- General Loader
  (See General Loader manual.)

- Utility
  (See Utility manual.)

- Bulk Media Conversion (BMC)
  (See Bulk Media Conversion manual.)

- Accounting Summary Program (GSEP)
  (See Summary Edit Program manual.)

- Unified File Access System
  (See Unified File Access System - UFAS manual.)

- UTL2 Routine
  (See UTL2 Routine manual.)

SECTION II

CATALOGING

FMS STRUCTURE

The File Management Supervisor (FMS) administers a structure of mass storage records to keep track of information about files and authorized users. A separate substructure is created for each user to record information about files cataloged under that user name. A System Master Catalog (SMC) is employed as an index to these substructures. The SMC has an entry for each user authorized to reference cataloged files, to use the Time Sharing System, or both.

The records in each user substructure are organized to allow hierarchical grouping of files for the user. At the lowest level, there is a file description for each file in which is kept:

1. information to allow mapping from file to source location
2. specifications by file creator
3. counts of jobs currently using the file
4. the information recorded by FMS about the file

For each group of files there is a catalog that serves as an index to the file descriptions. Catalogs may also index subordinate catalogs.

At the highest level in the user substructure is a User Master Catalog (UMC) that indexes the substructure for the user. Where there are no subordinate catalogs, the UMC indexes each file description. When there are subordinate catalogs, the UMC indexes catalogs as well as file descriptions immediately subordinate to the UMC.

FMS imposes no theoretical limits on the depth of subordination but, because the name of each (sub)catalog must be provided and allowed for within GCOS (along with allowance for a possible password) plus a file name, the neccessity for central store ("in-memory") buffers by various portions of GCOS makes for practical subordination limits. These practical limits for subordination are as follows:

FMS - 49 (sub)catalogs

System input processor - 6 (sub)catalogs

Time Sharing System - 3 (sub)catalogs

MME GEMORE - implicity matches FMS limits

(The form "(sub)catalog" is used here to point up the fact that each descriptor string must begin with a UMC name that the user might not regard as "subordinate".)

As the diagram of the FMS structure below shows, qualification is needed to not only make files for one user distinct from those with the same name for another user but also to make files under one catalog distinct from others with the same name and for the same user but subordinate to another catalog.

NOTE: The term "catalog/file string", often encountered in discussions of file systems, is replaced in this manual by the term "qualified file name" in order to provide a meaningful expression.

Removable Structure

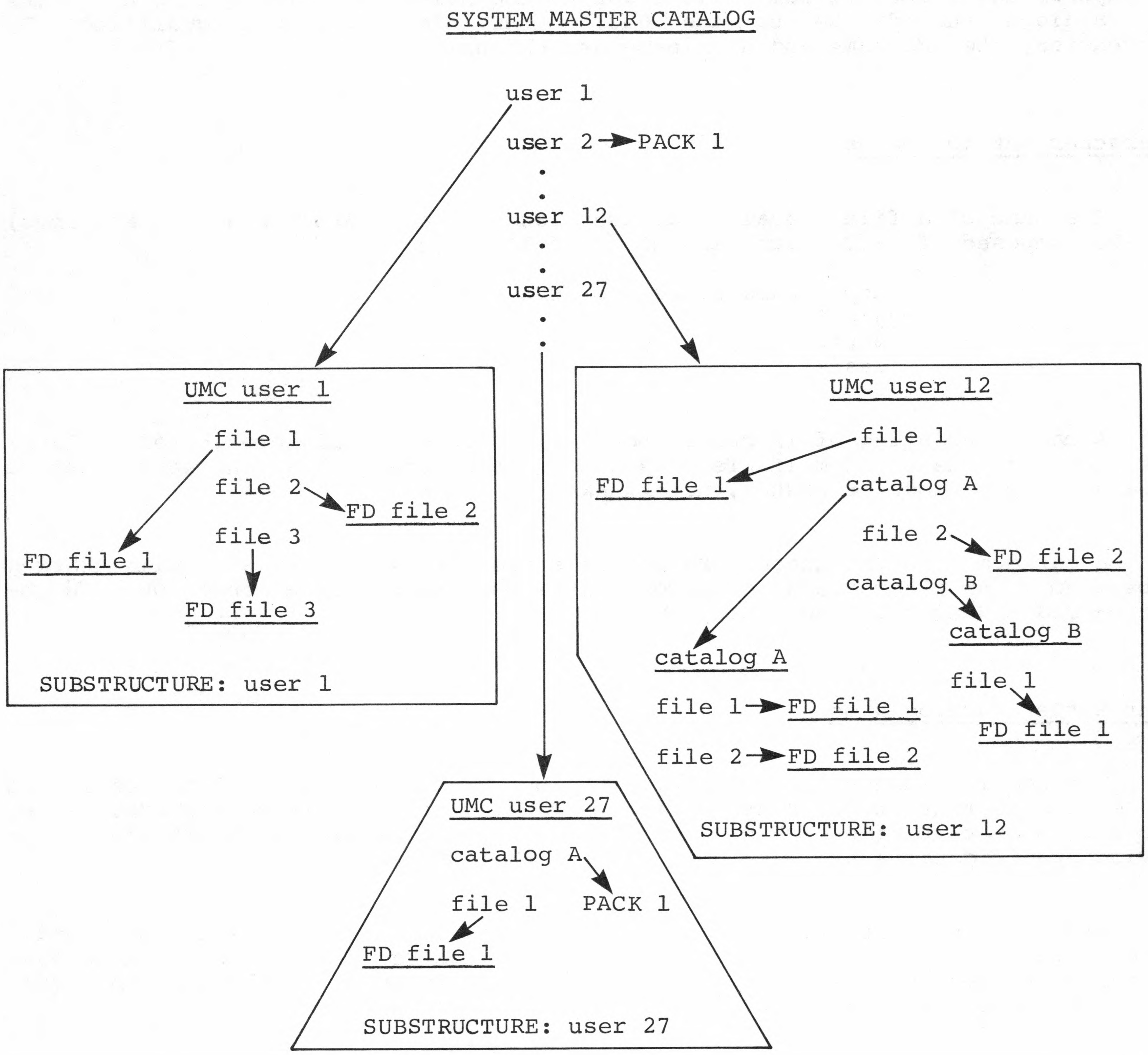
To allow files and catalogs to be kept offline, the FMS structure described above is repeated on each removable structured disk pack, but substructures are provided on a pack only for users authorized to place catalogs and/or files on that pack.

Instead of the System Master Catalog, a Pack Master Catalog (PMC) is constructed on each pack. It has an entry for each user that can catalog on the pack. The PMC entry indexes the substructure on the pack for that user.

The entire substructure for a user can be contained on a single removable structured disk pack by creating the UMC indexed from the SMC entry for the user on the pack. Typically, a user may have several substructures, one on nonremovable devices and one on each of several packs. In this case, the UMC on a nonremovable device references subcatalogs on the different packs as well as files and subcatalogs on the nonremovable devices.

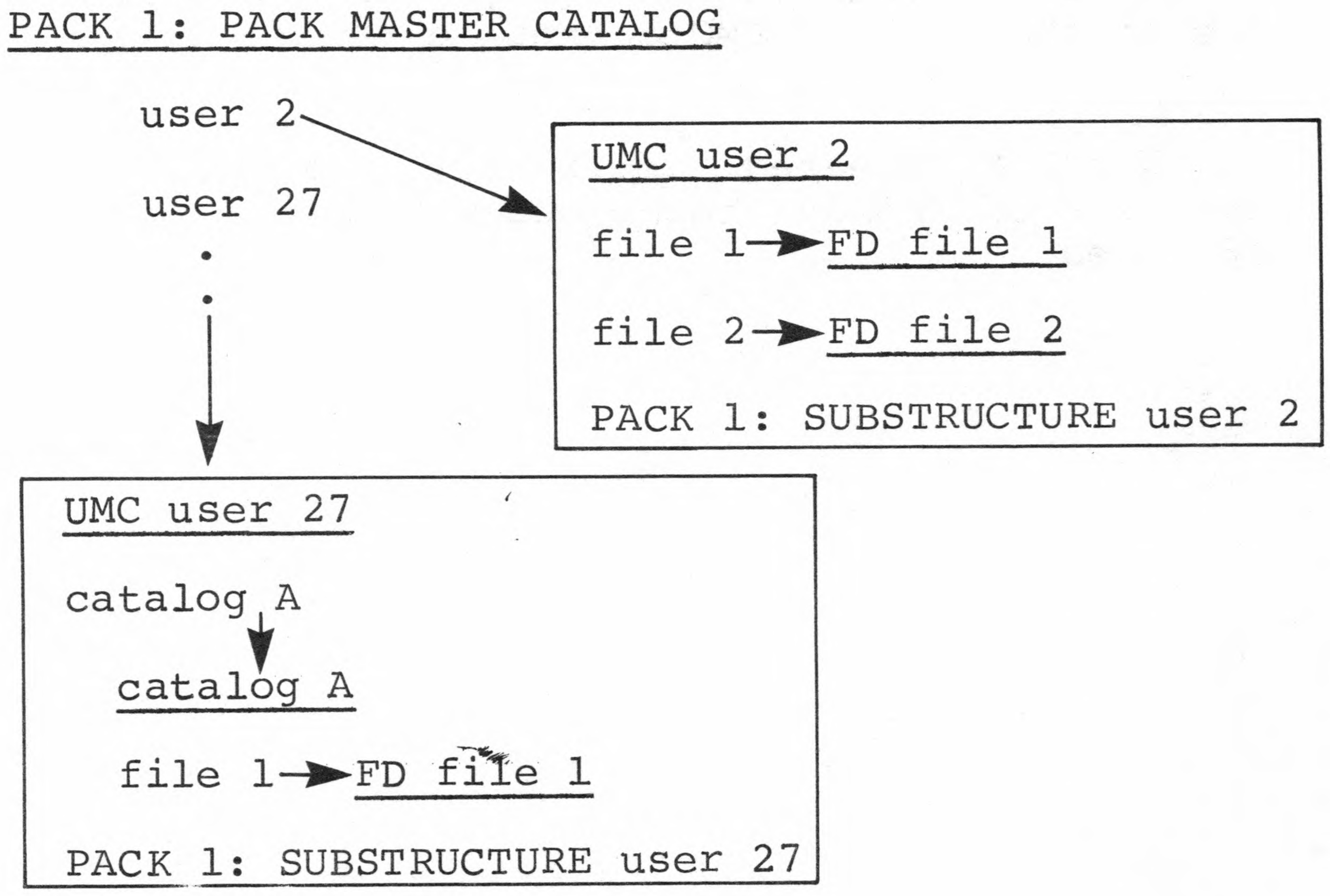
FMS Structure Example

An example of the FMS structure follows:



Files shown:

user 1/file 1
user 1/file 2
user 1/file 3
user 12/file 1
user 12/file 2
user 12/catalog A/file 1
user 12/catalog A/file 2
user 12/catalog B/file 1
user 2/file 1
user 2/file 2
user 27/file 1
user 27/catalog A/file 1



FD = file description

NAMES

Files, catalogs, and users are named. File and catalog names are qualified by superordinate catalog names (at least by UMC name) and perhaps also by names of catalogs subordinate to the UMC. User names are not qualified. By convention, the UMC name and user name are the same.

Character Set for Names

The name of a file, catalog, or user (as well as passwords and pack names) may be composed of 1-12 characters of the following:

upper case letters of the alphabet
digits
periods
dashes

A name consisting of 12 zeros, however, is specifically prohibited. Blanks are not permitted. If multiple word names are desired, then, the words must be separated by periods or dashes, not blanks.

A maximum of eight characters per file name is imposed by some time sharing subsystems. An alternate name temporarily substituting for a longer one can be introduced and used in each subsystem.

User Master Catalog Name

The UMC is usually created implicitly by FMS in the course of responding to an explicit request to create a file or catalog subordinate to the UMC. If no UMC exists at that time, FMS creates one with no password, permissions, or device specifications and with user name as the UMC name.

The UMC can be created, explicitly, however, in order to specify passwords, permissions, or device. In this case, the name specified as the UMC name must be the user name under which the UMC is to be cataloged, and the creator must identify himself as that user.

A UMC that has been created implicitly can have passwords and/or permissions specified for it by means of a catalog modify directive. But the name of the implied UMC cannot be modified.

Whether created implicitly, explicitly, or modified after creation, the name of a UMC always coincides with the name of the user under which it is cataloged.

Qualification

A user name is never qualified. A file name is always qualified (at least by the UMC name) and perhaps by the name of a catalog subordinate to the UMC it is subordinate to, by the name of a catalog subordinate to such a catalog, etc. A catalog name, unless it is the UMC, is also qualified in the same way as a file name.

Names of qualifying catalogs precede the name of the qualified file or catalog name, and the qualifying and qualified are separated by a slash. The format for all names except the UMC is:

$$\text{UMC name/ } \left[\text{catalog name/}\right] \ldots \begin{Bmatrix}\text{file name}\\ \text{catalog name}\end{Bmatrix}$$

The brackets indicate options, the braces indicate choice of selection. See the example of FMS structure above for examples of qualification.

To ensure that qualification results in distinct names, a request to create a file or catalog is refused if the name in the request is already present in the catalog immediately subordinate to the requested file or catalog. For example, a file named A/B/C cannot be created if a file or catalog named C, subordinate to catalog A/B, already exists.

CATALOGING OPERATIONS

The allowable operations on FMS file descriptions and catalogs are:

1. Create, modify, delete, and list:
   SMC entries
   PMC entries
   catalogs
   files (modify and list file descriptions only, not file contents)

2. Lock files and catalogs.

3. Inquire about files (file descriptions only, not file contents) and SMC entries.

4. Allocate, reallocate, deallocate, supply more mapping information for, and grow files.

The following definitions apply:

1. Creation - the construction of a catalog or file description, and particularly in the case of file creation, usually means the assignment of space for the file.

2. Modification - the changing of a catalog or file description, but no changing of file content.

3. Deletion - the removal of a catalog and any subordinate catalogs or file descriptions and usually the release of space assigned to files. Deletions are achieved in two forms: in one, the file space is overwritten with zeroes before release and in the other, this relatively time-consuming operation is skipped.

4. Listing - the display of catalogs and file descriptions but no display of file content.

5. Locking - prevent creation of subordinates to a catalog or deletion or allocation of (subordinate) files.

6. Inquiries - provide program access to information in catalogs or file descriptions, but not to file content. Inquiries about SMC entries are used by the Time Sharing System for logging on a new time sharing user and for logging the user off.

7. Allocation - grant read or read and write access to file content and supply mapping information that this requires (more fully described in "Requesting Allocation", below). The other operations listed after allocation are described in "After Allocation Cataloging Services", below.

Sources

These cataloging operations are performed in response to requests received via:

1. $ FILE ... NEW control card in job: to create and allocate file.

2. $ FILDEF control card at System Startup: to create file.

3. MME GEFSYE from job: to create, modify, and lock files and catalogs and to delete and inquire about files.

4. Time sharing subsystem MAST: to create, modify and delete SMC entries.

5. Time sharing subsystem MAS2: to list SMC entries.

6. Time sharing subsystem ACCESS: to create, modify, delete, lock, and list catalogs and files.

7. $ FILSYS activity: to create, modify, delete, lock, and list catalogs and files and to create, modify, delete and list SMC and PMC entries, by way of directives.

8. $ PRMFL or $ FILE...OLD control card in job: to allocate file.

9. Time sharing command: to allocate or deallocate file.

10. MME GEMORE from job: to allocate or grow file.

11. MME GERELS from job: to deallocate file.

12. System calls: to perform all operations.

In addition to information about "Requesting Allocation" and "After Allocation Cataloging Services," this manual also describes (3) MME GEFSYE from job, and (7) directives input to $ FILSYS activity. For information about the other sources of cataloging operations, consult the following references:

TSS General Information
Control Cards Reference Manual
General Comprehensive Operating Supervisor

Rules for User Operations

There are restrictions as to who can perform different cataloging operations. Any operation on an SMC or PMC entry is privileged.

For the other operations, the user must be identified by means of a $ USERID control card (or a special USERID directive to FILSYS) to determine whether the operation can be performed. The control card (or directive) contains the user name and password of the user submitting the request.

File or catalog creation can be performed by the user identified by the SMC entry indexing the substructure in which the file description or catalog is to be placed. For example, a file to be cataloged under user A can be created by a user identified as user A.

However, a user different than the user under which a file or catalog is to be placed can create a file or catalog provided the user has create permission. Permission to create under another user name may be a prerequisite if the ability to create under one's own name is not authorized.

File or catalog modification can be performed, on the other hand, not by the user under which the file or catalog is cataloged but by the creator of the file or catalog. Because of the create permission, these may not be the same (although usually they will be). In addition, a user with modify permission can perform the file or catalog modification.

Similarly, file or catalog deletion can be performed by the user who created the file or catalog or by a user who has purge permission.

File or catalog listing can be performed only by the file or catalog creator. The listing indicates any list subordinates which may have been created by another user.

File or catalog saving and restoring are similar to listing in that they can be performed only by the file or catalog creator. But the creator of a superordinate (or the owner, in the case of the UMC) can save or restore subordinates.

The following list summarizes these rules of use:

create - substructure owner, or user with create permission

modify - creator, or user with modify permission

delete - creator, or user with purge permission

list - creator; owner can list subordinates

save - creator; owner can save subordinates

restore - creator; owner can restore subordinates

See the discussion on "Permitted Actions" under "Permissions", Section III for more information about create, modify, and purge permissions.

REQUESTING ALLOCATION

Before an operation on the contents of a file - such as read or write - can be performed, an allocation of the file to the user for that operation must be provided. The allocation is provided by the GCOS Peripheral Allocator or the Time Sharing Executive. When the file is a cataloged file, the FMS verifies that the user has permission to perform the requested operation on the file and, if the request is granted, supplies the information for the allocation.

A file allocation table (called a PAT, for Peripheral Assignment Table since it is also used to control reference to any peripheral) is prepared by FMS for a cataloged file. The PAT is used by GCOS to map file address to device and device address, and to constrain operations on the file to those for whom an allocation was granted. Because of the limited memory space available for PATs, only a finite number of files can be used by a single activity or time sharing user. For batch users about 20 to 25 files can be used. For time sharing users, a maximum of 20 files are available for each user.

For the majority of cataloged files, providing the PAT is all the FMS does to control operations on the file. Some of the file protection (against device failure or incomplete update, for example) however, requires FMS to intervene on every read or write.

In addition to mapping information placed in the PAT, FMS also provides for returning a word (36 bits) of information on allocation. The word includes one bit to indicate whether the file has been written in since it was created (if it has, the first bit is set on). Other bits in the word may be whatever was cataloged with the file on creation or modification of the file description.

An example of a use of this 35 bits of user-supplied information (as it is called) is to record a security classification of the data in the file. This word is returned only when allocation has been granted. If not granted, a code and message is returned, indicating why the request was denied.

To grant allocation, FMS requires four types of information:

1. identification of user making request

2. identification of file

3. whether random or sequential treatment is requested

4. type of allocation requested

This information is provided by the Peripheral Allocator in two different ways depending upon whether the request is static (by means of control cards submitted with the job) or dynamic (by means of MMEs issued by the job) and by the Time Sharing Executive (either from time sharing commands or subsystem-issued derails (DRLs)).

For more information, see TSS General Information; see the Control Cards Reference Manual for $ PRMFL or $ FILE control cards.

Identification of User

The user identification is the 1- through 12-character name and 1- through 12-character (log-on) password entered in a System Master Catalog to authorize use of cataloged files by the user.

In batch use, the user name and password are submitted on a $ USERID control card. Allocation requests are submitted on control cards or dynamically by means of a MME GEMORE.

In time sharing use, the user name and the log-on password are submitted by a terminal operator in response to a request for USERID and PASSWORD, respectively.

Identification of the File Requested

The file identification is the name of each catalog to which the file is subordinate, beginning with the UMC name (usually the same as the user name) followed by names of subordinate catalogs (if there are any) and ending with the file name. Each catalog and the file name must be suffixed with a dollar sign and password if a password has been specified for the catalog or file. Name and password are separated from a following name by a slash.

The format of the file identification is:

```
UMC name [$UMC password] [/catalog name [$catalog password]]...
     /file name [$file password]
```

The brackets in the format imply options and indicate that the minimum (and by far most common) file identification is:

```
UMC name/file name
```

or since the UMC name is usually the same as the name of the requesting user:

```
requesting user name/file name
```

Notice that the UMC name is that of the user under which the file is cataloged, which may or may not be the same as the requesting user.

The format above is required on a $ PRMFL control card and may be provided in time sharing commands. The Time Sharing Executive provides two services to allow shorter file identification:

1. When the file has been previously allocated in the same terminal session (and not explicitly removed from the user's active file table), only an alternate name (a one- through eight-character name given the file on allocation) need be given. Instead of the alternate name (which may not have been given), only the file name can be given provided it is the only file with that file name in the active file table.

2. When the file is immediately subordinate to the UMC for the requesting user (the name of the UMC and user are the same), only the file name need be given. The Time Sharing Executive will supply the requesting user's name as the UMC name. If the allocation is refused because a password is specified for the file, the Executive will ask the terminal operator for the password. If desired, the password can be given with the file name.

Thus, in time sharing, three short forms can be used:

1. alternate name

2. file name

3. file name$password

If the format required on $ PRMFL card for a file identification is used in a time sharing command (the one containing catalog name qualification), any required passwords (for catalogs or file) must be included.

For dynamic allocation via MME GEMORE or DRL FILACT, names and passwords are provided in the calling sequence - catalog name first followed by its password or blanks if no password is required, followed by subordinate catalog name and its password or blanks, and so on until the file name and its password or blanks are given. Names and passwords are written left-justified, blank filled, each in two words. The variable length string is terminated by a word containing the value of minus 1.

File Codes, Logical Unit Designators, Alternate Names

On a $ PRMFL or $ FILE...OLD control card, a two-character file code must be specified as the internal and local identification for the file whose qualified name appears on the card. The file code is used within the program in calling sequences, file control block, error notices, etc. The file code to file name equivalence applies only to the activity in which the control card appears.

Another form of identification, the so-called logical unit designator, can also be specified on a $ PRMFL or $ FILE...OLD control card. The purpose of this designator is to allow $ FILE control cards in subsequent activities of the same job to refer to the file by a different file code without establishing an equivalence between the new file code and file name in this activity. The designator serves to equate the file code in the subsequent activity with the file assigned to the same designator in a previous activity, the allocation for which has been continued from that activity to this one. Three requirements must be met to identify a file by means of the logical unit designator:

1. both activities must be in the same job,

2. the same designator must occur on a $ PRMFL or $ FILE...OLD control card in a previous activity,

3. and the allocation of that file must be continued from that activity to the present one.

One reason for using the logical unit designator is to require that only one control card be changed when the qualified file name changes, as when the names of any catalogs the file is subordinate to or of the file itself or any of their passwords changes. If the qualified file name occurs on control cards in several activities of the job, each such occurrence would have to be noted, located, and changed when the qualified file name is changed. The use of a logical unit designator is not permitted when the file uses either incomplete update protection or device failure protection (see Sections V and VI). R (Release) disposition is always assumed for files using these protection options.

In time sharing usage, the name of the file itself is used as an internal and local identification of the qualified file name that appears in a time sharing command. The unqualified file name, without a password, can appear in time sharing program statements. For this identification to be effective, however, the following constraint must be observed:

> No other file with unqualified file name composed from the same initial eight characters can have occurred in a previous time sharing command unless that file was previously released.

When this constraint cannot be met, or it is not clear whether the constraint has or has not been met, an alternate file name can be given on the time sharing command. The alternate file name consists of no more than eight characters and must be different than other alternate file names and the initial eight characters of unqualified file names already given (and not yet released). The alternate file name to qualified file name equivalence is effective for the duration of the time sharing session or until the file is released. Thus it serves to fulfill the functions of both the file code and logical unit designator on control cards.

For more information and examples, consult Control Cards Reference Manual and TSS General Information. Notice that FMS does not participate in these local file naming facilities.

Random or Sequential Files

When a file is allocated, the request can indicate by means of the MODE option whether the file is to be randomly or sequentially read and/or written. If the file is created as a sequential file, it can be either sequentially or randomly accessed, but a file created as a random file can only be accessed randomly. Sequential access to a file declared to be random is denied. A request to write beyond the end of the file results in an end of file return by Input/Output Supervisor (IOS) if access is sequential, but an abort if it is random.

Sequential access is a simulation by IOS of magnetic tape data access - get next, skip forward, skip backward. The simulation does not require any change in file format or content; the prohibition of sequential access to a random file is arbitrary.

Formerly, specifying a file to be random when it was created caused the file space to be assigned contiguously or not at all. Now, IOS makes any noncontiguity invisible. Contiguous assignment is attempted but noncontiguous is accepted for either random or sequential files. Hence the only reason for specifying a file to be random is to prevent sequential access to it and writing beyond the end of the file.

On a $ PRMFL control card, the letter R or S is required to indicate random or sequential access, respectively. Files using the ACCESS/MONITOR/ option must be allocated with random access. If sequential access is requested for a file using ACCESS/MONITOR, an I/O error is returned on the first access to the file.

On a time sharing command, only the letter R need be given, and it is optional. If no letter is given, random access to a file specified to be random file and sequential access to a sequential file is assumed. Note that no denial is possible then because of a request for sequential access to a random file.

Type of Allocation Requested

One or more words or abbreviations are required to indicate what operations are to be performed on the file and what conditions of the file are tolerated during the allocation.

On a time sharing command, either words or abbreviations for operations can be given and if more than one is given, each is separated from the other by a comma. If none are given on a command, a default of WRITE is assumed.

On a MME GEMORE or DRL FILACT, to allocate a cataloged file, the calling sequence includes a six-bit field encoded to indicate the type of allocation requested.

Basically, there are two actions on a file that an allocation allows - read or read and write. IOS or the Time Sharing Executive enforces the allocation by refusing requests to write to a file when allocation for read only has been granted. Allocations to append or execute can be converted momentarily (by content managers or the Time Sharing Executive) to write or read, respectively, to enforce the restrictions those types of allocation require. If software to make this conversion is not in place, however, append or execute allocations allow no action on this file.

The Time Sharing Executive enforces execute allocation by limiting reading of the file to subsystems it knows to be loaders or compilers. Obtaining execute allocation outside Time Sharing allows neither reading from nor writing to the file. Append allocaton is currently treated as read allocation.

The variety of other types of allocation that can be requested are used to indicate that the requester will accept the file in exceptional conditions; recovery accepts the file in a damaged or inconsistent state, for example, and read while changing accepts the file while it is being written to.

Usually a request to read or write a file is denied if this file is already being written to. A user can express his acceptance for the file being concurrently changed, however, by suffixing his allocation request with the letter C.

If monitoring of concurrency by the FMS is specified or if other control of concurrency is expected to prevent interference between one writer and a reader or another writer, the request marked to accept a changing file also accepts any overhead imposed by this control. On the other hand, if no concurrent control can be expected for the file and yet concurrent change has been allowed by the file creator, a request that accepts the uncontrolled concurrency also accepts any interference that may result. If the creator does not allow concurrent change, however, a request accepting concurrent change is treated the same as one that does not. An exception is provided with a special form of read-only while being concurrently changed; this form is called "query allocation". A query allocation request is accepted whether or not the creator allows concurrent change and whether or not the file is being concurrently changed. A read and write allocation that similarly ignores the creator's specification is not provided, however, for it could cause damage to a file for which protection against concurrent update has been specified, whereas the query allocation cannot.

Allowable Allocation Types

The words, abbreviations, and encodings that can be used for each type of allocation requested are shown in Table 2-1. For each type, the operations that are allowed on the file when the allocation request is granted are also shown. The conditions the file content must meet before the allocation request are shown in another column. It should be understood that no allocation is granted if the file (or a superordinate catalog) is security locked unless the requesting user has LOCK permission or is the file creator. See the discussion of "Security Locking", Section III.

The last column of Table 2-1 gives the permission the requesting user must have on the file if the allocation is to be granted. It should be understood, however, that:

1. the creator of the file has all permissions

2. a user with MODIFY, PURGE, or RECOVERY permission has READ, WRITE, APPEND, EXECUTE, and RECOVERY permissions

3. a user with WRITE permissions also has READ, EXECUTE, and APPEND permission

4. a user with READ permission also has EXECUTE permission

See the list of "Permitted Actions" in the discussion on "Permissions", Section III.

Table 2-1. Allowable Allocation Types

Type of Allocation					
Word	Abbrev.	Encoding	Allowable Operations on File Content	File Conditions Required	Permissions Required
READ	R	100000	read	no writers, not abort locked	READ
WRITE READ/WRITE	W R/W	010000 110000	read and write	no other writers, not abort locked	WRITE
APPEND	A	001000	read	no other writers, not abort locked	APPEND
EXECUTE	E	000100	execute	no writers, not abort locked	EXECUTE
READ/APPEND	R/A	101000	read	no other writers, not abort locked	READ and APPEND
RECOVERY	REC	111100	read and write	no other writers, abort locked	RECOVERY
SELECT	SELECT	101100	read	no writers, not abort locked, not random	READ
QUERY	Q	000001	read	none	READ
READ/C	R/C	100001	read	not abort locked	READ
TEST	T	000010	read and write to scratch file	no writers, not abort locked	READ
TEST/C	T/C	000011	read and write to scratch file	not abort locked	READ
WRITE/C READ/WRITE/C	W/C R/W/C	010001 110001	read and write	not abort locked	WRITE
PRIVATE LOAD	P L	010100 111000	Same as read and write	no other writers, not abort locked	WRITE

Notes Pertaining to Table 2-1

1. Only QUERY and the types of allocation with a C - READ/C, TEST/C, and WRITE/C - tolerate the file being written by another user.

2. Even when a user indicates tolerance for a file being written by another, request may be denied because the file is being written to and the creator of the file has specified that allocation, while the file is being changed, is not to be allowed. See "Table Summarizing Interaction of User Request and Concurrency Option" under "Concurrent Update Protection", Section VII.

3. Only two types of allocation tolerate the file being abort locked - QUERY and RECOVERY. See the discussion of "Abort Lock" under "Improper File Update Protection", Section VI.

4. If the file occupies space marked defective, a request to allocate the file is denied unless it is for QUERY or RECOVERY or the file has protection specified for it which requires FMS intervention on every read or write. See the discussion of "Defective Space" under "Device Failure Protection", Section V.

5. The only type of allocation that can be granted, no matter what the condition of the file, is QUERY. The file can be abort locked, have defective space, and be already write-allocated (even when concurrent change is disallowed). The user cannot QUERY a file unless READ permission has been granted for the file.

6. Only one type of allocation - SELECT - is sensitive to the file being defined at creation time as a random file. SELECT is used by the GCOS System Input Subsystem to ensure that its reading of the file with tape-like commands will be accepted.

7. The unnecessary forms, READ/WRITE and READ/WRITE/C, are provided for compatability. At one time, WRITE did not include READ.

8. PRIVATE allocation is the same as WRITE unless the file specifies RWW access mode. In this case, W is scored as W/C (first writer only) while PRIVATE is scored as WRITE. Hence, PRIVATE prevents succeeding R/C allocations.

9. LOAD allocation is the same as PRIVATE insofar as no other allocations, except QUERY, will be allowed. In addition, the protection options JOURNAL, ROLLBACK, and MONITOR are ignored, thus reducing system I/O during initial loading (or total recovery) of a file created with these options.

Scoring Allocation

When an allocation request is granted, a count for the type granted is advanced. The counts are kept in the file description for the file and are used to determine whether and in what manner the file is busy. When the file allocation is terminated, the count for the type being terminated is reduced.

Correct maintenance of the counts is important, for subsequent requests for allocation are often denied because the counts show the file to be already busy in a way incompatible with the request. For example, a request to write will be refused if the file is busy with readers who will not tolerate change to the file.

On system failure, normal termination of file allocations cannot be expected, and the files are left with busy indications even though the allocations are no longer active. To protect against this inaccuracy, a system startup sequence number that is incremented on each restart is recorded each time an allocation is granted. Then each time the busy counts are being examined, the restart sequence number recorded with the counts is compared with the present one. If not the same, the counts must pertain to allocations made during a previous restart and therefore they are erased. If ABORT LOCK or ROLLBACK has been requested, the abort lock would be set at this time and only QUERY or RECOVERY allocation would be allowed. Similarly, if a TDS activity was using a file with monitor control, it would be TDS locked.

If allocation is denied for any reason, the file definition is not rewritten by FMS. Therefore, an abort lock may not show as such on a catalog listing, and, if the user wishes to override the lock, a RESET/BUSY/ may be necessary rather than a RESET/ABORT/. To avoid confusion, it is suggested that both options be used; i.e., RESET/BUSY,ABORT/. Refer to the File Modify directive, Section IX, for details. If the file was in use by TDS, a TDSLOCK directive may be necessary to override the lock.

Sometimes failure of the Time Sharing System also results in failure of allocations of files to time sharing jobs to be terminated. Since the Time Sharing System can be restarted without restarting the entire operating system, the restart sequence number is not advanced, and the method described above for detecting allocations no longer active does not work. Hence a Time Sharing System sequence number independent of the restart sequence number is used, and time sharing allocations are counted separately from other allocations. In this way, counts for time sharing allocations can be erased when the Time Sharing System is restarted without disturbing the counts for other allocations that are still active. When the Time Sharing System is not restarted, an UNBUSY directive can be used to erase time sharing allocation counts.

A file modification option to reset busy indicators for the file is provided in case anomalies in system operation mistakingly cause a particular file to be indicated busy. The reset option erases all counts for the file and thus should be used only when it is known that the file is not busy.

AFTER ALLOCATION CATALOGING SERVICES

Once a cataloged file has been allocated, the PAT produced on allocation can be used to identify the file for further services, rather than the qualified name of the file being used for identification. The PAT for a cataloged file contains the device and device location of the file description for the file.

Both cataloging and protection services use the PAT to locate the file description. The cataloging services are:

1. reallocation of the file on activity restart

2. deallocation of the file

3. PAT refresh

4. file growth

5. provide file attributes (for I-D-S/I)

Although cataloged, a file already existing on magnetic tape or nonstructured disk pack is allocated as though not cataloged. That is, the PAT is not marked as for a cataloged file, nor is the address of the file description contained in the PAT. As a result, none of the services described here pertain to such a file.

File Reallocation

To provide economical recovery in case of system failure, activities in execution at the time of the failure can be restarted without resubmitting the job and re-executing previous activities. On such an activity restart, cataloged files must be reallocated, for the method described above for erasing busy counts on system restart may have removed any indication of allocation from the file description for cataloged files.

The PAT for the file is retrieved by the activity restart facility from a job detail file also used to allow a file allocation to be continued from one activity to the next. Then the FMS is called to obtain from the PAT (actually from a reference to the PAT called the PAT pointer) the type of allocation previously granted and to try to grant that same type now. The file description is located using the address in the PAT, and busy counts in the file description examined to see if the allocation type can be granted.

Usually the request can be granted because activity restarts are considered before new jobs and, if previously granted, the allocation type must have been compatible with concurrent activities that are being restarted. Sometimes, however, the request is denied and activity restart fails. One example is the case when a concurrent allocation was to write and the file is protected, as I-D-S/I files are by default, by an abort lock. Then, on system restart, the abort lock is set, which prevents any allocations except query or recovery type allocations.

File Deallocation

When a job terminates, normally or by aborting, files whose allocation is not continued to a subsequent activity are deallocated. Deallocation also occurs when the file is released via a MME GERELS. When the file is a cataloged file, the FMS is called to decrease the counts of outstanding allocations on the file.

In addition, for a write allocation, a test is made to determine if IOS has marked the PAT to show at least one write has occurred. If so marked, the file is marked non-null (thereby allowing subsequent read-only allocations), and the date and time of last change to the file are set to the current date and time. If the job is terminating by aborting, the file can be abort locked or other protection against incomplete update is performed. See the discussion under "Improper File Update Protection", Section VI.

Finally, if a request was received to delete the file when it was busy, the file description is marked to show that deletion has been deferred until the file is no longer busy. If deallocation results in the file no longer being busy, the deletion is then performed.

Since one of the reasons for deleting a file is to ensure that enough space remains of that allowed for a user to permit creation of another file to proceed, the count of space used is decreased by the amount that is released when the file is no longer busy.

PAT Refresh

The most important information about a file that is kept in its file description is mapping information: what devices the file is contained on and where on each device. A file can and often will, occupy several noncontiguous spaces on one device; these spaces are called "extents". The mapping information consists, then, of a descriptor for the device and one descriptor for each extent.

When the file is a cataloged file, these descriptors are moved from the file description for the file to the PAT (kept in the Slave Service Area for the job requesting allocation) on allocation of the file to a job. The PAT has room for only a few descriptors, so if there are many, only the first few are moved to the PAT.

When a seek to a part of the file is attempted to which the descriptors in the PAT for the file do not apply, the IOS mapping routine calls the FMS to obtain descriptors from the file description for the part of the file sought and to place those in the PAT. This procedure is called "PAT refresh".

Each time a seek to a part of a file is performed, IOS calculates how many hardware sectors (64 words apiece on disk) there are from the part sought to the end of the extent. This count is then given to the hardware controller to limit the number of sectors that will be transferred (to the device for a write or from the device for a read). If the end of the extent is encountered, the controller terminates transmission, sets an end of extent status return, and indicates the data control word it was transmitting for and how many characters remain to be transmitted.

IOS responds to this information by calculating a new seek and data control word to continue transmission to or from the next extent, if there is one. The next extent may require PAT refresh. If there is not another extent assigned to the file, IOS returns an end-of-file unless the request was for random write.

The job may (and the File and Record Control subroutine or Time Sharing System will) request that the file be grown if the transmission was a write. If the request was for random write, IOS or the Time Sharing System aborts the request.

File Growth

File growth is initiated by issuing a MME GEMORE that references the PAT for the file and can indicate by how much the file is to be grown. If the file is a cataloged file, the FMS is called to assign more space to the file. Descriptors for the additional space are placed in the file description for the file, and PAT refresh is relied on to place these in the PAT if writing is to continue on the file. (File growth can be initiated at any time, of course, not just when writing encounters the end of assigned space.)

The additional space assigned is constrained by a number of limits. Maximum stated for the user and for this file is enforced as discussed in "Space Specifications" under "FMS Use of Mass Storage Space", Section IV. Confinement of the space to a particular device or class of devices, as discussed in "Device Specifications" under the same heading, causes the growth to be limited to what is available at the moment on the specified device or device class. Subject to these limits, the file is grown by the amount requested. If no amount is requested, the file is grown by an amount proportional to the current size. (The proportion is about one-eighth; the current size is divided by eight and one is added.)

Usually the file growth causes a new extent to be assigned, but sometimes the new space assigned is contiguous with the last extent assigned to the file, as when there is rapid and repeated file growth. In this case, the descriptor for the last extent is altered to include the newly assigned space.

Growth is refused in four cases. These four cases and remedial action for each is as follows:

1. File maximum is reached. Use file modification to increase maximum.

2. User maximum is reached. Use file delete of other file(s) to decrease space assigned to user, or have installation increase user maximum by means of master modify.

3. Space is not available on any device of same type as the last part of the file was assigned to or on the device the file description specifies for the file. Use file delete of other file(s) assigned to devices of that type or to the specified device. Or save and restore file, possibly changing device specifications.

4. Descriptions for new file space require a file description extension for which space is not available. To make cataloging space available, perform a catalog create of a dummy catalog. Or save and restore file to compact space so fewer descriptions are required.

If growth is refused for a file assigned to a removable structured pack, delete a file or create a dummy catalog on that pack. The installation can increase user maximum on that pack or delete inactive files of other users assigned to the pack.

File Attributes Specifications

The File Management Supervisor provides for attributes of a file to be specified when the file is created or modified. The attributes are any information about the file that a data management supervisor may find of use in performing its services. Instead of recording this information in the file itself or in another file, the information can be cataloged with the file description.

Once the file is allocated, the data management supervisor can call FMS to obtain and return the attributes information.

I-D-S/I is the only data management supervisor that uses this facility.

Files Not Cataloged

There is no way to perform PAT refresh when the file is not cataloged, for there is then no file description to hold mapping information that will not fit in the PAT. As a result, any operation such as file growth, or creation of a relatively large file that requires more than a few extents, cannot be performed on an uncataloged file.

When the initial assignment of space to an uncataloged file is in one extent, two growths and often more can be performed. For if the newly assigned space is contiguous with the last file extent assigned, the descriptor for the last can be extended to describe the newly assigned space. When there are many growths with time intervals between in which other jobs running concurrently can obtain the space adjacent to the last space assigned, however, the attempt to continue file growth will fail.

If the initial assignment of space occupies several extents, of course, little or no room is left in the PAT for additional descriptors. It is possible that descriptors for the initial file will not fit into the PAT, so the file cannot be created.

Inability to create occurs when the size requested initially is large relative to the fragmentation of space on devices of the class requested (or to the 32,767 llinks that can be mapped with one descriptor). Fragmentation occurs not only because of assignment of space to other files but also because some space is withdrawn from assignment as defective. When the class of device requested has a total amount of space small relative to the file size requested, device descriptors for the additional devices are required.

Deallocation of an uncataloged file does not cause busy counts for the file to be reduced since the file can only be used by one user at a time. If the file is temporary, space assigned to the file is released. A file can be marked in the PAT as uncataloged but not temporary, as is a nonstructured disk pack file; in this case, no space is released on deallocation. An uncataloged file can also be deleted (including purging) on deallocation. See the description of MME GERELS in the manual General Comprehensive Operating Supervisor.

Integrated Data Store (I-D-S/I) provides a method by which uncataloged files can be used even though the attributes information is not available from FMS. See the manuals I-D-S/I User's Guide and I-D-S/I Programmer's Guide.

SECTION III

FILE SECURITY

GENERAL

Unauthorized use of a cataloged file can be prevented by means of passwords, permissions, and security locks. All of these can be specified for catalogs and/or files. Audit records can also be specified so as to show file usage.

PASSWORDS

A password must be specified for each user at the time a System Master Catalog (SMC) entry is prepared. Passwords can optionally be specified for catalogs and files.

If a catalog or file password is specified, every reference to the name for which it is specified must be accompanied by the password. If no password is specified, however, none can accompany the name.

To accompany a name with a password, the name is written followed by a dollar sign and the password:

```
name$password
```

Passwords are composed from the same set of characters as a file name: 1 through 12 characters, comprising the alphabet (upper case), digits, dashes, or periods.

Passwords need not be unique; the password for one file can be the same as one for another file subordinate to the same catalog, or the password for one user can be the same as that for another user.

User (Log-on) Passwords

A password must be specified for each user for whom a SMC entry is provided. (A SMC entry is required for each user authorized to reference cataloged files or to use the Time Sharing System, or both.)

The user password is sometimes called a log-on password, for it is required to be presented when a user "logs on" to use the Time Sharing System. In this context, the user name is presented first, and the Time Sharing System requests the password, providing a strikeover mask for it (or a switch to non-display mode on some terminal types) to prevent the password from being visible.

The user name and password are entered on a $ USERID control card for batch jobs. The password is not visible when the $ USERID control card is listed (as are other cards submitted with the job). The password stated on the $ USERID control card is not printed on the execution report printout in order to maintain password secrecy.

The $ USERID control card is required in every batch job that references a cataloged file (except for a FILSYS activity, for which USERID directives can be used instead). For jobs submitted thorugh the Time Sharing System the $ USERID control card is supplied automatically.

File and Catalog Passwords

A password may or may not be specified for a file or catalog. If one is specified for a catalog, it must accompany the catalog name when the catalog name is used. Since the catalog name must be presented to qualify names of any files and/or catalogs subordinate to it, a catalog password serves to limit use of subordinates. Even when the catalog has a password, a subordinate file can have one also. Then, of course, reference to that file requires that both catalog and file passwords be presented.

Specifying a password for a file or catalog prevents those who may be able to obtain the file or catalog name from being able to properly reference the catalog or file. Names of subordinate files and catalogs are obtainable by listing the superordinate catalog, but the listing does not include any passwords (except by means of a privileged directive). Since only the name and not the password for the subordinate is known, an attempt to modify, delete, or list the subordinate or to use a subordinate file will fail unless the password is known.

When it is desired to restrict use of a file among those users currently able to use the file, its password can be changed, or a password specified for the file if it does not already have one. Then only those users informed of the change can continue to use the file. Changing the file name would have the same effect, but if there are programs that reference the file by name, they would have to be found and fixed. And of course file name changes can be discovered by catalog listing or even listing of programs that reference the file by name.

When a password is specified for a file or catalog, it is possible to perform operations that affect the file without providing the password. A file description and a file can be saved or deleted by requesting a save or delete of a catalog that is subordinate to the file description. Similarly, a catalog can be saved or deleted by performing the operation on a catalog to which it is subordinate. When the highest level catalog, the UMC, has a password, saving or deleting of any subordinates requires that this password be given.

* Passwords are exposed on a listing only when the request is submitted by a privileged directive. Hence, passwords can be used to protect against unauthorized access.

It is true that the installation can save, delete, and list the UMC and all of its subordinates for any user by means of privileged directives, but both the gaining of that privilege and the resulting save volumes or listing reports can be physically controlled by the installation. Even when the installation prepares a listing of files and catalogs for all users, it can suppress the display of passwords if it so desires.

Timed Passwords

Instead of a single password being specified for a catalog or file, a list of passwords can be specified, along with the times of day between which each of the passwords applies. With such a list, copying the password from a terminal display or control card is not sufficient for entry; the password must be used at the allotted time of day.

Another use of timed passwords constrains the interval of time during which the file or catalog may be referenced. In this use, perhaps only a single password is specified but only for a small interval of time - say fifteen minutes - or for an interval during which terminals are under surveillance - say during the working day. In this way, any attempt to reference the file or catalog at a time other than the stated time will fail. Once a file has been allocated (by submitting a request for it during the time it can be referenced), the file can be read and/or written beyond that time for so long as the allocation remains active.

Timed passwords can only be used for files or catalogs, not for the user (log-on) password. To specify a timed password, each password is followed by a colon, the starting time, an asterisk, and the ending time and the next password is preceded by a comma. The format, then, is:

```
          PASSWORD/password:start time*end time,password:
                   start time*end time,...,password:start time*end time/
```

Start and end times are in hours and minutes using a twenty-four hour clock (e.g., 0800*1300). If the end time is not greater than the start time, two entries with the same password are assumed - one from start to midnight, and another midnight to end. From one to eight entries can be specified.

PERMISSIONS

When a file or catalog is created or modified, the creator or modifier can specify what actions are permitted on the file, catalog, or subordinate files or catalogs by what users. Permissions can be specified for anyone, in general, or for specific named users. Specific permissions replace, not extend, general permissions. Both general and specific permissions can be specified for a catalog to apply to subordinate files or catalogs. When there are several levels of subordination, the permissions at each level are accumulated.

Permitted Actions

The actions listed below are permitted. Some actions apply only to a file and, if the permission is given for a catalog, imply permission to perform the action for subordinate files. Other actions apply either to a file or catalog. One action - create - applies only to a catalog.

READ or R — Allow transfer of information from file to program but not from program to file.

WRITE or W — Allow transfer of information both from file to program and program to file. Anyone with WRITE permission, thus, has READ permission.

APPEND or A — Same as READ permission.

EXECUTE or E — Allow RUN on file only in time sharing mode. EXECUTE permission is restricted to time sharing mode.

RECOVERY or REC — Allow WRITE when file is abort locked or has defective space. Also accept MME or directive to abort lock the file or to reset an existing abort lock. Anyone with RECOVERY permission is also given permission to WRITE and hence READ.

PURGE or P — Allow file to be deleted (file description to be deleted and file space to be returned with or without prior overwrite of space) or catalog to be deleted and all subordinate files to be deleted. Anyone permitted to PURGE can also perform any of the actions permitted by RECOVERY, including WRITE and hence READ.

CREATE or C — Allow catalogs and files to be entered as subordinate to this catalog.

LOCK or L — Allow MME or directive to security lock the file or catalog (which security locks subordinate files) or to remove an existing security lock. A security lock does not apply to a user with LOCK permission (since the user is able to remove the lock).

MODIFY or M — Allow catalog or file description to be modified. Allow entries to be made in catalog for subordinate files or catalogs. Anyone permitted to MODIFY is allowed to perform any actions, since permissions can be changed. Hence MODIFY includes CREATE, LOCK, and PURGE, that in turn includes RECOVERY and hence WRITE and READ.

General, Specific, and EXCLUDE Permissions

A permission given to everyone is called a general permission; one given only to a named user is called a specific permission. If a file or catalog has both general and specific permissions specified the general apply to everyone except those users named in specific permission specifications. A general permission to WRITE, for example, can be restricted to a READ permission for a particular user by naming the user in a specific READ permission.

A user never has both the permissions specified for everyone and also those specified specifically for him. The user has only those permissions specified specifically for him, or if there are none, then those specified for everyone.

In order to restrict a user from all general permissions without at the same time giving him some permission, the user can be named in an EXCLUDE permission. The EXCLUDE, which is really a specific permission, withdraws general permissions from the named users without giving them any specific permission.

Testing for Permission

When a request is received to perform an action that requires permission, the following procedure is used:

1. For each catalog encountered in locating a subordinate file or catalog, the permissions are accumulated.

   a. General permissions are accumulated in one field.

   b. Specific permissions for the requesting user, if there are any, are accumulated in another field.

   c. If an EXCLUDE permission is detected for the requesting user, any specific permissions already accumulated for the user are discarded, but a note is kept to ensure that specific permissions and not general ones are used for this user.

2. If the requesting user is the creator of the file or catalog for which the action was requested, the user is considered to have permission. If the request is not from the creator, however, step 1 is continued for the file or catalog.

3. If the specific permission field is not empty, a test is made to see if the user has accumulated specific permission for the requested action.

4. If the specific permission field is empty, the same test is made but of the general permission field.

5. If the test in 2, 3, or 4 fails, the request is denied.

Format for Specifying Permissions

General permissions are specified by simply naming the actions permitted everyone, one after the other, in any order, with separating commas:

```
action,action,...,action
```

Specific permissions are specified by naming the action permitted, followed by a slash-enclosed, comma-separated list of names of users permitted that action. After that action, a list for another action can follow. EXCLUDE can be substituted for the action. Each list can have one or more user names. The actions may be in any order.

$$\left\{\begin{array}{l}\text{EXCLUDE}\\ \text{action}\end{array}\right\} \text{/user,user,...,user/,}$$

$$\left\{\begin{array}{l}\text{EXCLUDE}\\ \text{action}\end{array}\right\} \text{/user,user,...,user/,...,}$$

$$\left\{\begin{array}{l}\text{EXCLUDE}\\ \text{action}\end{array}\right\} \text{/user,user,...,user/}$$

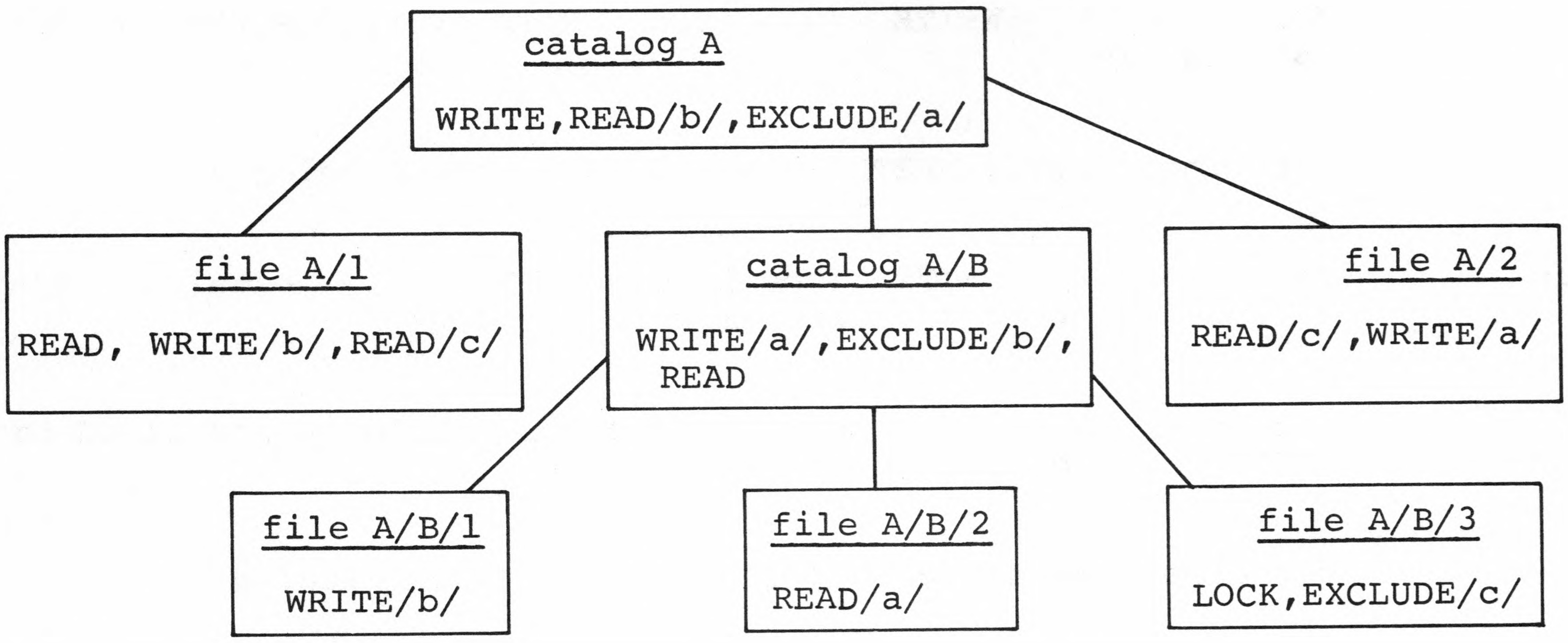
If a user has more than one action specifically permitted, the user's name is given in the list of user names for each action. It makes no sense to name the same user in the list for EXCLUDE and a list for an action, but if this is done, the listing for any action is ignored.

The same action name can be given more than once when it is convenient to include new user names permitted that action. Similarly, more than one EXCLUDE list can be given.

General and specific permissions can be intermixed, the only difference between them being that specific permissions have a slash-enclosed user list after the action name. EXCLUDE, however, requires a user list since it is only a specific permission.

When modifying permissions, if one or more general permissions are specified, the set of already assigned general permissions, if any, is replaced (not added to); if one or more sets of specific permissions are specified, any corresponding set(s) of already assigned specific permissions, by named user, are replaced (not added to).

Example of Permissions



The resulting permissions are:

file A/1

for user a, none (EXCLUDE for catalog A cancels general WRITE)

for user b, WRITE (specific WRITE for file A/1)

for user c, READ (specific READ for file A/1 prevents general WRITE for catalog A from applying)

for other users, WRITE (general WRITE permission for catalog A)

file A/2:

for user a, WRITE (specific WRITE for file A/2 not cancelled by EXCLUDE for catalog A since it is given after the EXCLUDE)

for user b, READ (specific READ for catalog A)

for user c, READ (specific READ for file A/2)

for other users, WRITE (general WRITE for catalog A) file A/B/1: for user a, WRITE (specific WRITE for catalog A/B)

for user b, WRITE (specific WRITE overrides EXCLUDE for catalog A/B)

for other users, WRITE (general WRITE for catalog A)

file A/B/2: for user a, WRITE (specific WRITE for catalog A/B accumulated with specific READ for file A/B/2)

for user b, none (specific READ for catalog A cancelled by EXCLUDE for catalog A/B)

for other users, WRITE (general WRITE for catalog A)

file A/B/3: for user a, WRITE (specific WRITE for catalog A/B)

for user b, none (specific READ for catalog A cancelled by EXCLUDE for catalog A/B)

for user c, none (EXCLUDE for file A/B/3 cancels all general permissions)

for other users, WRITE, LOCK (general WRITE for catalog A accumulated with general LOCK for file A/B/3)

AUDITING RECORDS

Five events that can affect the security of a file can be recorded:

1. creation, modification, or deletion of System and Pack Master Catalogs (type 16 record)

2. creation, modification, or deletion of catalogs and file descriptions (type 17 record)

3. setting or resetting of abort or security lock for file or file and catalog, respectively (type 17 record)

4. allocation of a file when the request is denied because required permission or password is missing or if the request is accepted or denied (type 18 record)

5. attempt to perform an action not permitted - such as write to a file when only a read allocation was requested and granted (type 18 record)

The events are recorded with a Statistical Collection File (SCF) (accounting tape) record of the type listed. The events can then be reported by means of the GCOS Summary Edit Program that reads the SCF and produces reports for each type of record.

To select the recording of these events, the installation enters a $ ACCBUF control card at startup time, indicating whether or not type 16, 17, and 18 records are to be produced. The same control card is used to indicate whether the operator can select or withdraw the selection of those record types.

In addition, type 18 records are produced in response to events 4 and 5 above only if the file allocated has had auditing specified for it. The auditing specification indicates whether all or only denied allocation requests are to be recorded. The format for this specification is that of the AUDIT option:

$$\text{AUDIT/}\left\{\begin{array}{l}\text{ALL}\\ \text{DENIED}\\ \text{NONE}\end{array}\right\}\text{/}$$

The AUDIT/NONE/ selection cancels a previously specified audit.

SECURITY LOCKING

A file can be security locked to limit allocation of the file. Security locking of a catalog limits allocation of any files subordinate to the catalog.

Allocation is denied to a security locked file if the request for allocation is from anyone but a user with LOCK permission or the creator of the file. If a catalog is security locked, any file subordinate to the catalog cannot be allocated except to users who have LOCK permission for, or are creators of, the file.

Security locking is performed by a catalog operation similar to file create or modify. It can only be performed by the user who is the creator or has LOCK permission for the file or catalog being locked.

Similarly, removing a security lock is performed by a catalog operation that can only be performed by a user who is the creator or has LOCK permission for the file or catalog being unlocked.

The catalog operation to security lock or unlock can be initiated either by means of a directive or a MME.

SECTION IV

FMS USE OF MASS STORAGE SPACE

Mass storage space is not managed by the File Management Supervisor (FMS) but by a space allocator in another General Comprehensive Operating Supervisor (GCOS) subsystem. FMS controls limits on space assigned to any user and also enables the user to select the devices to be used if system default selection is overridden. In addition, FMS provides a privileged facility to alter the contents of some areas on mass storage devices.

MASS STORAGE TYPES

Mass storage in GCOS is categorized into three types:

fixed device

removable structured disk pack

nonstructured disk pack

A fixed device is either a nonremovable device or a disk pack that can be physically removed but which GCOS treats as not removable. A fixed device, then, is always online.

A removable structured disk pack is a device that, while it is mounted, is treated by the Mass Store Manager as a fixed device but which GCOS allows to be removed. Such a disk pack can be online or offline.

A nonstructured disk pack is a device that is treated by GCOS as a magnetic tape; it is always removable, and it is allocated to one user at a time. The Mass Store Manager thus ignores it.

FMS USE OF THE DIFFERENT TYPES

The FMS obtains space for files to be created and releases space for files to be deleted on fixed devices and on removable structured disk packs that are mounted at the time of the create or delete request.

FMS places catalogs on fixed devices initialized at startup time as accepting catalogs. FMS also places catalogs on a mounted removable structured disk pack that is specifically named in a catalog request or that already contains a catalog to which the file or catalog to be created is subordinate.

The only way the user can control placement of his catalogs and file descriptions, as distinct from files, is to specify that the catalog is to be placed on a removable structured disk pack or his file is to be subordinate to such a catalog.

FMS, like the rest of GCOS, treats a nonstructured disk pack as a magnetic tape. It neither obtains nor releases space on a nonstructured disk pack nor does it ever place catalogs on such a disk pack. Like a cataloged file said to be on a magnetic tape, FMS simply passes to the Peripheral Allocator Subsystem the serial number of the pack to cause it to be mounted when the file is requested by its cataloged name. The pack, like a tape reel, may be only the first of a sequence of volumes containing the file.

DEVICE SPECIFICATION

FMS can be constrained in choosing a device to obtain space for a file by means of device specifications in a superordinate catalog or for the file. If a device specification is provided for the file, it overrides any specified in a superordinate catalog. If there are several superordinate catalogs with device specifications, the catalog specified in the most immediately superordinate is used.

If a device specification does not occur either in a superordinate catalog or for the file, FMS seeks space from the device with the most available space and with the most suitable allocation unit.

A device specification for either a catalog or a file can name a specific device to obtain space from or a type of device. If enough space is not available on either the specific device or on all devices of specified type, the file create request is denied. When FMS selects a device, denial occurs, of course, only when not enough space is available on all devices.

If the file is grown, the device specifications also constrain the devices on which space for growth is obtained. If a specific device was named, growth is constrained to that device. Even when a specific device was not named, an attempt is made to grow the file on the device that the end of the file is placed on. If space is unavailable there and a specific device was not named, space is sought from the device of the same type with the most available space. Finally, if unavailable there, the file growth request is denied.

When the file is created subordinate to a catalog on a removable structured disk pack, however, space for the file is obtained only from that disk pack either for file create or file grow. Without such a constraint, the ability to move a file from offline to online would be lost since mounting instructions are issued only for a single pack, and the mounted pack cannot be dismounted since it contains the file description.

SPACE SPECIFICATION

There are three places where the amount of mass storage space to be used by the FMS can be specified. These are:

1. SMC entry - the maximum amount of space on fixed devices that can be used both for catalogs and files cataloged under the user to which the SMC entry pertains.

2. PMC entry - the maximum amount of space on the removable structured disk pack (to which the PMC entry pertains) that can be used both for catalogs and files cataloged under the user to which the PMC entry pertains.

3. File description - the maximum amount of space to be assigned to the file (but not for a file that already exists on magnetic tape or nonstructured disk pack).

Each maximum can be specified to be unlimited and each can be changed. When a maximum specification is changed, it cannot be changed to be less than the amount currently assigned.

The SMC maximum is checked on each file create or modify and each file grow on fixed devices. A PMC maximum is checked on each file create or modify and each file grow on that removable structured disk pack. A file maximum is checked, of course, only on each growth of that file. Where the maximum would be exceeded, the request for file create, modify, or file growth is denied.

File or catalog delete releases space, thereby decreasing the amount currently assigned and increasing the amount that may be newly assigned.

The initial amount of space to be assigned a file (except for an already existing file on magnetic tape or nonstructured disk pack) can also be specified. When not specified, an initial amount of 1 link (3840 words) is assumed. If an initial amount is not specified, the maximum cannot be specified and it is assumed to be 1 link. If an initial amount but no maximum is specified, the initial and maximum amounts are assumed to be the same. When a file is implicitly created by Time Sharing as a result of a SAVE command, the initial size is set to the current size of the file to be saved and the maximum to 20 times the current size.

DEVICE SPACE UNITS

Space on mass storage devices, either fixed or removable structured disk packs, is assigned in units of space called allocation units that can be different on different devices. The smallest allocation unit is one llink, which is 320 words, 1280 9-bit bytes, or 1920 6-bit characters. Other allocation units are integral multiples of llinks:

1, 2, 4, 6, 12, 24, 36, 48, or 60.

The allocation unit for each device is specified at startup time on $ INIT control cards.

The space assignments requested from the FMS when a file is created or grown is either in terms of llinks or links. A link is 12 llinks (3840 words, 15,360 bytes, 23,040 characters). (The neologism "llinks", in fact, means little links.) The initial and optionally maximum space to assign for a cataloged file are specified when the file is created, and the maximum can be respecified when the file is modified.

When FMS is called to grow the file, the growth request can specify the number of llinks to grow the file by. (If no specification is given, FMS grows the file by an amount proportionate to its current size.)

When FMS assignment and device allocation unit are incompatible, space wastage can occur. For example, a maximum size of eight llinks cannot be honored on a device with a 12-llink allocation unit without wasting four llinks. To prevent so much space wastage, FMS attempts to initially assign a file to a device with allocation unit compatible with the initial space requirement, as long as it is free to select the device. FMS does not charge for wasted space in the accounting of total space currently assigned to files and catalogs.

PAGE SPECIFICATION

When protection is specified for a file that requires FMS intervention on every read from and write to the file, a specification is also required of the amount of file space that is read or written at a time. The amount is called a page (corresponding to I-D-S/I usage). The page size must be specified when the file is created, and can be respecified when the file is modified. A page size is also required for a file that is to be used with TEST allocation.

Every job writing and reading to the file, (except QUERY jobs that read from the file) must write and read one page at a time. A read or write of a number of words smaller than the page size is accepted as long as the reading or writing begins on a page boundary. A read or write of more than a page size, however, or one that does not begin on a page boundary, is detected and reported as an error. The error is reported by returning an octal status of 47771 (in bits 0-14 of the first I/O return word). See Appendix E. The page size specified must be an integral multiple of 64 words, which is the resolution provided on all mass storage devices. The maximum page size is 4032 words. If no page size is specified, a page size of 320 words is assumed. When sequential access is used, IOS and FMS assumes a page size of 320 words, which is the system standard for mass storage files, and ignores any other specification.

For a random file, page size must be specified to agree with the page size used to read from or write to the file. When using I-D-S/I, ISP, or UFAS, the page size corresponds to the page size specified for the content manager. For COBOL-68 random files, the page size must be the smallest multiple of 64 words that is larger than the record size. Likewise, if the record size is specified on a $ FFILE card, the page size defined to FMS must be the smallest multiple of 64 words that is larger than the record size (e.g., if the record size is 100 words, the FMS page size must be 128 words). For random FORTRAN files where the record size is specified through a CALL RANSIZ, a 320-word page size must be used. In some instances, the proper page size can be determined either empirically or by calculating the number of words written on each I/O. Note that if the page size is not consistent, such as in SORT collating files, FMS protection cannot be used.

SPACE UTILIZATION ACCOUNTING

Three fields in each System Master Catalog (SMC) and each Pack Master Catalog (PMC) are maintained to measure utilization of space for each user, on fixed devices and each removable structured disk pack, respectively. The fields are:

1. total space currently assigned for catalogs and files (in llinks = 320 words)

2. time of last change in space assigned (in 1/10 hour)

3. accumulated product of space and time up to time of last change

When a change in space occurs - either an increase or decrease - the space currently assigned (field 1) is multiplied by the difference between the current time and time of last change in space (field 2) and the product added to the existing accumulation (field 3). Then the total space currently assigned (field 1) is changed by the increase or decrease in space that has just occurred, and the time of last change in space (field 2) is reset to the current time.

The result of this procedure is that space utilization is measured in terms of how long the space was used rather than how much is in use at the time a report of space utilization is requested.

To obtain a record of space utilization, the installation can submit a privileged directive to cause a Statistical Collection File (SCF) record to be produced for:

1. all users on fixed devices and mounted packs

2. an individual user on fixed devices and mounted packs

3. all users on specified mounted packs

When the record is produced, the time of last change (field 2) is reset to the current time and the accumulation (field 3) is zeroed. In this way, overflow of the accumulation and ambiguity in time is prevented by requesting a space utilization record. The field capacities are such that a record is required for this purpose at least once every two years.

In addition to producing a SCF record, the installation receives a display of the accumulation brought up-to-date when a master or pack list is prepared. The display does not zero the accumulation or reset the time of last change, however, as an SCF record production does, since it is expected that the report will be used only for information and not to prepare accounting charges, as the SCF record is used.

Notice that there is no accounting of space utilized by cataloged files on magnetic tape or nonstructured disk pack.

DEVICE UTILIZATION REPORTING

A FILSYS activity can be used to report the amount of space available on all fixed devices and mounted removable structured disk packs. Space in use or withdrawn from availability is also reported.

Information about device activity and errors is also reported as an aid in selecting a device.

ALTERING CONTENTS OF MASS STORAGE

The privileged FILSYS directive, PATCH DEVICE/, can change the content of data located on mass storage devices provided the device name, absolute sector address, and correct data pattern are known. The FMS checksum can be ignored or honored for reads and writes, according to directive options.

SECTION V

DEVICE FAILURE PROTECTION

PROTECTION FACILITIES

Six facilities provide differing degrees of protection against common forms of device failure. Either files or catalogs, or both, can be protected.

1. Verifying file writes may prevent data from being written to device space from which it cannot be subsequently read.

2. Duplicating a file permits immediate recovery if one or more pages of a file on a device are unreadable.

3. Journaling file changes permits recovery if one or more pages of a file on a device are unreadable, but the recovery time is not as quick as is from duplicates.

4. Saving a file permits recovery of the entire file. Unlike duplication or journaling, the entire file must be restored. The restoring provides the version of the file at the time of the save. Hence, the file is generally out-of-date at the time of the recovery.

5. Duplicating catalogs permits immediate recovery if a catalog is unreadable. Duplicating catalogs limits the number of files that require restoring to those contained on the failed device.

6. Saving catalogs permits recovery of an unreadable catalog but recovery time is not as quick as is from duplicates. Moreover, restoring from a catalog save requires that all catalogs and all files referenced from the catalogs be restored. Restoring generally provides out-of-date versions of the catalogs and files.

DEFECTIVE SPACE

Information to map a file address to a device and device address is carried in the Peripheral Assignment Table (PAT). The PAT is used for access to any peripheral. Each job that can read from or write to the file is provided with a PAT. When the file is a cataloged file, the source of the mapping information is the file description for the file.

Space Marked Defective by Operator Withdrawal

When a check character data alert is returned from a read of a file on a mass storage device, the GCOS Exception Processing Subsystem retries several times and then gives the operator the option of withdrawing as defective the one or more allocation units of device space that include what was read (W-option). If the operator selects this option, the space is withdrawn from subsequent assignment and the PAT mapping information for the space for which the read was attempted is marked to show that file space is defective. If the file is cataloged, FMS is also called to mark the mapping information in the file description for the file so any subsequently produced PATs for the file will also show this file space as defective.

Space Marked Defective by FMS

In addition to space being marked defective because of operator selection of the withdrawal option, FMS also marks defective any file space for which recovery should be attempted on subsequent access to the space. An example is the case when a file is saved and unreadable parts of the file are encountered. On restoring the file from such a save to another device, FMS marks defective those parts unreadable on the save. Without such a mark, no warning of the need to recover those file parts would be given, since the device the file is now restored to is not going to return check character data alerts when these parts are read.

Access to Defective Space

When the IOS receives a request to map to a device space that is marked defective, it either accepts the request or aborts the job, depending upon whether the file is cataloged or not. The request is accepted if the file is cataloged because either FMS expects to intervene on each read from and write to the file, in the course of providing protection services, or because there will be no intervention. FMS denies the allocation request except in the case where the request expressly accepts a file with defective space. Allocation requests for QUERY or RECOVERY are the two types that indicate acceptance of a file with one or more defective spaces.

Identifying, Replacing, or Re-marking Defective Space

A MME provides the means by which defective space in a file or the duplicate can be:

1. identified

2. replaced by newly assigned space and mapping information changed to map to replacement instead of defective space

3. re-marked so that space is no longer shown to be defective

The calling sequence indicates the file and operation being requested. The file must be allocated to the user.

The calling sequence to identify is accepted from any user who has the file already allocated.

The calling sequence to replace or re-mark is accepted only from a user with a file allocation for RECOVERY.

The identify calling sequence not only provides a list of device spaces containing the files that are marked defective in the file description for the file, but also provides information about the file relevant to recovery, such as date and time of last change to the file, and serial number of volume with most recent FMS save of the file.

The replace or re-mark calling sequence requires device spaces to be described in the same format as provided by the identify calling sequence, and these spaces must be marked defective before they can be replaced.

Alternate Track

When a check character data alert is returned on a write request, it is usually as a result of a write and verify command. Exception Processing, after retrying several times, offers the operator the option of using an alternate track. If this option is selected, Exception Processing looks in the directory of tracks assigned for alternates on the device and not yet used, for one to use. If one is available, data in the current track is read and written to the alternate track, the requested write is performed to the alternate track, the current track header is altered to point to the alternate track, the alternate track is withdrawn from the alternate track directory, and the current track is listed as defective.

Once the track points to an alternate, subsequent reads from and writes to the track are automatically directed to its alternate, without any change in file mapping information.

EXCEPTION PROCESSING INTERFACE WITH FMS

When the GCOS Exception Processing Subsystem responds to an operator option, after having failed on its own to recover from a device error, and if the option is to abort or return status to the user, it calls FMS if the PAT for the file shows it to be one for which FMS intervenes on each read or write.

Thus, when the operator response is to abort the job (A), return to user (U), or withdraw (W) - ordinarily followed by an abort of the user - FMS is called to try to take further recovery action since it is providing protection services for the file.

Notice that when the operator selects an option that does not bring the error to the attention of the user, such as retry (R), alternate track (V), or exchange (X), FMS is not called. Hence these recovery efforts can continue to be tried prior to engaging the FMS recovery attempts.

WRITE VERIFICATION

To prevent unreadable areas on a disk from being written to, a file can have verify write specified. Then on every write to the file, the FMS intervenes to change the normal write command to a write verify command. The write verify command causes the write to be performed and the data written is then read again to see if data alerts arise. The reading requires a delay on rotating devices for the data to be under the read heads, but no time is required to position the heads.

(If no write verify command is available for the device being written to, FMS submits the write command unchanged.)

If a check character data alert is returned, Exception Processing is called to respond by obtaining an alternate track and copying data from the current to the alternate track, as described above.

Verification of writes to a file is specified with the VERIFY option by including

```
VERIFY/YES/
```

in the specifications for the file when the file description is created or modified. This facility is not available for files already assigned to magnetic tape or nonstructured disk pack.

FILE DUPLICATION

A file can have specified file duplication, in which case the FMS intervenes on every file write to issue a select and write to the duplicate.

File duplication allows a file to be used as long as neither copy of the file encounters unrecoverable I/O errors. This provides protection against mass storage device failures, which make an entire device inaccessible, as well as transient faults or problems affecting only a portion of a device.

To distinguish the two copies, they are designated Copy A and Copy B. During normal system operation, reads are made from Copy A and writes are made to both copies.

If an unrecoverable I/O error occurs, either on a read or a write request, Exception Processing calls FMS. If the request was a read (of Copy A), FMS reissues the read request using Copy B. If this read is successful, Copy A of the file is marked defective and additional file processing uses only Copy B. If Copy B is also unreadable, the I/O error status is returned to the user program. If the request was a write, and one of the two writes was unsuccessful, the copy of the file having the error is marked defective. If the request was a write and errors occurred on both copies, the I/O error status is returned to the user program. In every instance the system operator is notified of the file encountering the difficulty so that recovery procedures may be planned.

Once a copy of the file is marked defective, processing continues using only the "good" copy of the file. If an error occurs on this copy, exception processing is invoked as if the file was not duplicated.

If an error occurs and only one copy of the file is being used, it is necessary to use the good copy to bring the defective copy up to date. This is normally done by performing a RESTORE of the file, which causes the defective copy of the file to be restored from the "good" copy of the file. Upon successful termination of the restore, both copies, which have identical content, are again used for normal processing.

Specifying File Duplication

Duplication is specified for a file on creating the file description by including a specification of the RDERR option:

$$\text{RDERR/DUP}\left[\,:\left\{\begin{matrix}\text{device name}\\ \\ \text{device type}\end{matrix}\right\}\right]\,/$$

When no device specification is included, FMS assigns the duplicate to devices different than any the original is assigned to. If a device type is specified, the assignment is restricted to devices of that type different from any the original is assigned to. If a device name is specified, it must be different than the one the original is assigned to and must have enough space for all of the duplicate. A user specification of the same device for both original and duplicate is interpreted as an error.

Duplication cannot be specified for an already existing file on magnetic tape or nonstructured disk pack. Nor can a file subordinate to a catalog assigned to a removable structured disk pack be duplicated.

FILE JOURNALING

If journaling is specified for a file, changes to the file are collected online and dumped offline at convenient times, either on operator command or when the online collection space is in danger of being exhausted. Each collection is indexed by an entry that identifies the file and the job changing the file. A collection file is not eligible for dumping until the entry shows the job making the changes has terminated, or the job has indicated the changes are complete.

The collection file in which changes to the protected file are temporarily held is created, grown, and released by FMS without instruction from the job. Each allocation has a separate collection file; a job changing two different files has two collection files - one for each file - and two jobs changing the same file have two collection files - one for each job.

The separate collection files can cause a problem when journaling is specified and concurrency is allowed but FMS control of concurrency is not specified. See "Concurrency without FMS Control", Section VII.

The journal is used for two purposes:

1. To bring up-to-date (rollforward) a restore from a save of a part of the file, when the device containing that part fails, by remaking each change to the part of the file restored.

2. To restore a file to its condition at a specified time (rollback), when changes since then are considered to be incorrect or incomplete, by restoring the entire file from its most recent complete save and then remaking each change to any part of the file made prior to the specified time.

The specification on file creation or modification that causes a file to be journaled has the following form of the RDERR option:

```
RDERR/JOURNAL/
```

Journaling cannot be specified for a file that already exists on magnetic tape or nonstructured disk pack, but it can be specified for a file assigned to a removable structured disk pack.

Recovery of a File Page

If Exception Processing calls FMS to report a read error, FMS either returns the error to the user or aborts the job, depending upon whether return-to-user (U) or abort (A or W) is the option selected by the operator.

The installation or a file manager is responsible for initiating the recovery of the file page(s), perhaps as soon as the reported error is received either because the job needs to be rerun as soon as possible or it is suspected that numerous other jobs will encounter the same page.

Often, however, recovery of the reported errors is delayed until either several are to be recovered from the same save or journal volumes or resources for the search of these volumes are available. It is, of course, more efficient to recover a number of errors from the same sequence of journal volumes at one time.

When the device and type of error commonly reported for several errors indicate a failing device (so that future errors can be anticipated), the installation may choose to replace the device and restore and rollforward for all files on the device.

The job to recover a defective page utilizes the REPLACE directive in a FILSYS activity, followed by a rollforward activity. Instead of obtaining from the save and journal volumes and writing to the file all pages for the file or part of file on a device, however, only the defective pages are obtained and written.

The rollforward program is used for recovery of specific pages in the interest of reducing the number of programs to develop and maintain. Often recovery would be more rapid if the journal was searched in reverse order of production to locate the last change to the page. Then if the page had been changed recently, only a few journal volumes and no save volume would have to be searched. Since page recovery is expected to be required infrequently, however, and pages to be recovered sometimes would be those most likely changed not long after the last save of the file, the expected inefficiency of this double use of the rollforward program should be small.

Rollforward

When a device fails and files on this device are recovered by restoring them from their most recent saves, the restoration returns each file to its version at the time of the save. If the journaling of changes was specified for a file, however, the restored version can be brought up-to-date by using the journal as a source of the changes made since the save. This use of the journal is called "rollforward" to contrast with the restore that causes a rollback of the file to its version current at the time of save.

Ordinarily, not only the part of the file contained on the failed device but also parts of that file on unfailed devices need to be restored to provide a consistent version of the file. When the file is journaled, however, only the part of the file on the failed device needs to be restored, for it can be made consistent with the parts on unfailed devices by rollforward. In this case, the journal is used as a source of changes made to the part of the file on the failed device since the save.

Thus, rollforward allows files to be brought up-to-date and, when a file is on several devices, requires that only the part on a failed device (or parts on failed devices) need be restored.

Rollforward is also used to recover individual failed pages.

The rollforward procedure is as follows:

1. Restores from the latest save.

2. Gets earliest journal volume with file changes made since the save.

3. Searches on this and each later journal volume for changes to restored pages.

4. As each such page is found, writes it to the file and continues searching and writing to the file to make sure that the latest change to each restored page is the last one written to the file.

5. Stops when a journal volume has been searched, the last change on which is later than the latest change to any of the files restored.

The save restored from can be a save performed by the FMS or one performed by a utility that uses knowledge of the file format to compress the time or space required to save the file, such as an I-D-S/I utility, for example.

For the purposes of rollforward, the restore can be all of an entire file, the part(s) of one or more files on failed device(s), or all defective pages in one or more files.

When the restore uses a save written by FMS, the journal volume current at the time of the save is recorded on the save volume. For stranger saves, the date and time of the save should be used by the installation to locate the earliest journal volume with which to start.

Each journal volume contains one or more dumps of eligible collections. Each dump begins with entries showing each file to which changes have been dumped. The entry also gives the range of pages included in the collection dumped. The search examines the entries first to determine whether there are any collections in the dump with changes it needs; if not, the next dump (if there is one) on the journal is searched. If there is a collection of interest, the position of the collection on the volume can be calculated and search can begin with that collection.

Each journal also contains in its header the volume serial number of the next and prior journal volumes so that the sequence of volumes can be searched.

Entries at the beginning of each dump on each volume give the date and time of deallocation for each collection so that the last journal volume to search can be determined.

The restore program replaces the date and time of last change recorded in the file description for each file it has restored by the date and time of last change in the version from which it was restored. If journaling is specified for the file, the restore program reports the replaced date and time of last change. By noting the latest of these reported times, the installation can determine the last journal volume that is required. (It may be that a journal dump is needed to provide a late enough volume.)

If the restore program also restores the file description, the date and time of last change to the file must be assumed to be the current time to be safe, and the current time is so reported by the restore program for such files.

The restore program also abort locks files that require rollforward. The abort lock prevents any further allocation of the file, except to QUERY or RECOVERY requests. When rollforward is finished, the abort lock is removed to allow unrestricted allocation, and the date and time of last change is reset from the date and time of last change read from the journal and written to the file.

When rollforward is for specific pages, however, the file is not abort locked, and access to the file concurrent with the rollforward is allowed. When the journal search encounters the entry on the journal that announces the defective page, the last changes to the pages are known to have been read from the journal and written to the file. At that time, the defective mark for the page is erased. Until then, any job reading the page receives an error, but access to other pages of the file is unrestricted.

A special FMS/TDS rollforward procedure is necessary for files that have monitor and journal control and were accessed concurrently by TDS and batch programs. This procedure is described in the TDS Site Manual.

Rollback

The rollforward program is also used to provide rollback of one or more files to their condition at a time specified for each. The only change in the rollforward procedure is as follows: the stopping rule is changed to stop rolling forward each file when the next journal entry for a file has a date and time later than that specified for the file. The last change date and time is reset to show what version of the file is produced.

FILE SAVING AND RESTORING

Periodic dumping offline is called "saving", and reading from the dump and writing online is called "restoring". Thus saving and subsequent restoring of files serves a number of purposes, including utility operations to reassign files to devices or maintain some files offline. A major use for saving and restoring is for protection against device failure.

Not only does saving and restoring serve as a basis for rollforward or rollback when changes to the file are journaled (as described above) but serve also to provide a consistent copy of the file that can be used when journaling has not been specified or, if specified, has failed.

A file save cannot, however, be used to replace an individual, failed part of a file since pages restored from the save will not generally be consistent with unfailed and hence unrestored parts of the file. Without journaling or duplication of a file, then, failure of any part of the file requires rollback of the file to its version current at the time of its last save.

Even the most recent save may not always provide a consistent version, since it may be necessary to save the file while it is being changed. Often files are saved only when access to them is unlikely, but if a file is found write busy when it is to be saved (unless the file has journaling), saving is delayed until other files have been saved. If the file is still write busy, rather than delay further or not save the file at all, the file can be saved while it is being changed.

When a device fails, restoring each file contained on the device to its version at the time of its last save usually provides a consistent but out-of-date version. To bring it up-to-date requires that changes be remade.

Files already existing on magnetic tape or nonstructured disk pack are not saved (since they are already offline) and files on removable structured disk packs are saved only when the pack is mounted at the time the save is initiated and the pack is specified on the save directive or a file or catalog on the pack is specified to be saved.

Incremental Saving and Restoring

When frequent saving is performed to reduce the expected number of changes that have to be made on restoration from the last save, the time for and volume of the save can usually be greatly reduced by saving only files changed since the last save. This is called "incremental saving".

Although incremental saving reduces the effort required to provide frequent and hence more up-to-date saves, it increases the time required to restore all files on a failed device. For, in general, the volume of the sum of the incremental saves is greater, due to repetitions, than that of a single total save. To restore from the extra volume takes longer.

To reduce the volume of an incremental save, only small files are assumed to be candidates for saving, if they have changed since the last save. This assumption can be reversed by specifying the INCRSAVE option at a time when a file is created or modified:

$$\text{INCRSAVE/}\left\{\begin{matrix}\text{YES}\\ \text{NO}\end{matrix}\right\}\text{/}$$

The same restrictions as for total saves apply to incremental saves of files on removable media.

The time required to restore from incremental saves depends not only on the volume of each save, but also on how many increments there are since the last total save. For example, suppose taking an incremental save three times a day allows each save to fit on a single tape reel; if a total save is taken weekly, a total restore in the middle of the week requires 10 or more reels of incremental saves as well as the total save.

When a device failure occurs, first all files on that device on the most recent incremental save are restored, then those on that device on the next most recent save, and so on until all files changed since the last total save have been restored. Finally, files not changed since the total save are restored from the total save. Although the time required to restore in this way is greater than that to simply restore from the total save, the most recently changed files are restored first and all changed files are restored to their most recent version.

Since Saving

Another way to reduce the volume of frequent saves is to perform a total save followed by several saves with the SINCE option specifying that only files that have changed since the total save are to be saved. This differs from an incremental save in two ways: (1) increments are accumulated in the sense that a second since save includes all on the first since save, and (2) files are saved whether incremental saving is specified or not.

Both differences cause the volume of since saves to be larger than incremental ones. However, on restore, only the total save and the last since save need be used instead of the total save and each incremental one.

The date and time specified on a since save can be that of the previous total or since save, and with this usage restoring requires the total save plus each since save, for the last since save does not accumulate previous saves. The only difference between this procedure and the incremental save, then, is that a file is saved whether or not incremental saving is specified for the file.

To avoid the loss of files from intervening restores in a system where since saves are used, the following action is taken:

1. Every RESTOREMAST (and RESTORE, if the files do not already exist in the system) sets the restore bit to indicate that this file has been added through a restore.

2. Since saves and incremental saves save a file if the restore bit is on.

3. Any SAVEMAST that is not a SINCE unconditionally turns off the restore bit.

4. During a SINCE SAVEMAST, the RESET/RFILES/ option can be used to turn off the restore bit to prevent redundant saving in succeeding SINCE saves.

   NOTE: The RESET/RFILES/ option should not be used if successive SINCEs use an unchanged date and time parameter. The total restore would use the latest SINCE tape for restoration and those files must be present on the tape. Subsequent SINCE saves with different date and time parameters should use the RESET/RFILES/ option following restores. The total restore would use all intervening SINCE saves and would not be dependent upon the latest SINCE tape.

   If a total save is not taken following a total restore, the first SINCE save saves all files within the eligible userids.

Device Saving

When files are contained on many devices, the installation can perform the total save in more than one sequence of save volumes, each sequence to contain only files on some devices and and those different from devices being saved on other sequences. For example, with a configuration of six devices, three sequences can be produced, each with a different set of two devices.

The sequences can be produced concurrently so that save time is reduced because of overlapping reading and writing for one with that for another, or the sequences can be produced at different times.

When a device fails, only the sequence containing that device needs to be used to restore from. The saving in time can be dramatic. For example, when each of six devices requires four reels of tape to save files on the device, rather than 24 reels to search on failure of any one of the devices, only eight reels in the one of the three sequences containing files from the device need be used.

Whether the save has been organized into several save sequences or not, the restore should restore only files on the failed device to reduce the inconvenience of having all files rolled back to their version current at the time of the save.

When a single file is contained on more than one device, that file is saved on each sequence for a device that the file is on. Upon failure of any of the devices, the entire file can be restored from a single sequence to obtain a consistent version of the file. When the file is either duplicated or journaled, however, only the part of the file on each device need be saved, since rollforward from the journal or copy from the duplicate can be relied on to provide a consistent version.

An effort is made on file creation and growth to constrain a file to a single device. Hence only files that escape this constraint and are neither duplicated nor journaled are saved on more than one sequence.

An option, DEVICE/ONLY,.../ is available, however, that permits only the part of file to be saved that is on a device saved for the file. The DEVICE/ONLY,.../ option is useful at times when a large, multi-device file or several multi-device files need to be saved frequently. Parts of the file(s) on different devices can be saved or restored on separate saves or restores by concurrently running programs. Note that all of a file must be restored (unless the file is duplicated or journaled) when only the part on one device fails; this requirement ensures a consistent version of all parts of the file.

A device save can be used both for total and incremental saves, but usually the single sequence produced by an incremental save is small enough that splitting it into multiple sequences is not worthwhile. Where it is large enough to justify multiple sequences, perhaps the volume of changed files is so great that frequent, device-oriented total saves would be more practical. Whether the incremental save is device oriented or not, though, the restore should be only of files on the failed device.

Failed Device with Catalogs

When the failed device contains both catalogs and files, the device restore is not possible unless the catalogs have been duplicated. If the catalogs have to be restored from the last save, every file whose catalog is restored has to be restored, not just those files on the failed device. For if the file was deleted since the save, space occupied by the file may be in use for another file (temporary or cataloged, created or grown since the save). By restoring the file, new space is assigned not currently in use. Since there is no convenient way to tell whether a file has been deleted and its space reassigned since the save, all files whose catalogs are being restored are themselves restored.

Every device, including those that do not fail, can, and in general do, contain files cataloged on the failed device; failure of a device with catalogs usually requires that most, if not all, save sequences be searched for files to restore.

One way to avoid having to use most save sequences when a device with catalogs fails is to use catalog duplication. Then an up-to-date copy of catalogs on the failed device is available so that no catalog for a file deleted since the save is restored. See the discussion of "Catalog Duplication".

Saving and Restoring As a Utility

In addition to preparing for, or in response to, a device failure, the save and restore facilities can also be used to serve a variety of utility functions:

1. reassign files and catalogs

2. age files and catalogs not referenced recently

3. temporarily keep files and catalog for some users offline

4. change user identification names

Reassignment is done to recover from fragmentation of mass storage space. Reassignment not only affects the time required to access cataloged files (file mapping information is swapped into memory on demand when there is too much to keep in memory) but also the ability to grow or create temporary files (for which such swapping of mapping information is not performed). Other reasons for reassignment are to change the configuration of mass storage devices or to recreate the inventory of space available on one or more devices.

Files and catalogs not recently referenced can be conveniently aged by performing a total save, followed by a restore only of those files referenced since a specified date. If reference to a file not restored is subsequently desired, the file can be restored from the total save. The total save must be retained for this purpose even after more recent saves.

Another utility function served by save and restore is to avoid restoration of files and catalogs in the substructure for one or more users known not to be required for the time being. Keeping the catalogs and files offline, and restoring them only when references to them occur not only gains mass storage space but also enhances security against unauthorized reference. A substructure used only weekly, for example, need not be online all week, and having it offline keeps the file secure from damage or unwanted access. A new user identification name can be assigned to an existing file structure by means of the NEWNAM option. The new name can be assigned during user save or restore operations for any catalog level.

User Save and Restore

The saving and restoring described previously is initiated by the installation to provide protection for several or all substructures. The same saves and restores can be initiated by a user, however, to provide protection for the user's substructure.

On a user save, only catalogs and files in the user substructure are saved on a separate save volume. On a user restore, only the user substructure is restored, but it can be restored either from a user save or from an installation-initiated save that includes the user substructure.

Because a user restore is controlled by existing catalogs (if there are any), a user can perform several forms of file modification by means of save and restore not otherwise available. In each case of modification, a save is performed, the old file is deleted, a new one created, and then the restore is done.

* The device or device type to assign the file content to can be changed when creating the file description. By specifying file duplication when the file description is created, the file content can be restored to both an original and duplicate.

CATALOG PROTECTION

Device failure can also cause a catalog to be unreadable, preventing access to a catalog file. Two methods are available for protection against catalog loss - duplication and restore from latest save.

System Master Catalog Protection

Loss of part of a System Master Catalog (SMC) causes such widespread loss of function that, until restored, the system may be useless. None of the files cataloged in the substructure indexed in the lost SMC part can be referenced, and none of the users entered in the lost part can use time sharing or reference catalogs or files cataloged in any substructure.

Fortunately, protection against SMC loss is easily achieved since the SMC is rarely more than three or four links. Duplication of the SMC thus provides a great deal of protection for use of little space.

Duplication of the SMC is effected by submitting a

```
$ INFO   SMCDUP/device name...
```

control card at startup time. The device named must be different from the one on which the original SMC is placed.

When duplication is not possible, or when it is used but fails, restoration of the SMC from the last save is required. This also requires restoration of all catalogs and hence all files, except those on removable structured disk packs.

Catalog Duplication

If the SMC is duplicated, all catalogs can be duplicated by submitting a

```
$ INFO   CATDUP...
```

control card at startup time. All catalogs on fixed devices are then duplicated on a device different from the one with originals. If the installation specifies that two devices are to be used for catalogs (by including the CAT specification on the $ INIT control card at startup), catalogs for half the users are on one device and their duplicates are on the other, while catalogs for the other half of users are on the other device and duplicates are on the first. If the installation specifies either no device or one device for catalogs, catalogs and duplicates are placed on the devices holding SMC and SMC duplicates.

Catalogs on removable structured disk packs, including the Pack Master Catalog on each, are not duplicated. This restriction avoids having to assign fixed device space for the duplicates, mounting two packs at a time, or assignment of duplicates to the same pack as the original. Timely saving of structured disk packs provides good protection both of catalogs and files on the pack.

The expense of duplicating all catalogs on fixed devices is large only when there are so many catalogs that both space and time to double update them are noticeable. It is precisely when the expense of duplication is large, however, that the benefits are also many. For when there are many catalogs, there are also many files and usually many devices holding files unprotected except by file saving and restoring. As noted in the discussion of "Device Saving", device saving can be used to reduce considerably the time required for restoration in case of failure of a device with catalogs only when catalog duplicates are available.

Catalog Save and Restore

Each file save is preceded by a catalog save (except for a device-oriented save when the devices do not have catalogs). On an incremental file save, all catalogs (not just those for changed files) are saved.

Saved catalogs are used only when catalog duplicates are not available or are to be avoided, as when rollback to an earlier catalog version is sought. The most common application is the case where a total restore is to be performed, either a planned one for utility purposes or an unplanned one when rollback or multiple device failure makes the saved catalogs essential.

When the restore is from incremental saves, the catalogs saved on the last increment are those used to restore from and catalog saves on earlier increments are ignored.

Restore Lock

When a file description is restored, it is marked restore locked. The restore lock is removed when the file content is restored. While the lock is set, requests to allocate the file or to modify the file description are refused with an error.

If the file content is never restored, the restore lock remains. In this case, the file may be restored from another save tape, by using different restore options, or the file description must be deleted and then re-created and the file content recovered before the file can be used. Listing the file description provides information useful in recreating it.

File content is not restored when device specifications in the file description cannot be met or when existing catalogs are used but the file was created after the save from which file content is restored. The content is not restored if the restoring is truncated because of media or if there is an operator error.

SECTION VI

IMPROPER FILE UPDATE PROTECTION

The File Management Supervisor (FMS) cannot protect against many possible causes of a file being improperly updated, of course, since this would require not only knowledge of the file format but also knowledge of the file application. There are two causes of improper file update that FMS can provide protection against, however, for any files. First, when a job that is updating a file terminates abruptly, either because of job or system failure, the file is left in a partially updated condition. FMS provides a way to cancel these partial updates. Second, for new application programs that have been inadequately tested, FMS provides a way for testing programs against the production file without any danger of damage to the file.

INCOMPLETE UPDATE PROTECTION

Two forms of protection against incomplete update are provided:

1. Preventing further allocation to a file when incomplete update to it is detected.

2. Copying a page before it is changed and writing the before copy to each changed page when incomplete update is detected.

Each of these forms of protection depend on detecting incomplete update. For a job, termination of the job because of abort or system failure is assumed to imply incomplete update. *

The first form of protection, that of locking the file to prevent further allocation, does nothing to correct the trouble, but does keep the file from being used until it can be corrected. The other form, actually corrects the trouble by cancelling changes made by the incomplete job.

One of the two forms can be specified for a file when it is being created or modified by including a specification of the ABORT option:

$$\text{ABORT/}\left\{\begin{array}{l}\text{LOCK}\\ \text{ROLLBACK}\end{array}\right\}/$$

When the file already exists on magnetic tape or nonstructured disk pack, none of these forms of protection can be specified for the file.

If the ABORT/ROLLBACK/ option is specified, requests from time sharing to access the file with write permission are denied.

Incomplete update protection on a file is cancelled by including the following specification on a file modify:

```
ABORT/NONE/
```

When no ABORT specification is included on create, ABORT/NONE/ is assumed, except for an I-D-S/I file, where ABORT/LOCK/ is assumed if ABORT/ROLLBACK/ is not specified.

Abort Lock

The abort lock is a facility to lock a file when a job changing the file terminates abruptly. The facility is available for any file on a structured disk pack.

Once a file is abort locked, only requests for query and recovery allocation on the file are accepted. The query allocation is a read only one that explicitly indicates that the file will be accepted in any condition (file has defective space, is being restored, is incompletely updated, or is being concurrently written to). The recovery allocation is a write allocation, accepted only from a user who is the creator of the file or has RECOVERY permission.

An abort locked file is unlocked (so that requests for other allocations are accepted) in one of four ways:

1. A job with recovery allocation for the file terminates normally and has not issued a countermanding MME GEFSYE to keep the file locked.
2. A job with recovery allocation for the file issues a MME GEFSYE to unlock the file.
3. A FILSYS directive is issued by the creator of the file or by a user having RECOVERY permission:

   ```
   ALOCK file name,OFF
   ```

4. A specification on a file modify directive is used:

   ```
   FMOD file name,RESET/ABORT/
   ```

A file is abort locked in any one of following ways:

1. File has specified ABORT/LOCK/ and job aborts that has write allocation and has issued at least one write to the file.
2. File has specified ABORT/LOCK/ and request is received to allocate the file but file description shows write allocation from previous startup. (A previous write allocation to the file was interrupted by system failure.)

3. File has specified ABORT/ROLLBACK/ and was not properly recovered after a system failure.

4. A MME GEFSYE is issued by the creator of the file or by a user to whom this file was allocated to set the abort lock or by a user who names the file and who has RECOVERY permission.

Change Cancellation

When a file is protected by the FMS using ABORT/ROLLBACK/, changes made to the file can be cancelled. Cancellation occurs automatically in the following three instances:

1. A job aborts that has made changes to the file and the changes have not been indicated as being complete.

2. The system fails when there are jobs that have made changes to the file and the changes have not been indicated as being complete.

3. A job issues a MME GEROLL to perform a program rollback.

Cancellation can also occur on demand when a job that has made changes to the file, and the changes have not been indicated as being complete, issues a MME GEFSYE to cancel changes made by the job to a specified file or to all files.

The rollback of the files that cancellation of changes produces can be prevented by indicating changes are complete. This can be done in any of the following ways:

1. Normal termination of the activity.

2. Deallocate the file by means of a MME GERELS.

3. Issue a MME GEFSYE to indicate changes to a specified file or to all files are complete.

Changes made in one activity are considered complete when the activity ends. Refer to "Job Restarts and Update Completion". Note that use of the ROLLBACK option does not allow an allocation to be continued from one activity to another; that is, S(save) or C(continue) dispositions are not honored but are changed to R(release) disposition.

Befores

A common method for protecting against incomplete update of a file is to make a copy of each page that is to be changed before it is changed. Then when changes made by a job to the file are to be cancelled, the "before" copies are read for each change made by the job and written to the file.

If there are several changes to the same page, the before for the first change should be written to the file, so that the file is left in its condition prior to any change by the job.

The FMS provides two methods of collecting befores. The first method (user supplied befores) allows a content manager, that is, I-D-S/I, ISP, and UFAS, to provide the FMS with a before image before the page is modified. In the second method (if the first method is omitted), the FMS intervenes on each write to the file, if a before image of the pages does not exist, reads a before copy of the pages to be updated, and writes them to the collection file before the file is updated.

When the job making changes deallocates the file or indicates that changes
*
made so far by it are complete, the collection file for the befores either is repositioned or released.

A separate collection file is provided for each allocation to a protected file; the same job allocated to two files gets two collection files, and two jobs allocated to the same file each get a collection file.

On restart of the system after failure, FMS must recover the last page read into the before space and not yet written to a collection file (if there is one), and then must initiate cancellation for all files. If a job was in normal termination at the time of system failure or otherwise indicates completion of changes, changes made by the job are not cancelled.

Befores are specified by use of the option ABORT/ROLLBACK/.

Call POST or Cancel Subroutine

A File and Record Control subroutine, POST, is the means by which a MME GEFSYE can be issued to indicate changes are complete or to cancel the changes. The format for the call to the subroutine POST is as follows:

CALL POST (file control block address or zero, buffer address, function code)

file control block:	standard
buffer:	BSS 35 + page size (minimum of 355 words)
function code:	40 = cancel changes 41 = regard changes as complete

If zero is given in place of file control block address, all files, whether they have file control blocks or not, are indicated.

If successful on return, bits 6-11 of the file control block word 3, if a file control block is addressed, will be zero. If unsuccessful, bits 6-11 show the code for the MME GEFSYE. If there is an error address in bits 0-17 of file control block word -5, that address is called with code 8 and file control block address as arguments.

If no address is present or there is no file control block address supplied when an error is indicated, the job is aborted with reason code 1. If a file control block address is given but is incorrect, the job is aborted with reason code 2.

The caller should ensure that any buffers used to block records are written to the file before this call is made so that any changed records in the buffers are indicated complete or cancelled. When the buffer provided for the call is the same as one used to block records, the caller should realize that the POST subroutine changes the buffer contents.

Job Restarts and Update Completion

On system failure, jobs in execution can be restarted. The restart can be at the beginning of the job, beginning of the last activity, at the last checkpoint, or at the beginning of the last transaction. Restart is not performed by FMS, but FMS does perform the rollback of files changes by the restarted job. The rollback must be synchronized with the restart. Changes made prior to the point where the job is to be restarted must not be cancelled, because the job will not have an opportunity to remake them. Changes made after the restart point must be cancelled so that the job cannot make these changes twice.

Rollback is accomplished by a FMS recovery job that is spawned on system restart. The recovery job uses information saved at the time of system failure to determine what files need to be rolled back and to locate the collection files to use for this purpose. When a job changes several files protected by FMS against incomplete update, the treatment of all files is made consistent. If cancellation for one file was interrupted by system failure, cancellation for it is resumed and cancellation for all other files is performed. If update completion for one file was interrupted, completion is resumed and is performed for all other files. If recovery is not done or is incomplete, because of errors in cancellation processing or because of external problems (that is, device powered is off or removable pack is not online), then unrecovered files are left in abort lock status (see File Recovery Techniques). Since abort locking of a file limits allocation of the file to jobs requesting query or recovery, restart of a job using a file that is abort locked is prevented, unless the job had query or recovery allocation.

Transaction restart requires that the job itself include programming to indicate complete any changes made to files in the course of processing each transaction. The program may call the POST subroutine or issue MME GEFSYEs for this purpose after calling the Transaction Processing Executive to indicate that its processing of a transaction is complete. In this way, system failure during the processing of a subsequent transaction causes only the last transaction to be redelivered and message outputs and file changes made since finishing the last transaction to be cancelled. Refer to the Transaction Processing System User's Guide.

Checkpoint restart requires that the job perform a MME GECHEK. Changes made since a prior checkpoint (if there was one) or the beginning of the activity (if there were none) are then indicated to be complete. A subsequent restart cancels only changes since the last checkpoint.

Activity restart requires no adaptation in the program. File changes made in prior activities have been indicated complete by activity termination; changes made in the last activity are cancelled on system restart and the job is then restarted at the beginning of the interrupted activity.

Job restart generally cannot be performed since end of activity causes file changes to be indicated complete. Hence, changes made in any activities that have terminated cannot be cancelled. Restarting the job so that it would repeat the terminated activities would cause file changes to be made twice. For incomplete update protection to work, a multiple activity job must either accept activity or avoid restart altogether, unless only the last activity changes files protected against incomplete update.

TEST UPDATE PROTECTION

The test update protection, specified by T or TEST in the permission field of a $ PRMFL card, provides a convenient and safe way to test against a production file. This option causes all file updates to be made to a collection file instead of to the specified file. Subsequent reads from pages that were updated are made from the collection file, whereas reads from pages not updated are made from the specified file. Thus, the file content is never modified by this permission, but the program behaves as if the content was modified.

Unlike all other forms of protection provided by FMS, test update protection can be provided for an allocation to any cataloged mass storage file (no matter what protection is specified for the file when it is created or modified). In fact, the file can specify befores for protection against incomplete update or no protection at all, and yet test allocations can be provided to the file. However, the size of a page read or written in test mode must be specified for the file, or an assumed page size of 320 words must be accepted. (See Page Specification in Section IV for further details on page size specification.)

INDICATING CHANGES COMPLETE FOR TEST ALLOCATION

When a job with test allocation indicates its changes are complete, either explicitly or by terminating an activity, the changes are not actually made to the file. A note of the last change in the collection file at the time of the completion indication is made, however, to enable a subsequent cancel request to accurately cancel only those changes made since the completion indication.

FILE RECOVERY TECHNIQUES

During initial system startup operations, the FMS creates a file to keep track of system failures. The startup sequence number (see Scoring Allocation), is entered in the file when the failure occurs and is used by the recovery program to determine recovery requirements. In addition, this file contains the current sequence number which is the last entry in the file. Another file is also created and initialized during startup operations for the alternate recovery tables. This file is periodically updated with pointers to collection files when befores protection is in use; otherwise no updates occur.

On a system failure for which recovery is required, the sequence number is marked to indicate need for recovery; otherwise it is removed. The information pointing to the collection files in use at the time of the failure is written to another protection tables file. Finally, the recovery program is spawned to attempt the initial recovery and operates as previously described, using the protection tables file or the alternate tables file to locate the collection files when rollback is specified. If recovery is complete, the recovery tables files and the sequence number of the recovery are removed from the system. If recovery is not completed, these files remain known to the system and the sequence number is marked as requiring deferred recovery.

Usually, the initial recovery attempt is completed for all files. If not, then a privileged FILSYS directive may be used, at a later time, to run the recovery program again.

RECOVERY DIRECTIVES

File Recovery

The options of the RECOVER directive are used for deferred recovery which should not be attempted until the condition that caused the previous recovery attempt to fail has been corrected. It is possible, though not likely, that recovery may be required for more than one sequence number at a time. In this situation the recovery program works on the first sequence number in the list requiring recovery. No action is taken for the other sequence numbers. Once recovery is completed the sequence number is removed from the beginning of the list and the next number becomes the first number automatically.

Recovery Tables Dump

In some instances, it is desirable to dump parts of the information remaining for recovery. This may be accomplished with the RECDUMP option.

Remove Recovery Tables

Occasionally, it is desirable to remove parts of the information remaining for recovery. This is accomplished by means of the REMOVE option.

NOTE: This directive must be used with extreme care because no file recovery is performed.

SECTION VII

CONCURRENT UPDATE PROTECTION

CONCURRENCY OPTIONS

One of four ACCESS options must be selected to control concurrent allocations to a file. The default option is the NORMAL option to allow either multiple readers or a single writer. With this option, no interference of one job with another concurrently allocated can occur since only concurrent readers are allowed.

Another option, the READ WHILE WRITE, allows both multiple readers and a single writer. The writer may interfere with the operation of the readers, but since only a single writer is allowed, interference among writers which might damage the file content cannot occur, and, of course, readers cannot interfere with the operation of the writer.

Two other options, CONCURRENT and MONITOR, allow both multiple readers and multiple writers to be allocated at the same time. With MONITOR, reads and writes to the file are controlled so that any that might interfere with the operation of another job are either prevented or delayed until the chance of interference is past. With CONCURRENT, no such control over reads and writes is exercised by the File Management System (FMS). The chance of interference of one job with another must be tolerated, controlled by other means such as programming in each job or in a common file content manager like I-D-S/I, or at least controlled enough to decrease the chance or types of interference to a tolerable extent.

The CONCURRENT option is specified by including in the creation or modification of a file specification of the ACCESS option:

```
        ⎧ NORMAL            ⎫
ACCESS/ ⎨ READ WHILE WRITE  ⎬ /
        ⎪ CONCURRENT        ⎪
        ⎩ MONITOR           ⎭
```

If the file already exists on magnetic tape or nonstructured disk pack, the tape or pack is assigned privately to each allocation so that even concurrent reading is not possible. An ACCESS specification is ignored in this case, and requests for allocation that accept concurrent change do not, of course, ever encounter any concurrency.

CONCURRENCY WITHOUT FMS CONTROL

When concurrency is allowed but FMS control is not specified (the option ACCESS/CONCURRENT/ instead of ACCESS/MONITOR/ is selected), there may be problems introduced if befores and/or journaling are specified for the same file.

When befores are specified, the possibility of cancelled changes introduces potential problems. Cancellation can cause two reads of the same page to obtain different versions of the page. Another troublesome condition occurs when changes made by one job are cancelled; the changes by other jobs made to the same pages may also be cancelled. These are the kinds of problems encountered without page level control of concurrent change.

Separate collection files also cause a problem when FMS journaling is specified without page level control of concurrency. Changes made to the same page by different jobs appear in separate collections on the journal; the most recent change to a page changed by several concurrent jobs cannot be detected from the journal. Rollforward or recovery of a page is not certain to restore to the latest version.

If concurrency is allowed but FMS control is not specified, it is usually unwise to use FMS journaling and the consequences of FMS cancellation should be studied carefully. When FMS control over concurrency is specified, however, no problems with journaling occur because one job is not allowed to change a page changed by another. FMS change cancellation (using befores) is recommended, even though not required, when FMS control over concurrency is specified.

INTERACTION OF REQUEST FOR ALLOCATION AND CONCURRENCY OPTION

Even when one of the options is selected for a file that allows concurrent write allocation or reads with a write, the job must specify that it will accept an allocation concurrent with a writer. Without such a specification, the job is not exposed to either the chance of interference that concurrent allocation with a writer can entail or the delays in operation that control to prevent interference imposes.

On the other hand, even when the job specifies it will accept an allocation concurrent with a writer, allocation without a concurrent writer is forced unless an option has been selected for the file that allows the concurrent writer.

An exception exists in that the job can be granted a special request to read with a concurrent writer no matter what option has been selected for the file, and in this case, monitoring to avoid interference is not provided even though the MONITOR option has been selected for the file. This is the QUERY request.

A corresponding special request to write is not provided since a writer can damage the file content, whereas a reader cannot.

INTERACTION SUMMARY

Table 7-1 shows, in summary form, when a request is granted and when denied as a function of the type of request, the concurrent option selected for the file, and existing allocations already granted to the file. If there are no allocations to the file, no request would be denied (because of concurrency restrictions).

Table 7-1. Interaction of User Request and Concurrency Option

TYPE OF ALLOCATION REQUESTED	EXISTING ALLOCATION(S) TO FILE FOR EACH DEGREE OF CONCURRENT CHANGE SPECIFIED									
	NORMAL		READ WHILE WRITE				CONCURRENT OR MONITOR			
	R	W	R/C	R	W/C	W	R/C	R	W/C	W
R/C,read while changing	A	D	A	A	A	D[a]	A	A	A	D[a]
R,read	A	D	A	A	D	D	A	A	D	D
W/C,write while changing	D	D	A	D	D	D	A	D	A	D
W,write	D	D	A	D	D	D	D	D	D	D
P,private	D	D	D	D	D	D	D	D	D	D
L,load	D	D	D	D	D	D	D[a]	D	D	D

[a] Denials in these cases assume that a job granted or requesting R/C allocations accepts the file while it is being changed because any writers are being controlled to prevent interference with other users. Such control is omitted when W allocation is granted; hence concurrent R/C and W allocations are denied. If uncontrolled changing is acceptable, Q instead of R/C should be requested. It should be noted, however, that Q also permits access to abort-locked files.

Notes Pertaining to Table 7-1

1. TEST is treated as READ and TEST/C as READ/C.
2. QUERY allocations are not shown because they never cause denial, and can be denied only if the number of query allocations reaches system limits (currently 63).

3. If exclusive W access is required to a READ WHILE WRITE file, the file should be accessed with P or L.

4. Request is either Denied (D) or Accepted (A) according to type of allocation requested, degree of concurrent change specified, and existing allocations to the file.

5. EXECUTE is treated like READ. APPEND, READ/APPEND, and RECOVERY are treated like WRITE.

6. W on a READ WHILE WRITE file is scored as W/C, but only one writer is allowed.

*

MONITOR OPTION

When the concurrency option ACCESS/MONITOR/ is specified, FMS allows multiple update activities to use a single file as if each of the activities had exclusive use of the file. To accomplish this, FMS detects cases where activities may potentially interfere with one another by keeping track of which users have accessed file pages and whether the access was for a read or write. File access rules applied at the file level under the default ACCESS/NORMAL/ are applied at the page level under ACCESS/MONITOR/. For any given page only multiple readers or a single writer are allowed.

Access is allowed to the page if:

1. The page has not been previously accessed by another activity, or

2. The access is to read the page and no other activity has written the page (i.e., multiple readers of a page are allowed).

Access is either delayed or denied if:

1. The access attempt is to write and another activity has accessed the page for either reading or writing (i.e., a writer of a page must have exclusive use of the page), or

2. The access attempt is to read and another activity has accessed the page for writing (i.e., one cannot read a page that another user has written until the other user indicates its changes are complete).

In these cases, access is denied if the activities causing the delay are themselves delayed (either directly or indirectly) by the activity requesting the access. Otherwise, the activity is delayed until the activities currently using the page either indicate their changes are complete or cancel their changes.

A special case concerns the Transaction Driven System (TDS) as a user of the file. Access is never delayed or denied to TDS as an activity; however, separate TDS input/output requests may be delayed. When a TDS input/output request for pages is delayed, FMS forces the batch program, holding the pages which TDS is attempting to access, to rollback to the preceding checkpoint and release its pages to TDS.

On a denial, the activity requesting the access may, in some cases, be able to continue, but more likely it would need to cancel changes made so far to files and start over again. On rerun, the activity would either find that the other activity accessing the page had indicated its updating was complete or would be delayed on an attempt to read the page until that update indication was received. The cancellation and rerun of an activity can be done by means of a MME GEFSYE or by an abort and restart.

This access algorithm ensures that an activity could continue to read the same version of a page until it either writes the page, indicates changes are complete, or cancels the changes. Similarly, an activity writing a page can read the updated version of the page until the activity rewrites that same page, indicates changes are complete, or cancels the changes. As soon as the activity indicates changes are complete or cancels the changes, it cannot expect to read the same version of the page, as other activities may have been allowed to write to the page.

Because changes are considered to be complete at the end of an activity, a reader cannot expect a page read in one activity to be the same in another. By completing the changes in each activity, jobs are not delayed waiting for pages by a concurrent job that is delayed between activities. But a multi-activity job cannot expect summaries produced in one activity, for example, to agree with details in an earlier activity if concurrent change to the file is allowed.

Hence, when a job requires pages to remain unchanged by another job from one activity to another, the job should not accept concurrent changes to the file. Then, although changes it makes are still considered complete at the end of each activity, its allocation to the file prevents allocation to another job that could change the file.

A job should indicate that its file accessing is complete on monitored files promptly so as to release pages it has read or written for use by others. Completion indication should be given as soon as the job has arranged (by end of transaction indication, checkpoint, or other means) that any file changes it has made will not be remade on restart and/or when it no longer requires that pages it has read not be changed by others. Other jobs may be waiting until this job indicates completion to read pages it has changed or to indicate their own completion, that in turn other jobs can be awaiting. When completion is indicated, the indication applies to all files accessed with monitoring by the activity. This is required to prevent deadlock situations that could arise if completion for only a subset of activities-monitored files was allowed.

MONITORING AND PROTECTION AGAINST INCOMPLETE UPDATE

When the concurrency option MONITOR is selected, befores protection (ABORT/ROLLBACK/) against incomplete update is recommended. Then, if a deadlock occurs between two or more activities using the file, the changes made by one of these activities can be cancelled so that the other activities can continue. Without ABORT/ROLLBACK/, such a deadlock can not be broken without the possibility of leaving the file in a partially updated state. Only when deadlock does not occur or when cancellation is not necessary should the MONITOR option be used without the ROLLBACK option.

FMS CONTROL OF CONCURRENCY SUMMARY

Table 7-2 summarizes times when a requested read or write is allowed, delayed, or denied when the file is being monitored to prevent concurrent interference. Denial is with an octal status 47772 (in bits 0-14 of the first I/O return word). See Appendix E.

Table 7-2. Control of Concurrency

REQUEST FOR SAME PAGE BY ANOTHER JOB TO	PAGE ALREADY ACCESSED BY JOB FOR	
	read	write
read	allow[b]	delay or deny[a]
write	delay or deny[a]	delay or deny[a]

[a] Job is delayed unless a deadlock situation is caused by the delay, in which case access is denied.

[b] A site-option patch (see System Operation Techniques manual) can be used to change "allow" to "delay" or "deny"; that is, only a single job is allowed to access the page.

RESERVING PAGES

A MME GEFSYE can be issued to reserve pages before input/output processing is initiated, if the job knows in advance the pages needed to read or write. Two separate MME GEFSYEs are available. One reserves a page for exclusive use (for writing) and another reserves a page for shared use (for reading). These functions are useful to detect potential deadlocks before processing starts or to provide a gating mechanism between programs. Indiscriminate use of these functions is to be avoided because of the added overhead and concomitant increase in run time.

RELEASING PAGES

Normally, pages reserved for concurrent control are released for use by other programs by indicating that changes are completed or by cancelling changes (i.e., file checkpoint or rollback). However, specific pages can be released from concurrent control by means of a MME GEFSYE before changes are completed or by cancelling changes. This function is effective for pages that were only read by the program. In particular, this function is intended to allow other programs to update the released pages and to reduce interference on high-use pages.

PERFORMANCE CONSIDERATIONS

To use the MONITOR option effectively, it is necessary to avoid frequent updates of a single page (or a very small group of pages) by different programs and access to many pages between checkpoints. Frequent updates of the same page result in an excessive number of delays and denials (rollback). For example, if a database has a single record that is always updated before the changes are completed, many conflicts occur and programs tend to run serially. With a database having such a pattern of access, it is possible to increase throughput significantly and reduce response time by running jobs serially without concurrent access control at the page level.

A site option is also available (see System Operation Techniques manual) which sometimes allows the difficulties stated above to be overcome. This option treats all page requests as write requests whether the page is accessed for read or write. With this option no page is used simultaneously by several programs, thus the degree of concurrent use of the file is reduced, but conditions causing deadlock between programs are also reduced. Testing is the only certain way to determine if the site option is useful for a particular site. Need for testing is reduced, because FMS adjusts dynamically its algorithms on a file by file basis to treat all requests as write requests when the ratio of denials to delays is high. This adaptation is intended to prevent extreme cases when excessive denials cause program progress to be impeded.

Another critical factor in the successful use of the MONITOR option is the number of pages accessed between checkpoints. This option is most effective when several active programs are running concurrently, each accessing a small (5 to 20) number of pages between checkpoints. The first thing to consider is that the more pages accessed between checkpoints, the greater the likelihood of conflict. Second, FMS must search a list of reserved pages on each request; therefore, the shorter the list, the less overhead is incurred by the search.

The list of reserved pages resides in memory and is limited in length. The default for the maximum length of the list for any single file is 300 pages; that is, the total number of pages reserved by all programs using the same file must be less than or equal to 300 pages. The default value (300) can be increased or decreased by a site-option patch as described in the _System Operation Techniques_ manual. When the page limit is reached, denial status returns are issued to the programs. If all entries in the list apply to a single user, an error status return is issued, which usually causes the program to abort (see Appendix E).

SECTION VIII

FILSYS DIRECTIVES

FILSYS ACTIVITY

Cataloging services are requested by means of directives input to a $ FILSYS activity or by commands accepted by the Time Sharing Subsystem, ACCESS.

A FILSYS activity is initiated by use of a $ FILSYS control card (see typical deck setups for the FILSYS activities in Appendix C), along with appropriate control cards and FILSYS directive cards for the requested function. FILSYS activity cannot be used to obtain allocation to a file.

Two kinds of FILSYS activity directives are accepted, one from any user, and one from a privileged user.

The first kind are user directives that provide for file and catalog creation, modification, deletion, listing, and locking and are described in Section IX. User directives are submitted with a $ USERID control card to identify the user. If the FILSYS activity is to serve several users, USERID directives can be submitted in addition to or instead of the $ USERID control card to identify each user.

The second kind are privileged user directives that provide for System and Pack Master Catalog creation, modification, deletion, and listing and are described in Section X. Privileged user directives that provide for saving and restoring of files and catalogs for several or all users are described in Section XI. Privileged user directives are submitted with a $ PRIVITY control card which requests operator approval before the FILSYS activity can be executed.

DIRECTIVE FORMAT

The directives are provided as input data cards to the FILSYS activity. The format for directives is as follows: first the directive word or abbreviation, followed by one or more blanks, followed by a variable field. The directive may begin in any column and the variable field may be continued from one card to the next provided the preceding card ends with a comma or a slash. Only the first 72 columns of the card are read.

Typically, the variable field begins with a catalog or file name followed by one or more options separated from the catalog or file name and from one another by a comma or a comma and one or more blanks. For example, a File Create directive has the following format:

```
FC qualified file name, option,option, ...
```

Options consist either of a single word or a word identifying the option followed by a slash-enclosed list of words, codes, and/or numbers, each separated from one another by a comma or a comma and one or more blanks. For example, a File Create directive has options to express size, mode, device, etc.:

```
FC qualified file name, SIZE/number,number/,MODE/code/,DEVICE/name/
```

Options may appear in any order in the variable field; but, if a catalog or file name is required, that name must immediately follow the directive word or directive abbreviation and must precede any options.

DIRECTIVE DESCRIPTIONS

In the descriptions of FILSYS directives that follow (Sections IX, X, and XI), the title and format for the directive are given first, followed by the list of options applicable to the directive, separated into required and not required. The format is depicted using familiar COBOL conventions:

- braces ($\left\{ \right\}$) enclose choices, one of which must be supplied

- brackets ([ ]) enclose items that are optional
- underlines indicate required items
- ellipses (...) indicate repetition
- words in upper case are to appear literally as shown
- words in lower case are representations of the literal entries.

The formats and choices for all items are described in detail for each directive to simplify reference, except for three items: qualified names, passwords, and permissions. These are given below.

Qualified Names

A qualified file name consists of the name of one or more catalogs to which the file is subordinate followed by the name of the file, each name separated from the next by a slash. If the password is required for any name, it follows the name, separated by a dollar sign. The highest level catalog to which others are subordinate, the User Master Catalog (UMC), always is the first name and is the name of the user under which it is cataloged.

user name [$password] [/catalog name [$password]] .../file name[$password]

A qualified catalog name is written the same as a file name; the catalog name is the last name given instead of the file name.

Individual names are one through 12 characters long, where the characters can be digits, letters of the alphabet, periods, or dashes.

A short form notation to represent the UMC name can be used. The UMC name can be omitted when a qualified catalog or file name is required. If the variable field begins with a slash, the user name in the $ USERID control card or a USERID directive will be used as the UMC to qualify the file or catalog name.

[/catalog name [$password]] .../file name [$password]

When a file or catalog is being created, the last name in the qualified name is given without a password. Its password is specified by means of a password option. See the discussion "Names", Section II and "Passwords", Section III.

PASSWORDS

When a password is specified by means of the option PASSWORD/.../, the slash-enclosed list can consist of a single name or of one or more names each suffixed by a pair of times specifying the time interval during which the password applies. If a pair of times is given, it is introduced by a colon and the two times are separated by an asterisk.

```
        PASSWORD/name/

             or

        PASSWORD/name:time*time, .../
```

When a password is given in a qualified name, only the password that applies at the moment is given. See the discussion of "Passwords", Section III.

PERMISSIONS

When general permissions are specified, the permitted actions are named or abbreviated in any order, with commas separating one from the other.

```
        permitted action,...,permitted action
```

When specific permissions are specified, the permitted action is followed by a slash-enclosed list of names of users specifically given permission for that action, each name separated from the next by a comma. In addition to a permitted action, the word EXCLUDE may appear, indicating that users named after it have no permissions.

$$\left\{\begin{matrix}\text{EXCLUDE}\\ \text{permitted action}\end{matrix}\right\}\text{/user name,.../,}\left\{\begin{matrix}\text{EXCLUDE}\\ \text{permitted action}\end{matrix}\right\}\text{/user name,.../,...}$$

See the discussion of "Permissions", Section III.

MEANING OF OPTIONS

For the meaning of any option, consult the index for the discussion of the option in the text preceding this description of directives. For example, the meaning of the MODE/RAND/ or MODE/SEQ/ option is discussed in "Random or Sequential Files" under "Requesting Allocation", Section II.

USAGE NOTES

1. Comments can intervene between one card and another. Comment cards are indicated by an asterisk in column 1 of the card, with the comment following the asterisk.

2. If an error is encountered in processing any directive, the error is reported and only syntax checking of remaining directives is performed. An exception is that the LIST directive is honored so that one can determine the effect of directives processed prior to the error. Status messages, in addition to syntax errors, that can appear in a $ FILSYS activity report are listed in Appendix B.

3. Passwords are not shown on the listing of directives unless requested. Passwords are shown on listings produced for MASLST directive or CLIST with privity. Passwords on the MASLST listings can be suppressed, if so desired.

4. Six mode directives can be used to change the rules 2 and 3 above or to cancel the effect of a previously given mode directive. These directives must appear in columns 2-12 of a card, with a blank in column 1. A mode directive must precede (not necessarily immediately) the directive to which the mode directive applies.

   IGNORE ERRS — Causes directives after a detected error to continue to be processed. Allows catalog deletion to proceed even when an error is returned in deleting a subordinate. Allows a save or restore to be performed if a naming error is detected.

   NOTICE ERRS — Cancels the effect of a previous IGNORE ERRS directive.

   SYNTAX ONLY — Causes all directives to receive only checking of syntax, except for list directives which continue to be performed.

   USER PER PG — Causes the listing of catalogs and files for each user to be started on a separate page of a MASLST listing.

   SHOW PASSWDS — Causes passwords to appear in directive listings and cancels the effect of a previous HIDE PASSWDS.

   HIDE PASSWDS — Suppresses printing of passwords on MASLST or CLIST listings and reverses effect of a previous SHOW PASSWDS directive.

   A seventh mode directive requires that a $ PRIVITY control card be present in the job.

   TAKE SNAPS — Causes all MME GEFSYEs to be preceded by a snapshot of the calling sequence.

   An eighth mode directive is as follows:

   STOP SNAPS — Stops the taking of snapshots as requested by a previous TAKE SNAPS directive.

   The DUMP option on a $ FILSYS control card is not honored unless a $ PRIVITY control card is present in the job. The restrictions on snapshots and dumps prevent unauthorized disclosure of passwords.

SECTION IX

USER DIRECTIVES

This section describes user file and catalog directives.

NOTE: Devices to be used with these directives are in some cases restricted:

For Series 6000 Only	For Series 60 Only	For Series 6000 and Series 60
DSS167	MS0310	DSS180
DSS170	MS0400	DSS181
DSS270	MS0500	DSS190
167PK	310PK	DSS191
170PK	400PK	MS0450
BSS001	500PK	180PK
	MTS67	181PK
	MTS69	190PK
	TAPE27	191PK
	TAPE29	450PK
		TAPE
		TAPE7
		TAPE9
		MTS57
		MTS59

FILE DIRECTIVES

File Create

Format

```
( FCREAT )
{ FC     }  name, option(s)
( CF     )
```

Required

Qualified file name.

Options

1.
```
           ( name                 )
PASSWORD/  { name:time*time,...   }  /
```

No password is assumed if PASSWORD option is not specified.

2. Permission(s). None is assumed if permission(s) not specified.

3. Device or type of device this file is to be placed on:

```
         ( device name )
         | DSS167      |
         | DSS170      |
         | DSS180      |
         | DSS181      |
DEVICE/  { DSS190      }  /
         | DSS191      |
         | DSS270      |
         | BSS001      |
         | MS0310      |
         | MS0400      |
         | MS0450      |
         ( MS0500      )
```

Default specified for most immediate superordinate catalog is assumed or most available device is used. Device specification is ignored when file is subordinate to catalog on removable structured disk pack, for all files are to be placed on that pack.

4. Size of file:

```
( BLOCKS )
| LLINKS |                    [   ( UNLIMITED      ) ]
{ LINKS  }  /initial number   [ , {                } ]  /
( SIZE   )                    [   ( maximum number ) ]
```

A size of 1 link, initial and maximum, is assumed if size is not specified. UNLIMITED may be given to prevent any control being exercised over the size of the file, except that provided by a user maximum, if there is one. Any request for a maximum file size greater than 262,143 llinks is treated as UNLIMITED.

(BLOCKS = LLINKS = 320 words, LINKS = SIZE = 3840 words.)

5. $\text{MODE/}\left\{\begin{array}{l}\text{RAND}\\\text{SEQ}\end{array}\right\}\text{/}$

RAND is assumed for I-D-S/I, SEQ for others.

6. $\text{ACCESS/}\left\{\begin{array}{l}\text{NORMAL}\\\text{READ WHILE WRITE or RWW}\\\text{CONCURRENT}\\\text{MONITOR}\end{array}\right\}\text{/}$

NORMAL is assumed if ACCESS option is not specified. If MONITOR is specified, the ABORT/ROLLBACK/ option is assumed unless some other ABORT option is explicitly specified.

7. $\text{AUDIT/}\left\{\begin{array}{l}\text{NONE}\\\text{DENIED}\\\text{ALL}\end{array}\right\}\text{/}$

NONE is assumed if AUDIT option is not specified.

8. $\text{VERIFY/}\left\{\begin{array}{l}\text{NO}\\\text{YES}\end{array}\right\}\text{/}$

NO is assumed if VERIFY option is not specified.

9. $\text{ABORT/}\left\{\begin{array}{l}\text{NONE}\\\text{LOCK}\\\text{ROLLBACK}\end{array}\right\}\text{/}$

NONE is assumed if ABORT option is not specified.

10. $\text{RDERR/}\left\{\begin{array}{l}\text{NONE}\\\text{JOURNAL [,DUP [: device name or type]]}\\\text{DUP :[device name or type] [,JOURNAL]}\end{array}\right\}\text{/}$

NONE is assumed if RDERR option is not specified.

Device name or type are as for option 3, above. If not specified, devices different from devices file resides on are selected. DUP cannot be specified for a file subordinate to a catalog on removable structured disk pack.

11. INCRSAVE/ $\left\{\begin{matrix}\text{NO}\\\text{YES}\end{matrix}\right\}$ /

INCRSAVE/NO/ is assumed for an I-D-S/I file, one subordinate to a catalog on a removable structured disk pack, or a file whose initial size is more than 25 links (over 100,000 words or about 1/2 million bytes). For all other files, INCRSAVE/YES/ is assumed. However, either YES or NO can be specified for any file. A file on a removable structured disk pack is not saved, even though the disk pack is mounted at the time of the incremental save.

12. PAGESIZE/number of words in page/

If PAGESIZE option is not specified, 320 is assumed. Maximum page size is 4032.

Note

File is neither I-D-S/I nor does it already exist on magnetic tape or nonstructured disk pack.

Examples

1.
```
FCREAT  CLASS21/PROB1INPUT,READ
```

The file named PROB1INPUT is to be created directly under the UMC for the user CLASS21. The file is one link long, no growth is possible, either random or sequential access is allowed, and the file is placed on any device. Anyone can read the file but only the creator can write in it. No password is required. Only one writer or multiple readers, but not both, are to be allowed. No protection is required, but the file will be incrementally saved.

2.
```
CF  CORPORATE$SEPT25/CUSTOMER,PASSWORD/NOW:0800*1200,
THEN:1515*1630/,READ/AKC,DFD,RLN/,WRITE/TSE/,
LLINKS/200,200/,MODE/RAND/,INCRSAVE/NO/
```

A file named CUSTOMER is to be created directly under the UMC for user CORPORATE where the UMC has the password SEPT25. The file is given two passwords, one called NOW from 8 to 12 a.m. and the other called THEN from 3:15 to 4:30 p.m. Outside of these times, the file cannot be accessed. Specific permission to read is given three users, and write (which includes read) is given to one. The size of the file is 200 llinks and it is not allowed to grow, nor is sequential access allowed. It is not to be protected nor incrementally saved. NORMAL control over allocation is specified by default.

3.
```
CF  INSTALL/SYSA/DETAIL,LINKS/3000/,DEVICE/DSS190/,
RDERR/DUP:DSS181/,ABORT/ROLLBACK/,ACCESS/MONITOR/
```

A file, DETAIL, is to be created subordinate to a catalog, SYSA, which is indexed in the UMC for the user, INSTALL. The file is to have 3000 links assigned on a 190 disk. The file is to be duplicated on (several) 181 disks and protected against incomplete update by befores. FMS controlled concurrency is allowed. The file will not be incrementally saved (since it is larger than 25 links). Only the creator of the file can read from or write to the file, since no permissions are specified. No password is specified for the file.

File Modify

Format

$$\left\{\begin{array}{l}\text{FMOD}\\ \text{FM}\\ \text{MF}\end{array}\right\}\ \underline{\text{name}},\ \underline{\text{option}},\ \text{additional option(s)}$$

Required

1. Qualified file name, with password (if any).

2. At least one option.

Options

1. $\text{PASSWORD/}\left\{\begin{array}{l}\text{name}\\ \text{name:time*time,...}\end{array}\right\}\text{/}$

   PASSWORD without slash-enclosed password(s) specifies removal of existing password(s).

2. Permission(s)

3. Maximum size of file:

   $$\left\{\begin{array}{l}\text{BLOCKS}\\ \text{LLINKS}\\ \text{LINKS}\\ \text{SIZE}\end{array}\right\}\text{/}\left\{\begin{array}{l}\text{UNLIMITED}\\ \text{maximum number}\end{array}\right\}\text{/}$$

   (BLOCKS = LLINKS = 320 words, LINKS = SIZE = 3840 words.)

   The maximum number of llinks or links cannot be smaller than the current size of the file. UNLIMITED may be given to prevent any control being exercised over the size of the file, except that provided by a user maximum, if there is one. Any request for a maximum file size greater than 262,143 llinks is treated as UNLIMITED.

4. $\text{MODE/}\left\{\begin{array}{l}\text{RAND}\\ \text{SEQ}\end{array}\right\}\text{/}$

5. $\text{ACCESS/}\left\{\begin{array}{l}\text{NORMAL}\\ \text{READ WHILE WRITE or RWW}\\ \text{CONCURRENT}\\ \text{MONITOR}\end{array}\right\}\text{/}$

   Cannot be changed while file is busy. If MONITOR is specified, the ABORT/ROLLBACK/ option is assumed unless some other ABORT option is explicitly specified.

6. $$\text{AUDIT/}\left\{\begin{array}{l}\text{NONE}\\ \text{DENIED}\\ \text{ALL}\end{array}\right\}\ /$$

7. $$\text{VERIFY/}\left\{\begin{array}{l}\text{NO}\\ \text{YES}\end{array}\right\}\ /$$

Cannot be changed while file is busy.

8. $$\text{ABORT/}\left\{\begin{array}{l}\text{NONE}\\ \text{LOCK}\\ \text{ROLLBACK}\end{array}\right\}\ /$$

Cannot be changed while file is busy.

9. $$\text{RDERR/}\left\{\begin{array}{l}\text{NONE}\\ \text{JOURNAL [,DUP [:device name or type]]}\\ \text{DUP [:device name or type] [,JOURNAL]}\end{array}\right\}\ /$$

Cannot be changed while file is busy. Device name or type are as for option 3 of File Create. If not specified, devices different from the devices file resides on are selected. DUP device cannot be changed by File Modify. DUP cannot be specified for a file subordinate to a catalog on removable structured disk pack.

10. $$\text{INCRSAVE/}\left\{\begin{array}{l}\text{NO}\\ \text{YES}\end{array}\right\}/$$

11. PAGESIZE/number of words in page/

12. NEWNAM/unqualified name without password/

To specify new file name.

13. DELETE to specify removal of existing permission(s):

$$\text{DELETE/}\left\{\begin{array}{l}\text{[GEN'L,] user name,...,user name}\\ \text{user name,...,user name [,GEN'L]}\end{array}\right\}/$$

Where GEN'L specifies removal of all general permission(s) and user name specifies removal of all specific permission(s) for that user.

14. $$\text{RESET/}\left\{\begin{array}{l}\text{BUSY [,ABORT]}\\ \text{ABORT [,BUSY]}\end{array}\right\}/$$

Clears busy and/or abort indicators in file description. Clearing busy indicators also sets file non-null. This option is protected against unauthorized use; i.e., $ PRIVITY card is necessary.

Note

File is neither I-D-S/I nor does it already exist on magnetic tape or nonstructured disk pack.

Examples

1. `FMOD  CLASS21/PROB1INPUT,SIZE/3/`

   The maximum size of the file named PROB1INPUT cataloged directly under the user CLASS21,UMC is specified to be three links.

2.
   ```
   FM  CORPORATE$SEPT25/CUSTOMER$NOW,PASSWORD,
   NEWNAM/RELATIONS/,DELETE/AKC,DFD,RLN,TSE/,INCRSAVE/YES/
   ```

   Security protection is removed for the file named CUSTOMER cataloged under the UMC for user named CORPORATE (which has SEPT25 as password). The passwords for the file are removed, specific permissions are deleted, and the name of the file is changed to RELATIONS. The rejection of incremental saving for the file is reversed.

3. `MF  INSTALL/SYSA/DETAIL,LINKS/3210/,ACCESS/NORMAL/`

   A maximum size is specified for the file named DETAIL subordinate to a catalog named SYSA which is subordinate to the UMC for user INSTALL.

File Create - I-D-S/I File

Format

$$\left\{\begin{array}{l}\text{FCREAT/IDS/}\\ \text{FC/IDS/}\\ \text{CF/IDS/}\end{array}\right\}\underline{\text{name}}\text{, }\underline{\text{I-D-S/I specifications}}\text{, option(s)}$$

Required

1. Qualified file name.

2. I-D-S/I specifications.

   a. BASESIZE/total number of pages in database/

   b. RNG/page number of first page in this subfile, page number of last page in this subfile/

Options (in addition to those listed for File Create directive)

1. LINESPERPAGE/maximum number of records in page/

   If LINESPERPAGE option is not specified, 63 is assumed.

2. $\text{INVENTORY/}\left\{\begin{array}{l}\text{NO}\\ \text{percentage fill before page is}\\ \quad\text{inventoried as full}\end{array}\right\}\text{/}$

   If INVENTORY option is not specified, 75 is assumed.

3. $\text{MULTIUSER/}\left\{\begin{array}{l}\text{NO}\\ \text{YES}\end{array}\right\}\text{/}$

   If MULTIUSER option is not specified, NO is assumed.
   The option can also be spelled as MULTI-USER.

4. AREA/number/ assigned must be in the range 0-63.

5. $\text{COEXISTENCE/}\left\{\begin{array}{l}\text{NO}\\ \text{YES}\end{array}\right\}\text{/}$

   Provides batch mode support for I-D-S/I.

Refer to I-D-S/I User's Guide for a detailed description of these options.

Notes

1. ABORT/LOCK/ is assumed if ABORT/ROLLBACK/ is not specified. If the nonstandard ABORT/NONE/ configuration is desired, then a file modify request is required.

2. PAGESIZE for I-D-S/I must not be greater than 640 words.

Examples

1.
```
FCREAT/IDS/ SALTRIVER/ONE,SIZE/12000,12000/,
BASESIZE/235000/,RNG/1,144000/,INVENTORY/NO/,
ACCESS/MONITOR/
```

An I-D-S/I file is to be created with name ONE to be subordinate to UMC for user named SALTRIVER. The file size is to be 12,000 links, and growth is not allowed. This is a subfile of an I-D-S/I file with 235,000 pages of page size 320 words (assumed). Pages in this subfile are numbered 1 through 144,000. No inventory pages are to be provided. FMS protection against concurrent update and incomplete update are specified.

2.
```
FC/IDS/ SALTRIVER/TWO,LLINKS/91098,91098/,
BASESIZE/235000/,RNG/144001,235000/,INVENTORY/50/
```

Another I-D-S/I subfile with name TWO is to be created, again subordinate to UMC for user SALTRIVER. The file size is to be the remaining 91,000 pages (which are one llink apiece) in the base plus 98 llinks for the inventory which is specified for this subfile. Concurrent allocation is not allowed.

File Modify - I-D-S/I File

Format

$\left\{\begin{array}{l}\text{FMOD}\\\text{FM}\\\text{MF}\end{array}\right\}$ name, option, additional option(s)

Required

1. Qualified file name, with password (if any).

2. At least one option.

Options

1. I-D-S/I options:

   a. BASESIZE/total number of pages in database/

   b. RNG/page number of first page in this subfile,
      page number of last page in this subfile/

   c. LINESPERPAGE/maximum number of records in page/

   d. INVENTORY/ $\left\{\begin{array}{l}\text{NO}\\\text{percentage fill before page is inventoried as full}\end{array}\right\}$/

   e. MULTIUSER/ $\left\{\begin{array}{l}\text{NO}\\\text{YES}\end{array}\right\}$ /

   f. COEXISTENCE/ $\left\{\begin{array}{l}\text{NO}\\\text{YES}\end{array}\right\}$ /

      Provides batch mode support for I-D-S/I.

2. See options under File Modify directive.

Examples

1.
```
FMOD   SALTRIVER/ONE,LLINKS/144147/,INVENTORY/75/
```

   The I-D-S/I subfile, ONE, subordinate to UMC for SALTRIVER is modified to allow inventorying. The maximum size of the file is increased to allow room for the inventory pages.

2.
```
FMOD   SALTRIVER/TWO,ACCESS/MONITOR/
```

   The file, TWO, subordinate to UMC for SALTRIVER is modified to allow concurrent update and specify FMS control of concurrency. Notice that there is nothing in this directive to indicate that the file description being modified is an I-D-S/I file.

3.
```
FMOD   SALTRIVER/TWO,RESET/BUSY/
```

   The file, TWO, subordinate to UMC for SALTRIVER, is set not-busy.

File Create - Nonstructured Media

Format

```
⎧ FCREAT ⎫
⎨ FC     ⎬  name, device, option(s)
⎩ CF     ⎭
```

Required

1. Qualified file name.

2. Device; either:

   a. magnetic tape:

```
        ⎧ TAPE    ⎫
        ⎪ TAPE7   ⎪                          ⎡  ⎧ DEN0   ⎫ ⎤
        ⎪ TAPE9   ⎪                          ⎢  ⎪ DENSTD ⎪ ⎥
        ⎪ MTS57   ⎪                          ⎢  ⎪ DEN2   ⎪ ⎥     ⎧ FILEINFO ⎫
DEVICE/ ⎨ MTS59   ⎬,reel serial number       ⎢ ,⎨ DEN5   ⎬ ⎥/,   ⎩ VOLUME   ⎭ /reel name/
        ⎪ MTS67   ⎪                          ⎢  ⎪ DEN8   ⎪ ⎥
        ⎪ MTS69   ⎪                          ⎣  ⎩ DEN16  ⎭ ⎦
        ⎪ TAPE27  ⎪
        ⎩ TAPE29  ⎭
```

   DEN0, determine density from reel, is assumed if density is not specified.

   b. or nonstructured disk pack:

```
          ⎧ 167PK ⎫
          ⎪ 170PK ⎪
          ⎪ 180PK ⎪
          ⎪ 181PK ⎪                            ⎧ FILEINFO ⎫
DEVICE/   ⎨ 190PK ⎬ ,pack serial number/ ,    ⎩ VOLUME   ⎭ /pack name/
          ⎪ 191PK ⎪
          ⎪ 310PK ⎪
          ⎪ 400PK ⎪
          ⎪ 450PK ⎪
          ⎩ 500PK ⎭
```

Options

1.
```
PASSWORD/ ⎧ name                 ⎫ /
          ⎩ name;time*time,...   ⎭
```

   No password is assumed if PASSWORD option is not specified.

2. Permission(s).

   None is assumed if permission(s) not specified.

3.
```
MODE/ ⎧ SEQ  ⎫ /
      ⎩ RAND ⎭
```

   SEQ is assumed if MODE option is not specified.

Notes

1. File already exists on magnetic tape or nonstructured disk pack.

2. Reel or pack serial numbers can be 1 through 5 characters (letters or digits) used to identify reel or pack for mounting purposes.

3. Reel or pack name can be 1 through 12 characters (letters, digits, dashes, or periods) sometimes used to identify disk pack.

4. Reel, or pack serial number and name, and reel density are provided on allocation request for file.

5. No checks are made to verify if the specified device is configured on the system.

Examples

1.
```
FCREAT  J.P.JONES/EXPER-1,DEVICE/181PK,P0338/,VOLUME/EXPERIMENTS/
```

   A file already existing on a nonstructured disk pack is cataloged under UMC for J.P.JONES with name EXPER-1. The pack type, serial number, and name are specified.

2.
```
FCREAT  CHICAGO/PAY,DEVICE/TAPE9,13172,DEN16/,VOLUME/CHICAGO-PAY/
```

   A file already existing on magnetic tape is cataloged under UMC for CHICAGO with name PAY. The reel type, serial number, density, and name are specified.

File Modify - Nonstructured Media

Format

```
 ┌ FMOD ┐
<  FM    > name, option, additional option(s)
 └ MF   ┘
```

Required

1. Qualified file name, with password (if any).

2. At least one option.

Options

1.
```
           ┌ name               ┐
PASSWORD/ <                      > /
           └ name:time*time,... ┘
```

PASSWORD without slash-enclosed password(s) specifies removal of existing password(s).

2. Permission(s).

3.
```
       ┌ SEQ  ┐
MODE/ <        > /
       └ RAND ┘
```

4. NEWNAM/unqualified name without password/

To specify new file name.

5. DELETE to specify removal of existing permission(s):

```
         ┌ [GEN'L,] user name,...,user name ┐
DELETE/ <                                    > /
         └ user name,...,user name [,GEN'L] ┘
```

where GEN'L specifies removal of all general permission(s) and user name specifies removal of all specific permission(s) for that user.

Note

File already exists on magnetic tape or nonstructured disk pack.

Examples

1.
```
FM  J.P.JONES/EXPER-1,R/S.T.SCOTT/
```

   Read permission is given to S.T.Scott for the file (on a nonstructured disk pack). Note, however, that a file on nonstructured media can be accessed by anyone that knows the volume serial number simply by requesting volume rather than file allocation (unless physical control over the volume is exercised).

2.
```
FM CHICAGO/PAY,NEWNAM/ACCT3/,PASSWORD/PAY/
```

   The name of the file is changed to ACCT3 and a password is specified for the file.

File Delete

Format

$$\left\{\begin{array}{l}\text{FPURGE}\\ \text{FP}\\ \text{PF}\end{array}\right\}\ \underline{\text{name}}$$

$$\left\{\begin{array}{l}\text{FRELES}\\ \text{FR}\\ \text{RF}\end{array}\right\}\ \underline{\text{name}}$$

Required

Qualified name of file to be deleted, with password (if any).

Options

None

Notes

1. If the file is on a removable structured disk pack, that pack must be mounted.

2. If the directive is FPURGE, the space currently assigned to the file is overwritten with zeros before being returned for other assignment.

3. If the directive is FRELES, the space currently assigned to the file is returned for other assignment without being overwritten.

4. If the file is on magnetic tape or nonstructured disk pack, the space is neither overwritten nor returned for other assignment.

5. If the file is busy, delete is delayed until the file is no longer busy. Reference by name is no longer possible, however, since reference to its file description is deleted in the catalog the file is subordinate to, and the amount of space in use for the user is decreased by the size of the file even though space release occurs only when the file is no longer busy.

Examples

1. FR CLASS21/PROB1INPUT

   The file description for PROB1INPUT and entry in CLASS21, the catalog for this file, are deleted and the space assigned to the file is released for subsequent assignment without being overwritten.

2. FP CORPORATE$SEPT25/CUSTOMER$NOW

   The file description for CUSTOMER and entry in CORPORATE, the catalog for this file, are deleted and the space assigned to the file is overwritten with zeros. The space is then released for subsequent assignment.

3. FP J.P.JONES/EXPER-1

   The file description for EXPER-1 and entry in J.P.JONES, the catalog for this file, are deleted. Since the file is on nonstructured media, the space is neither overwritten with zeros (despite the request to purge) nor is it released for subsequent assignment.

CATALOG DIRECTIVES

Catalog Create

Format

```
    { CCREAT }
    { CC     }    name, option(s)
```

Required

Qualified catalog name.

Options

1.
```
    PASSWORD/ { name                 } /
              { name:time*time,...   }
```

No password is assumed if PASSWORD option is not specified.

2. Permission(s)

3. Device; either:

   a. placement of this catalog, which must be the UMC or a subordinate of the UMC, and confinement of all subordinate files and catalogs on structured disk pack:

```
              { 167PK }
              { 170PK }
              { 180PK }
    DEVICE/   { 181PK } ,pack serial number/
              { 190PK }
              { 191PK }
              { 310PK }
              { 400PK }
              { 450PK }
              { 500PK }
```

   b. or default placement of subordinate files (but not this catalog or any subordinate ones) on named device or specified device type:

```
              { device name }
              { DSS167      }
              { DSS170      }
              { DSS180      }
    DEVICE/   { DSS181      } /
              { DSS190      }
              { DSS191      }
              { DSS270      }
              { MS0310      }
              { MS0400      }
              { MS0450      }
              { MS0500      }
              { BSS001      }
```

Notes

1. Device name is a 3-character name assigned by the installation to each disk or other mass storage device.

2. For a create of the User Master Catalog (UMC), the name of the catalog must be the name of the user under which the catalog is to be created.

3. When a catalog is to be created on a removable structured disk pack (if the pack is mounted), both a reference to the catalog on the fixed device holding catalogs for the user and the catalog on the pack are created. If the pack is not mounted, only the reference is created. By not mounting the pack, then, linkage to an existing catalog on the pack can be provided in a computer to which the pack is being transported. When a catalog is to be created on a removable structured pack that is not mounted, neither passwords nor permissions can be specified, since these cannot be included with the catalog reference. The catalog reference is all that is created when the pack is not mounted.

4. The use of the DEVICE option is illegal on any level of catalog structure subordinate to a catalog on a removable structured disk pack.

Examples

1.
```
CCREAT  CORPORATE,PASSWORD/SEPT25/,DEVICE/181PK,P3715/
```

A UMC is to be created for the user named CORPORATE on a removable structured disk pack whose type and serial number is given. The UMC is given the password SEPT25.

2.
```
CC  XLNT/FAST,READ,WRITE/NSGD,AEWC/,DEVICE/DSS190/
```

A catalog is to be created subordinate to the UMC for user named XLNT. The name of the catalog is FAST. General permission to read and specific permission to write for two named users are specified. The 190 is specified as a default type of device on which to assign subordinate files.

3.
```
CC  J.P.JONES/RMVBL,DEVICE/181PK,P3715/,READ,WRITE
```

A catalog is to be created subordinate to the UMC for J.P.JONES. The name of the catalog is RMVBL, and it is to be created on a removable structured disk pack whose type and serial number are given. General permissions are specified. If the named pack is mounted, a reference to RMVBL is created in the UMC for J.P.JONES giving the pack serial number, and then the catalog RMVBL itself is created on the pack. (The user J.P.JONES must be authorized use of the pack first by means of a CRPACK directive naming the user.) If the pack is not mounted at the time of the catalog create, only the reference is created. Refer to Example 2 under the Catalog Delete directive for an example of how the reference is used.

Catalog Modify

Format

$$\left\{\begin{array}{l}\text{CMOD}\\ \text{CM}\\ \text{MC}\end{array}\right\}\ \underline{\text{name}},\ \underline{\text{option}},\ \text{additional option(s)}$$

Required

1. Qualified catalog name, with password (if any).

2. At least one option.

Options

1. $\text{PASSWORD/}\left\{\begin{array}{l}\text{name}\\ \text{name:time*time,...}\end{array}\right\}\ /$

   PASSWORD without slash-enclosed password(s) specifies removal of existing password(s).

2. Permission(s).

* 3. NEWNAM/unqualified name without password/

   To specify new catalog name.

4. DELETE to specify removal of existing permission(s):

   $\text{DELETE/}\left\{\begin{array}{l}\text{[GEN'L,]user name,...,user name}\\ \text{user name,...,user name[,GEN'L]}\end{array}\right\}\ /$

   where GEN'L specifies removal of all general permission(s) and user name specifies removal of all specific permission(s) for that user.

Notes

1. If the catalog is on a structured disk pack, the pack must be mounted prior to start of the job; a "no mount" message is issued to the operator if a $ nnnPK control card is not included with the $ FILSYS activity. *

2. The name of a User Master Catalog cannot be modified.

Examples

1. CMOD CORPORATE,PASSWORD

   The UMC for CORPORATE is modified to delete its password.

2. MC J.P.JONES/RMVBL,DELETE/GEN'L/ *

   The catalog RMVBL, subordinate to the UMC for J.P.JONES, is modified to delete all general permissions.

Catalog Delete

Format

$$\left\{\begin{array}{l}\texttt{CPURGE}\\ \texttt{CP}\\ \texttt{PC}\end{array}\right\}\ \underline{\texttt{name}}$$

$$\left\{\begin{array}{l}\texttt{CRELES}\\ \texttt{CR}\\ \texttt{RC}\end{array}\right\}\ \underline{\texttt{name}}$$

Required

Qualified catalog name of catalog to be deleted, with password (if any).

Options

None

Notes

1. Deletion of catalog includes deletion of files and/or catalogs subordinate to the catalog. If some subordinates are on removable structured disk pack, however, whether that pack is mounted or not, a delete is required of the highest level catalog on the pack to delete it and all subordinates on the pack.

2. If the catalog named is on a removable structured disk pack, that pack must be mounted.

3. If the directive is CPURGE, the space currently assigned to each subordinate file that is being deleted is overwritten with zeros before being returned for other assignment.

4. If the directive is CRELES, the space currently assigned to each subordinate file that is being deleted is returned for other assignment, without being overwritten.

5. If a subordinate file being deleted is on magnetic tape or nonstructured disk pack, the space is neither overwritten nor returned for other assignment.

6. If a subordinate file to be deleted is busy, the delete is delayed until the file is no longer busy. Reference by name to such a delayed delete file is no longer possible, however, since reference to its file description is deleted whether the file is busy or not.

7. If this directive is submitted by the creator of the catalog or privileged job, subordinate files are deleted whether or not purge permission is granted. Otherwise, deletion of subordinate files is contingent on purge permission for each file.

8. If an error is encountered in deleting a subordinate, other subordinates are deleted but the catalog is not (nor are any intervening catalogs). If the error is not one of denied permission, the mode directive IGNORE ERRS may be used to delete the catalog(s). The mode directive does not cause any space assigned to subordinate files to be located and released.

Examples

1. CR INSTALL/SYSA

   The catalog SYSA, the entry in the catalog INSTALL to which SYSA is subordinate, and the file description for DETAIL, a file subordinate to SYSA (refer to Example 3 under the File Create directive) are deleted, and the space assigned to the file (including space assigned to duplicate the file) is released without being overwritten with zeros.

2. CR J.P.JONES

   The UMC for user J.P.JONES, catalogs, and file descriptions subordinate, either immediately or mediately, to intervening catalogs are deleted and spaces assigned to files on fixed devices are overwritten with zeros and then released. The catalog RMVBL subordinate to the UMC (refer to Example 3 under Catalog Create directive) is not deleted, however, nor are any spaces assigned to files subordinate to this catalog zeroed or released. Once the UMC is deleted, of course, no reference to the catalog RMVBL or its subordinates is possible until the UMC and (reference to) the catalog are re-created, perhaps at a different installation. Both of these can be created by performing the Catalog Create directive used to create the RMVBL catalog originally but without the pack being mounted. (Because the pack is not mounted, only a reference to the catalog RMVBL is entered in the UMC for J.P.JONES.)

Catalog or File List

Format

```
{CLIST}
{LIST }  name, option(s)
```

Required

Qualified catalog or file name of catalog or file to be listed, with password (if any).

Options

1.
```
          { ALL  }
LISTOPT/  { ONLY }  /
          { BUSY }
```

ALL is assumed if the LISTOPT option is not specified. If ALL is assumed or specified, all files and/or catalogs subordinate to the named catalog are listed. If ONLY is specified, only files and/or catalogs immediately subordinate to the named catalog are listed. If the name is that of a file, only the file is listed. If BUSY is specified, only the busy file(s) is listed.

2.
```
          { NORM }
EXTENTS/  { ALL  }  /
          { NONE }
```

NORM is assumed if EXTENTS option is not specified. If NONE is assumed or specified, no file extents are shown. If ALL is specified, the file extents are shown in all cases. If NORM is specified, the file extents are shown as described in Appendix A.

3.
```
         { device name,...                 }
DEVICE/  { pack type, pack serial number   }  /
```

Notes

1. If the named catalog or file is on a removable structured disk pack, the pack must be mounted. If the named catalog has subordinate(s) on removable structured disk pack(s), the subordinates are referenced but detail and further subordinate information is not provided whether or not the pack(s) are mounted.

2. See Appendix A for descriptions of lists produced.

Examples

1. LIST SALTRIVER

   Information is printed not only about the UMC for the user named SALTRIVER but also about all file descriptions and catalogs for that user. The option LISTOPT/ALL/ is assumed. Note that only information about file descriptions, and not any contents of files, is printed.

2. LIST INSTALL,LISTOPT/ONLY/

   Information is printed about the UMC for the user INSTALL and about catalogs and file descriptions immediately subordinate to the UMC. The option ONLY prevents information about items subordinate to catalogs referenced in the UMC from being printed. For example, the file description for the file INSTALL/SYSA/DETAIL is not listed since it is immediately subordinate not to the UMC but to the catalog SYSA.

3. LIST J.P.JONES/RMVBL

   Information is printed about the catalog RMVBL that is on a removable structured disk pack (refer to Example 3 under the Catalog Create directive) and about all subordinates in that catalog. The pack holding RMVBL must be mounted.

4. LIST J.P.JONES

   Information is printed about the UMC for J.P.JONES and subordinates to it on fixed devices. The reference to the subordinate catalog RMVBL (see above) is printed but not any other information about that catalog, even though the pack it is on is mounted.

5. LIST J.P.JONES/EXPER-1,LISTOPT/ALL/

   Information about the file description for EXPER-1 is printed. The file description is on a fixed device even though the file is on a nonstructured, removable disk pack (refer to Example 1 under the File Create - Nonstructured Media directive). The specification LISTOPT/ALL/ is ignored since the item named is a file.

Catalog or File Lock

Format

$$\text{SLOCK}\quad \underline{\text{catalog or file name}},\ \left\{\begin{matrix}\text{ON}\\ \underline{\text{OFF}}\end{matrix}\right\}$$

$$\text{ALOCK}\quad \underline{\text{file name}},\ \left\{\begin{matrix}\text{ON}\\ \underline{\text{OFF}}\end{matrix}\right\}$$

$$\text{TDSLOCK}\ \underline{\text{file name}},\ \left\{\begin{matrix}\text{ON}\\ \underline{\text{OFF}}\end{matrix}\right\}$$

Required

1. Qualified catalog or file name including password for SLOCK, name of file including password for ALOCK.

2. One of the two options:

   a. ON to set the lock and restrict allocation.

   b. OFF to reset the lock and remove allocation restriction.

Options

None

Notes

1. SECURITY LOCK (SLOCK) restricts allocation of the file, or any files subordinate to the catalog, to users who have LOCK permission for or are the creators of the file or catalog, respectively.

2. ABORT LOCK (ALOCK) restricts allocation of the file to jobs seeking RECOVERY or QUERY allocation.

3. The SECURITY LOCK can be set or reset only by a user who has LOCK permission.

4. The ABORT LOCK and TDS LOCK can be set or reset only by a user who has RECOVERY permission.

5. An ABORT LOCK can be set or reset other than by this directive.

   a. It is set when an I-D-S/I file or one for which ABORT/LOCK/ has been specified is written to by a job that fails to terminate normally, either because of job or system failure. The abort lock is also set on failure of protection against incomplete updating specified as ABORT/ROLLBACK/. Finally, the abort lock may be set by means of a MME from a job with user name that has RECOVERY permission. *

   b. An abort lock can also be reset by a MME from a job with USERID that has RECOVERY permission. Another way an abort lock can be reset is by a File Modify with the RESET/ABORT/ option.

6. The security lock can only be set and reset by means of the SLOCK directive or a MME from a job with LOCK permission.

7. The TDSLOCK is set during system or TDS failure on ACCESS/MONITOR/ files currently in use by TDS and is reset automatically by normal termination of a TDS user with recovery allocation to the file.

Examples

1. SLOCK CORPORATE$SEPT25,ON

   The UMC for CORPORATE is security locked. This prevents access to subordinate files by all users except those using the name CORPORATE or any who have been given specific LOCK (and hence UNLOCK) permission.

2. ALOCK SALTRIVER/TWO,OFF

   The I-D-S/I file TWO, subordinate to the UMC for SALTRIVER, (refer to Example 2 under the File Create - I-D-S/I directive) is made available for access by removing an abort lock on the file. (If the file is not abort locked, the directive has no effect.)

User Identification

Format

USERID user name$log-on password

Required

Name of user and log-on password.

Options

None

Notes

1. The USERID directive is only required when files and/or catalogs cataloged under more than one user are being referenced. When only those cataloged under a single user are being referenced, the user name for it may be provided on a $ USERID control card rather than on the USERID directive.

2. When references are made to files and/or catalogs cataloged under a user other than the one named on the $ USERID control card, a USERID directive should precede the references. If more than one USERID directive is given, only the most recent is effective. Thus, if a reference is made again to user A, for example, after a reference to user B intervenes, three USERID's are required - one for user A, then one for user B, and then one for user A again.

3. No USERID is required for Master or Pack Create, Modify, Delete or List directives, however, even though catalogs for several users are being referenced. When individual files and/or catalogs are referenced (even though privileged directives have been used on the activity), a USERID directive is required.

Catalog Position/Release/Add

Format

```
CPOS   catalog name

CREL

CADD   catalog name
```

Required

1. To position to a specific catalog, the name of the catalog with its password (if any) is required with the CPOS directive.

2. To cancel this positioning, only the CREL directive is required.

3. To concatenate to any catalog name already referenced, the name to be added to the catalog name, with password (if any), is required with the CADD directive.

Options

None

Notes

1. The CPOS directive may be used to name a catalog to which files and/or catalogs referenced on subsequent directives are subordinate.

2. Once presented, the catalog name on the current CPOS directive is used to qualify any file and/or catalog names on File and Catalog Create, Modify, List, and Delete directives.

3. The CADD directive may be used to concatenate to any names already referenced.

4. The catalog name remains as a qualifier until it is replaced by the one on a subsequent CPOS directive, extended by means of a CADD directive, or is cancelled by a CREL directive.

5. Any privileged master directive, save directive, or restore directive also cancels the saved catalog name.

6. Use of other cataloging directives containing catalog or file names beginning with an implicit UMC name (/) will cancel a current CPOS directive.

Examples

1. The directives

```
CPOS  SWEETWATER/POWER$GEN4
FCREAT SUBSTATION
CCREAT ACCOUNT
FCREAT ACCOUNT/PROPERTY
FPURGE WELLS$DEEP6
```

are equivalent to the directives

```
FCREAT SWEETWATER/POWER$GEN4/SUBSTATION
CCREAT SWEETWATER/POWER$GEN4/ACCOUNT
FCREAT SWEETWATER/POWER$GEN4/ACCOUNT/PROPERTY
FPURGE SWEETWATER/POWER$GEN4/WELLS$DEEP6
```

2. In the example above, inserting a CREL directive just ahead of the FPURGE directive would cause the purged name to be unqualified, or inserting a new CPOS SWEETWATER/IRRIGATION$WETU just ahead of the FPURGE directive would cause the FPURGE file name to be:

```
SWEETWATER/IRRIGATION$WETU/WELLS$DEEP6
```

3. In the example above, inserting a CADD IRRIGATION$WETU directive just ahead of the FPURGE directive would cause the FPURGE file name to be:

```
SWEETWATER/POWER$GEN4/IRRIGATION$SWETU/WELLS$DEEP6
```

Pack Catalog Reference Delete

Format

```
RPURGE name
```

Required

Qualified catalog name of catalog to be deleted, with password (if any).

Options

None

Note

RPURGE can be used to remove reference to a subordinate and can be used when the subordinate is on a structured pack that has been removed or is otherwise inaccessible. The directive can also be used to remove an error to allow catalog deletion to proceed. To use this directive, the name of the file or catalog the reference to which is to be removed is named after the word RPURGE.

Example

```
RPURGE CORPORATE/CAT-A
```

The 3-word entry for the CAT-A subcatalog is zeroed in the SMC's UMC for CORPORATE, but is left intact in the PMC.

SECTION X

PRIVILEGED DIRECTIVES

This section describes privileged master and pack directives. A job with these directives must contain a $ PRIVITY control card and must be authorized by the operator before the job can be processed.

PRIVILEGED MASTER DIRECTIVES

Master Create

Format

```
CRMAST user name, PASSWORD/password/,option, additional option(s)
```

Required

1. Name of user for whom authorization is being granted to perform one or more of the following actions under his own user name:

   a. use cataloged files

   b. create catalogs and/or cataloged files

   c. use Time Sharing System

   d. initiate batch jobs from time sharing

2. Log-on password for the user:

   PASSWORD/password/

Options

1. Specification of the limit of the space on fixed devices that files and catalogs cataloged under the user name can occupy:

$$\left\{\begin{array}{l}\text{BLOCKS}\\ \text{LLINKS}\\ \text{LINKS}\\ \text{SIZE}\end{array}\right\} / \left\{\begin{array}{l}\text{UNLIMITED}\\ \text{maximum number}\end{array}\right\} /$$

   Numbers can be one through six digits. If not specified or specified zero, the default size of 1 llink is assumed. Catalogs and files can be created on removable structured disk packs and files on nonstructured media can be cataloged, however, as long as a catalog space of one llink is not exceeded. Use of cataloged files or creation of files cataloged under another user name is not prohibited.

   (BLOCKS = LLINKS = 320 words, LINKS = SIZE = 3840 words.) Any request for a maximum file size greater than 262,143 llinks is treated as UNLIMITED.

2. Specification of the limit on time sharing resources that can be expended under the user name:

   RESOURCES/maximum number of dollars/

   Number can be one through six digits. If not specified or specified zero, use of time sharing under the user name is denied and any of the remaining options are not applicable.

Options

If time sharing use is authorized, the following options can be used to control it.

1. To authorize initiation of batch job from Time Sharing System:

   CARDIN/ $\begin{Bmatrix}\text{NO}\\ \text{YES}\end{Bmatrix}$ /

   NO is assumed if CARDIN option is not specified.

2. If CARDIN/YES/ is specified, maximum allowable urgency for initiated batch job may be specified:

   URGENCY/ $\begin{Bmatrix}\text{NO}\\ \text{code}\end{Bmatrix}$ /

   Code can be a 2-digit number from 05 through 40. NO is equivalent to 05. Highest urgency that prevents job from being swapped out is 40.

3. The ability of a time sharing initiated batch job to initiate direct access connection with the terminal is authorized by:

   TALK/ $\begin{Bmatrix}\text{NO}\\ \text{YES}\end{Bmatrix}$ /

   NO is assumed if TALK option is not specified. TALK authorization requires CARDIN authorization, but not vice versa.

4. To enable debugging and tracing of user time sharing subsystem:

   LODX/ $\begin{Bmatrix}\text{NO}\\ \text{YES}\end{Bmatrix}$ /

   NO is assumed if LODX option is not specified. If YES is specified, user may load a program from a cataloged file to be used as a time sharing subsystem for his terminal. The program may be patched, traced, or saved.

5. To authorize tracing of an existing time sharing subsystem:

   LODS/ $\begin{Bmatrix}\text{NO}\\ \text{YES}\end{Bmatrix}$ /

   NO is assumed if LODS option is not specified. If YES is specified, user may load a specified time sharing subsystem and bind it with a trace program to facilitate debugging of that subsystem.

Examples

1. CRMAST CLASS21,PASSWORD/KIRK/,LLINKS/25/

   An SMC entry is to be created for the user named CLASS21. The user password KIRK is specified. The allowable space for this user on fixed devices is 25 llinks. No use of Time Sharing System is authorized.

2. CRMAST SALTRIVER,PASSWORD/1902/,SIZE/UNLIMITED/,
   RESOURCES/2500/,TALK/YES/,CARDIN/YES/,LODX/YES/

   An SMC entry is to be created for the user named SALTRIVER with password 1902. The space allowable on fixed devices is specified to be unlimited, and time sharing resources of $2500 is specified. Direct access from a batch job to the terminal is allowed as well as required use of the CARDIN subsystem. Use of LODX subsystem is also allowed.

3. CRMAST CORPORATE,PASSWORD/LUV/

   An SMC entry for the user CORPORATE is to be created with user password LUV. No space is allowed on fixed devices for this user, and no use of Time Sharing System is authorized.

4. CRMAST J.P.JONES,PASSWORD/110726/,SIZE/250/,RESOURCES/300/

   An SMC entry for J.P.JONES is to be created with user password 110726. Space is limited to 250 links on fixed devices, and time sharing resources to $300. No use of CARDIN or LODX is authorized.

Master Modify

Format

MODMAS user name, option, additional option(s)

Required

1. Name of user to identify which SMC entry is to be modified.

2. At least one option.

Options

1. PASSWORD/password/ to specify a new log-on password for the user.

2. $\left\{\begin{array}{l}\text{BLOCKS}\\\text{LLINKS}\\\text{LINKS}\\\text{SIZE}\end{array}\right\} / \left\{\begin{array}{l}\text{UNLIMITED}\\\text{maximum number}\end{array}\right\} /$

   To change the limit on fixed device space that catalogs and/or files can occupy. The maximum cannot be smaller than the amount of space currently occupied.

   (BLOCKS = LLINKS = 320 words, LINKS = SIZE = 3840 words). Any request for a maximum file size greater than 262,143 llinks is treated as UNLIMITED.

3. RESOURCES/maximum number of dollars/

   To change the limit on time sharing resources the user may spend.

4. $\text{CARDIN/}\left\{\begin{array}{l}\text{NO}\\\text{YES}\end{array}\right\}/$

5. $\text{URGENCY/}\left\{\begin{array}{l}\text{NO}\\\text{code}\end{array}\right\}/$

   Code is a 2-digit number, from 05 through 40. NO is equivalent to a code of 05.

6. $\text{TALK/}\left\{\begin{array}{l}\text{NO}\\\text{YES}\end{array}\right\}/$

7. $\text{LODX/}\left\{\begin{array}{l}\text{NO}\\\text{YES}\end{array}\right\}/$

8. $\text{LODS/}\left\{\begin{array}{l}\text{NO}\\\text{YES}\end{array}\right\}/$

Note

See options under Master Create directive for more information on options.

Examples

1. MODMAS CLASS21,LLINKS/60/,RESOURCES/300/

   The SMC entry for user named CLASS21 is modified to increase allowable space to 60 llinks and to allow use of Time Sharing System.

2. MODMAS SALTRIVER,SIZE/30000/,LODX/NO/

   The SMC entry for SALTRIVER is modified to limit space on fixed devices to 30,000 links and to delete authorization for LODX.

3. MODMAS CORPORATE,PASSWORD/SEPT25/

   The SMC entry for CORPORATE is modified to specify a new user password.

4. MODMAS J.P.JONES,CARDIN/YES/,URGENCY/25/

   The SMC entry for J.P.JONES is modified to authorize use of the CARDIN time sharing subsystem with a maximum urgency of 25.

Master Delete

Format

$$\left\{\begin{array}{l}\text{DELMAS}\\ \text{RELMAS}\end{array}\right\}\quad \underline{\text{user name}}$$

Required

Name of user to be deleted.

Options

None

Notes

1. A Master Delete directive is similar to a Catalog Delete directive except that, in addition, the master catalog entry for the user is also deleted. See the notes for Catalog Delete directive.

2. The DELMAS directive is used to cause subordinate files to be overwritten with zeros, as CPURGE does for Catalog Delete. The RELMAS directive, like the CRELES, causes space to be returned without being overwritten.

Examples

1. RELMAS CLASS21

   The SMC for the user named CLASS21, the UMC for the user, and all subordinate file descriptions and catalogs (on fixed devices) are released and spaces assigned to files cataloged under this user are released for subsequent assignment without being overwritten with zeros.

2. DELMAS J.P.JONES

   The SMC for J.P.JONES, the UMC, and all subordinates on fixed devices are released and spaces assigned to files on fixed devices are overwritten with zeros and then released for subsequent assignment. Files subordinate to J.P.JONES/RMVBL, which are on a removable structured disk pack (refer to Example 3 under the Catalog Create directive) and the file J.P.JONES/EXPER-1, which is on a nonstructured disk pack (refer to Example 1 under the File Create - Nonstructured Media directive) are not zeroed or released. The reference to RMVBL in the UMC is removed when the UMC is deleted, of course, as is the file description for the file EXPER-1, but neither catalogs, file descriptions, and files on the removable structured disk pack nor the file on the nonstructured disk pack are affected.

Master List

Formats

MASLST option(s)

MASLST user name, option(s)

Required

User name for format 2.

Options

1. ONLINE specifies that the Statistical Collection File records produced by the ACCOUNT directive be also produced here. A record is produced showing space usage for fixed devices and all mounted removable structured packs for each user (format 1) or for the named user (format 2). See Catalog and File Space Accounting directive below for more information.

2. LISTOPT/ $\left\{\begin{array}{c}\text{ALL}\\ \text{ONLY}\\ \text{BUSY}\end{array}\right\}$ /

   ALL is assumed if LISTOPT is not specified. If ALL is specified or assumed, the System Master Catalog (SMC) entry and all subordinate catalogs and files are listed for each or the one user. If ONLY is specified, only the SMC entry is listed for each or the one user. If BUSY is specified, all busy files are listed for each or the one user.

3. RANGE/32nd,32nd/

   Only files and catalogs or SMC entries for user hashing to the 32nds in the specified, inclusive range are to be listed. For example, RANGE/0,5/ specifies that any listing for users hashing to 32nds other than the first six is to be suppressed. This is useful in conjunction with SAVEMAST or RESTOREMAST with RANGE option or to allow several jobs to concurrently list all users. More than one RANGE/.../ option may be given to specify several ranges.

4. EXTENTS/ $\left\{\begin{array}{c}\text{ALL}\\ \text{NONE}\\ \text{NORM}\end{array}\right\}$ /

   NORM is assumed if EXTENTS is not specified. EXTENTS/ALL/ specifies that file extents are shown in all cases. EXTENTS/NONE/ specifies that no file extents are shown. EXTENTS/NORM/ specifies that file extents are shown as described in Appendix A.

5. DEVICE/ $\left\{\begin{array}{l}\text{ALL}\\ \text{device name,...}\\ \text{pack type, pack serial number}\end{array}\right\}$ /

   If the device names exceed ten in number, ALL should be specified. In addition to the reports produced, this option produces an additional list of users on the device(s) specified.

Notes

1. The first format specifies that all users are to be considered. The second format specifies that only the named user is to be considered.

2. If no name is given, a summary is produced after the listing of all catalogs and files. The summary shows space assigned to files for each user and the total space by SMC section. In addition, a device utilization report is produced.

3. Appendix A contains descriptions of reports produced by this directive. Note the effect of the mode directives USER PER PG and HIDE PASSWDS, described under "Usage Notes", Section VIII.

4. If an error occurs while attempting to retrieve an SMC entry, FILSYS attempts recovery by skipping to the next SMC range. In addition, the FILSYS buffer and calling sequence parameters are snapped as if there were an implied TAKE SNAPS directive (see "Usage Notes" in Section VIII).

Examples

1. MASLST

   A listing is provided of all catalogs and file descriptions (except those on removable structured disk packs) for all users.

2. MASLST CORPORATE

   A listing is provided of all catalogs and file descriptions for the user named CORPORATE.

3. MASLST LISTOPT/ONLY/

   A listing is provided of the SMC entry for each user.

4. MASLST CORPORATE,LISTOPT/ONLY/

   A listing is provided of the SMC entry for the user named CORPORATE.

5. MASLST ONLINE

   In addition to providing a listing of all catalogs and file descriptions for all users, accounting records are produced for each user showing space x time usage on fixed devices and all mounted removable structured disk packs.

6. MASLST J.P.JONES,LISTOPT/ONLY/,ONLINE

   In addition to the SMC for the user J.P.JONES being printed, accounting records are produced showing the space x time usage of J.P.JONES on fixed devices and all mounted removable structured disk packs. (One record is produced for the fixed devices and one for each structured pack that is mounted.)

7. MASLST LISTOPT/ALL/

   Same as Example 1.

8. MASLST CORPORATE,LISTOPT/ALL/

   Same as Example 2.

9. MASLST RANGE/0,3/,RANGE/7/

   A listing is provided of all catalogs and files for users hashing to the sections specified: 0, 1, 2, 3, and 7.

10. MASLST LISTOPT/ONLY/,RANGE/17,31/

    A listing is provided of the SMC entries for users hashing to the sections specified: any higher than 16.

Master Duplicate File Lock

Format

```
DLOCK file name, PASSWORD/password,option, additional option(s)
```

Required

1. Name of file to be processed.

2. Log-on password for the user.

3. At least one of each option.

Options

1. To set or turn off file copy defective indicator:

   ON
   OFF

2. To set or reset primary file copy defective indicator:

   COPYA

3. To set or reset secondary file copy defective indicator:

   COPYB

Example

```
DLOCK CORPORATE/OCTOBER$1357,ON,COPYA
```

The primary file copy defective indicator is set in order to use the secondary file copy in normal processing.

Catalog And File Space Accounting Directive

Format

```
            ┌ ONLINE        ┐
    ACCOUNT ┤               ├
            └ DEVICE/.../   ┘
```

Required

1. Either ONLINE or DEVICE/.../ is required.

2. ONLINE specifies that space accounting is to be performed for fixed devices and for each removable structured disk pack that is mounted.

3. The DEVICE option specifies that space accounting is to be done only for the specified pack(s):

```
                ┌ 167PK ┐
                │ 170PK │
                │ 180PK │
                │ 181PK │
        DEVICE/ ┤ 190PK ├  ,pack serial number,
                │ 191PK │
                │ 310PK │
                │ 400PK │
                │ 450PK │
                └ 500PK ┘
                ┌ 167PK ┐
                │ 170PK │
                │ 180PK │
                │ 181PK │
                ┤ 190PK ├  ,pack serial number,.../
                │ 191PK │
                │ 310PK │
                │ 400PK │
                │ 450PK │
                └ 500PK ┘
```

The specified pack(s) must be mounted.

Notes

1. One or more Statistical Collection File records are produced for each user, with space on fixed devices and/or on the specified packs. For the ONLINE option, if a user has space on fixed devices, a record is produced. If a user has space on any mounted removable structured disk packs, one record is produced for each such pack the user has space on. For the DEVICE option, a record is produced for each user on each specified pack; space on packs mounted but not specified is not recorded.

2. In addition to producing Statistical Collection File records reporting space utilization, the ACCOUNT directive also resets the space accounting fields in the System and Pack Master Catalogs to prevent their overflow.

3. Space in nonstructured files is not known and hence not accounted for.

Examples

1. ACCOUNT ONLINE

   Accounting records are produced for each user showing user space x time usage of filed devices and mounted removable structured disk packs.

2. ACCOUNT DEVICE/181PK,P3715/

   Accounting records are produced for each user showing user space x time usage of the named pack. The pack must be mounted.

3. ACCOUNT DEVICE/181PK,P3715,190PK,673/

   Accounting records are produced for each user showing user space x time usage of the named packs that must be mounted. One record is produced for each user with space on a named pack for each named pack.

Device Space Utilization Reporting Directive

Format

```
SPUTIL
```

Required

No requirements

Options

None

Notes

1. The SPUTIL directive produces a report giving information about space on fixed devices and mounted removable structured disk packs.

2. The report shows what space is available, for what purposes unavailable space is used, and indicates device activity and failure rates.

3. Appendix A contains a description of the report produced by this directive.

Example

```
SPUTIL
```

A report is printed showing space available on or assigned from each fixed device and mounted removable structured disk pack. The same report is also produced by a MASLST directive without a user being named.

Mark Time Sharing System Usage of Files and Catalogs At End

Format

UNBUSY

Required

No requirements

Options

None

Notes

1. This directive is only required when

   a. Time Sharing System (TSS) aborts,

   b. wrap-up for TSS fails, and

   c. neither TSS nor system is restarted.

2. Files being used by TSS at time of abort are left marked busy, which prevents subsequent allocation of the files to other jobs. To remove the busy mark, TSS or the system can be restarted or this directive may be used.

3. SMC entries being modified by TSS at time of its abort are left marked locked, which prevents subsequent modification of the SMC entry. (SMC modification occurs on log-off of a TSS user, as well as on explicit MODMAS.) To remove the SMC entry lock, TSS or the system can be restarted or this directive may be used.

Pack Create

Format

```
                                  ┌ BLOCKS ┐
CRPACK user name, DEVICE/.../,  < LLINKS  >  /.../
                                  │ LINKS  │
                                  └ SIZE   ┘
```

Required

1. Name of user for whom authorization has been granted to place catalogs and files on the specified removable structured disk pack.

2. Specification of the removable structured disk pack:

```
          ┌ 167PK ┐
          │ 170PK │
          │ 180PK │
          │ 181PK │
DEVICE/  <  190PK  >  ,pack serial number/
          │ 191PK │
          │ 310PK │
          │ 400PK │
          │ 450PK │
          └ 500PK ┘
```

3. Specification of the limits of space on the disk pack that files and catalogs cataloged under the user name can occupy:

```
┌ BLOCKS ┐
│ LLINKS │    ┌ UNLIMITED      ┐
│ LINKS  │  / └ maximum number ┘  /
└ SIZE   ┘
```

Number is one through six digits.

(BLOCKS = LLINKS = 320 words, LINKS = SIZE = 3840 words.) Any request for a maximum file size greater than 262,143 llinks is treated as UNLIMITED.

Options

None

Notes

1. Pack serial number can be one through five characters (letters or digits) required to identify pack for mounting purposes. The first two characters must not be zeros.

2. Space limit should be specified with knowledge of available capacity on each type of pack as well as authorizations granted for and space currently occupied by other users on the pack.

Pack Type	Capacity
167PK	7,992
170PK	14,400
180PK	14,400
181PK	14,400
190PK	47,944
191PK	61,864
310PK	14,400
400PK	61,864
450PK	123,272
500PK	123,272

Numbers are llinks and are approximate to allow variable amount of space for control information (label, defective space directory, inventory of available space).

3. Specified pack must be mounted. If a removable structured disk pack is required for a $ FILSYS activity, then a $ xxxPK control card is required in the deck setup after the $ PRIVITY control card.

4. No Master Create need have been performed for the user name prior to the Pack Create for that user.

Examples

1.
```
CRPACK   CORPORATE,DEVICE/181PK,P3715/,LLINKS/500/
```

A Pack Master Catalog entry is to be created on a 181 removable structured disk pack, with serial number P3715. The entry is for the user named CORPORATE, and an allowance of 500 llinks is specified.

2.
```
CRPACK   J.P.JONES,DEVICE/181PK,P3715/,SIZE/200/
```

A Pack Master Catalog entry is to be created for J.P.JONES on a 181 pack, with serial number P3715. The space allowed this user on this pack is 200 links.

Pack Modify

Format

```
                            ┌ BLOCKS ┐
                            │ LLINKS │
MODPAC user name,DEVICE/.../,┤ LINKS  ├ /.../
                            └ SIZE   ┘
```

Required

1. Name of user for whom authorization has been granted to place catalogs and files on specified removable structured disk pack.

2. Specification of the removable structured disk pack:

```
         ┌ 167PK ┐
         │ 170PK │
         │ 180PK │
         │ 181PK │
DEVICE/ ─┤ 190PK ├─ ,pack serial number/
         │ 191PK │
         │ 310PK │
         │ 400PK │
         │ 450PK │
         └ 500PK ┘
```

3. Change in specification of the limits of space on the disk pack that files and catalogs cataloged under the user name can occupy:

```
┌ BLOCKS ┐
│ LLINKS │     ┌ UNLIMITED      ┐
│ LINKS  ├ /  ─┤                ├─  /
└ SIZE   ┘     └ maximum number ┘
```

Number is one through six digits.

Number specified cannot be smaller than amount of space currently occupied by catalogs and files for user name on disk pack.

(BLOCKS = LLINKS = 320 words, LINKS = SIZE = 3840 words) Any request for a maximum file size greater than 262,143 llinks is treated as UNLIMITED.

Options

None

Note

See "Notes" of Pack Create directive.

Examples

1. `MODPAC  CORPORATE,DEVICE/181PK,P3715/,SIZE/UNLIMITED/`

   The Pack Master Catalog (PMC) entry for the user named CORPORATE on the removable structured disk pack, whose type and serial number is given, is modified to allow the user unlimited space on the pack.

2. `MODPAC  J.P.JONES,DEVICE/181PK,P3715/,LLINKS/1732/`

   The PMC entry for J.P.JONES on the pack is modified to decrease the space allowed that user on that pack.

Pack Delete

Format

```
{ DELPAC }
{ RELPAC }  user name, DEVICE/.../
```

Required

1. Name of user to be deleted on specified removable structured disk pack.

2. Specification of removable structured disk pack:

```
         { 167PK }
         { 170PK }
         { 180PK }
         { 181PK }
DEVICE/  { 190PK } ,pack serial number/
         { 191PK }
         { 310PK }
         { 400PK }
         { 450PK }
         { 500PK }
```

Options

None

Notes

1. A Pack Delete is similar to a Catalog Delete where the catalog is the highest level catalog for the user on the pack except that, in addition, the Pack Master Catalog for the user is also deleted. See the notes for Catalog Delete directive.

2. The DELPAC directive causes subordinate files to be overwritten with zeros, as CPURGE does for Catalog Delete. The RELPAC directive, as does the CRELES, causes space to be returned without being overwritten.

3. If the Pack and Master Delete directives are to be processed for the same user-name, then the Pack Delete directive must precede the Master Delete directive; otherwise space on the removable structured disk pack is not returned. If a removable structured disk pack is required for a $ FILSYS activity, then a $ xxxPK control card is required in the deck setup after the $ PRIVITY control card. Since the $ 500PK control card does not exist, the user must ascertain that this logical pack is on line before executing the job.

4. The user name must be specified as an entry in the System Master Catalog prior to the use of these directives.

5. A delete of a Pack Master Catalog (via a RELPAC or DELPAC statement to a FILSYS activity), causes the User Master Catalog on a fixed device to be deleted, unless the User Master Catalog has a subordinate that is not on the pack.

Examples

1. RELPAC J.P.JONES,DEVICE/181PK,P3715/

   The Pack Master Catalog for the user J.P.JONES and all catalogs and file descriptions for that user on the named pack are deleted. Spaces assigned to files on the pack cataloged under this user name are released for subsequent assignment. Note that files and catalogs for J.P.JONES on fixed devices or other packs are not affected, nor are files and catalogs for other users on this pack.

2. DELPAC CORPORATE,DEVICE/181PK,P3715/

   The Pack Master catalog for the user named CORPORATE and all catalogs and file descriptions for that user on this pack are deleted. Spaces assigned to files for the user on the pack are zeroed and then released for subsequent assignment.

File Recovery

Format

RECOVER option(s)

Required

No requirements

Options

1. ALL or null

   To specify that all files not yet recovered are to be processed in the recovery attempt.

2. SNUMB/#/

   To specify that all files not yet recovered for a given job are to be processed in the recovery attempt.

Notes

1. Recovery processing only occurs for the first system failure requiring recovery. Other failures are not processed until recovery of the first failure is completed.

2. Several attempts may be required to complete recovery for a failure; however, once recovery is completed, recovery of the next failure is attempted.

Examples

1. RECOVER

   An attempt is made to recover all files for first system failure.

2. RECOVER SNUMB/1776Z/

   An attempt is made to recover all files updated by job number 1776Z for the first system failure.

Recovery Tables Dump

Format

RECDUMP option,additional option(s)

Required

1. ALL

   To specify that recovery information for all files must be dumped. Additional options may be used to qualify the type of information to be recovered.

2. SNUMB/#/

   To specify that recovery information for a given job must be dumped. Additional options may be used to qualify the type of information to be recovered.

Options

1. $\text{DATA/}\left\{\begin{matrix}\text{HEADERS}\\ \text{ALL}\end{matrix}\right\}\text{/}$

   Specification of the portion of the collection file to be dumped. HEADERS specifies the collection file headers and all valid page headers. ALL specifies the entire collection file, that is, the collection file headers, page headers, and page images.

2. TABLES

   Specification of the recovered protection tables.

Note

The dumped collection file and recovery tables are those remaining after the first system failure.

Examples

1. RECDUMP SNUMB/3669T/

   Recovery information is provided for job number 3669T.

2. RECDUMP ALL,DATA/HEADERS/,TABLES

   Recovery information is provided with the collection file headers, page headers, and recovery tables for all jobs remaining after the first system failure.

Remove Recovery Tables

Format

REMOVE option(s)

Required

1. ALL

   To specify the removal of all recovery tables and collection files.

2. SNUMB/#/

   To specify the removal of the recovery tables and collection files of a given job.

Options

None

Note

The REMOVE directive is used to remove only those tables and files belonging to the first system failure.

Examples

1. REMOVE ALL

   All recovery tables and collection files for the first system failure are removed.

2. REMOVE SNUMB/6680Z/

   Recovery tables and collection files for job 6680Z from the first system failure are removed.

Patch Device

Format

PATCH DEVICE/option, additional options/

Required

1. A three-character BCD device name or a structured disk pack serial number. Note that the numbered disk pack may be on a spindle identified either as permanent or removable.

2. Absolute sector address in octal.

Options

1. Checksum indicator for last word of sector:

   a. NOCKSUM is used to ignore the checksum of reads and writes.

   b. CKSUM is used to validate the checksum after read and to recalculate a new checksum on write.

   c. NEWCKSUM is used to ignore the checksum on read and to recalculate a new checksum on write.

   d. READ or R is used to validate the checksum on read and to dump the sector. (If READ or R is used no more options are required. Otherwise, the following options are required. If NEWCKSUM is used, other options may be required.)

2. Word position in the sector to be changed.

   The number specified must be octal (first word is 0 and the last word is 77)

3. New contents of the word being changed.

   Contiguous words being changed may be listed in order separated by commas. If the contents of a disk are to be validated before making the change, then the old contents must precede the new contents and both are separated by a colon. In this instance, the word or words on disk are changed only if they agree with the specified pattern.

   The following message appears if the data pattern in this directive (x...x) does not correspond to the content of the disk location being changed.

   VALIDATION ERROR x...x SHOULD BE y...y

   No changes are made to the disk via the directive once this message appears.

Note

The sector contents are dumped before and after changes.

Examples

1. PATCH DEVICE/DS4,3257,NEWCKSUM/

   The absolute sector number 3257 octal is read from device DS4, words 0 through 76 are checksummed and stored in word 77, and the sector is rewritten.

2. PATCH DEVICE/C159,137542,CKSUM,25,712551466462, 252020202020,203726023465/

   If the checksum in word 77 of sector 137542 on disk pack serial number C159 is correct, then the contents of words 25, 26 and 27 are changed to the indicated values regardless of the current contents of the words. A new checksum is calculated before the sector is rewritten.

3. PATCH DEVICE/ST1,730,CKSUM,1,735001017:1017000735, 725001007:1007000725/

   If the checksum in word 77 of sector 730 on device ST1 is correct, then the contents of word 1 is changed to 001017000735 if it contains 000735001017 and the contents of word 2 is changed to 001007000752 if it contains 000725000107. A new checksum is calculated before the sector is rewritten.

4. PATCH DEVICE/DS3,63215,READ/

   Sector 63215 from device ST1 is dumped.

SECTION XI

SAVE AND RESTORE DIRECTIVES

This section describes master and user save and restore directives.

MASTER SAVE AND RESTORE DIRECTIVES

Master Save

Format

```
SAVEMAST   option(s)
```

Required

None

Options

1. LISTOPT/ $\left\{\begin{matrix}\text{YES}\\ \text{NO}\end{matrix}\right\}$ /

   YES is assumed. If YES is assumed or specified, names of all files saved are listed. If NO is specified, only names of saved files that are write busy, abort locked, with defective spaces, or wholly or partially unreadable are listed.

2. INCR

   Only files are saved which have:

   a. INCRSAVE/YES/ specified or assumed for them.

   b. Been written to or restored since their last save.

   c. Not been eliminated because of DONTDO or DOONLY and/or RANGE, SINCE, and DEVICE options on the SAVEMAST.

   d. Not remained write busy during the save unless journaling is also specified.

3. DONTDO/user name,.../

   Files for named users are not saved.

4. DOONLY/user name,.../

   Only files for named users are saved. DONTDO and DOONLY options must not be specified together in one use of the directive.

5. RANGE/32nd,32nd/

   Only files are saved for users identified on the master catalog listing as belonging to 32nds (into which user names are hashed) in the inclusive range specified. The 32nds are numbered 0,1....,31. For example, RANGE/0,5/ specifies that files for users hashing to any of the first six 32nds, those numbered 0,1,2,3,4,or 5, are to be saved. More than one RANGE/.../ option may be given to specify several ranges.

The RANGE option can be used to divide the work of saving all files among several concurrently executing save jobs. The DONTDO option may be used in conjunction with the RANGE to give a finer division of work, or to avoid the saving of files for some users in the range specified.

6. $\text{SINCE/}\left\{\begin{matrix}\text{TODAY}\\ \text{date}\end{matrix}\right\}\ [\text{,time}]\ /$

Date is in format mmddyy and time is in hhmm, hh:mm, or hh.hhh. Only files are saved that have been written to since the date (and time) specified. Files that remain write busy during the save are not saved, however, unless journaling is specified for the file.

The difference between a SINCE and INCR save is that the SINCE date and time can be that of the last total save so that incremental saves are accumulated into one SINCE save. Furthermore, only files for which INCRSAVE / YES / is specified or assumed are saved using the INCR option, while the SINCE option causes files to be saved regardless of the INCRSAVE specification. Use of both INCR and SINCE is possible but not often of value.

7. DEVICE/ [ONLY,] device name,.../

Files are saved that are assigned to the named devices. Files assigned partially to devices named and partially to devices not named are saved entirely unless the optional word ONLY is present, or the file specifies duplication or journaling. In such cases, only the portion of a file on a named device is saved. Catalogs are saved only if one of the devices named is ST1 (regardless of whether ST1 actually contains catalogs or not).

8. DEVICE/pack type,pack serial number/

Only catalogs and files are saved that are assigned to the specified pack which must be mounted.

9. RESET/RFILES/

This option saves specified files and resets the file restore indicator to the off position. Use of this option by itself causes redundancy unless a since option is used.

Notes

1. Files that meet the selection criteria of DONTDO or DOONLY and INCR, RANGE, SINCE, and DEVICE (whichever of these options are specified) are the ones that are saved. Ordinarily, it will not be necessary to use more than one option to select the files to be saved, but all can be used subject only to the constraint that not both DONTDO and DOONLY are specified. Often, of course, none of these options will be necessary, as when all files are to be saved in a single sequence of save volumes.

2. If a file with code XS is present, it is written to at the same time the primary save file, file code PS, is written to. File PS can be assigned one unit or two to allow unit switching, depending on number of volumes expected. If file XS is present, it should be assigned the same number of units as file PS.

3. All content of space currently assigned to each file is saved except that if the file description indicates that the file has never been written to, in which case no content is saved. In no case is a search made for an end-of-file mark to detect the last space written to. If the DEVICE option is used to constrain the files saved, all of the file is saved even though only a part is on the specified device(s). An exception is made when the file is protected by FMS journaling, or if duplication or the ONLY option is used in which case only the part on the specified device(s) is saved.

4. Files are saved whatever their condition - abort or security locked, write busy, or with one or more defective or unreadable spaces. If a file is write busy, an attempt is made to save the file later when it is no longer write busy, but when the attempt fails, the file is saved just as it is encountered. An exception is the case where the INCR or SINCE option is used and FMS journaling is not specified for the file; the file is not saved if it remains write busy. When the file is saved while write busy, a warning is provided, in case it is considered desirable to specially save such files later, and the save is so marked.

5. If a file is found partially or wholly unreadable when it is being saved, it is saved just as it is read, but both the file description and the save record are marked to show which parts of the file are unreadable, and a warning is provided in case recovery is to be initiated. If the file is duplicated, parts found unreadable from the original are read from the duplicate.

6. If the file is already marked as having defective space(s) or is abort locked, again the file is saved just as it is, but the save record is marked, and a warning is provided.

7. Files on removable structured disk packs are not saved unless the DEVICE/pack/ option is used and the specified pack is mounted.

8. A file on magnetic tape or nonstructured disk pack is not saved whether mounted or not.

9. If a file is protected by FMS journaling of file changes and the DEVICE option is specified on the save, only the parts of the file contained on the specified device(s) are saved. For it is assumed that the journal will be used to update the parts restored from the save so they will be compatible with parts not saved.

   Similarly, if journaling is specified for a file, the file is saved if it remains write busy even when the save is under INCR or SINCE options. For rollforward use, the journal is required to update the version restored from the most recent save whether or not the save was done while the file is write busy.

10. A file is regarded as having been written to, for the purposes of INCR or SINCE options, if the allocation has been marked by IOS to show that at least one write has been issued during the allocation, or if on system restart the file description shows a write allocation to the file to have been interrupted by the system failure. A test mode allocation is not interpreted as a write allocation.

11. The RESET/RFILES/ option may be used in conjunction with since and incremental saves to save a file, if the file restore indicator is set on, and to turn off the restore indicator. Any save that is not a SINCE or INCR, unconditionally turns off the file restore indicator. During a SINCE or INCR save, the RESET/RFILES/ option can be used to turn off the file restore indicator and prevent redundant file saving in succeeding save operations. This may be accomplished when the succeeding SINCE saves have a different date and time parameter. Then, only files changed since the last time specified and files restored since the last save are saved. If the date and time parameters are not changed in succeeding saves, the option should not be used, because files saved on previous saves will be saved again.

    When SINCE saves are used, if a total save is not taken following a total restore, the first SINCE save will save all the files. Furthermore, the total save that was used for the total restore must be maintained until another total save is taken.

12. Multiple SAVEMAST directives must not be used within the same activity. Only the last SAVEMAST directive in the input stream is processed.

Examples

1. `SAVEMAST`

   Files and catalogs on fixed devices for all users are saved. Only exceptions are reported.

2. `SAVEMAST DOONLY/CORPORATE/`

   Files and catalogs on fixed devices for the named user are saved.

3. `SAVEMAST DONTDO/SALTRIVER/`

   Files and catalogs on fixed devices for all users except the user named are saved.

4. `SAVEMAST DEVICE/181PK,P3715/`

   Files and catalogs for all users on the named pack are saved.

5. `SAVEMAST DONTDO/SALTRIVER/,DEVICE/DS1,DS2,DS3/`

   Files on the named devices for all users except the user named are saved.

6. `SAVEMAST RANGE/14,21/`

   Files and catalogs on fixed devices for any users whose names hash to the 14th, 15th, 16th, 17th, 18th, 19th, 20th, or 21st section are saved.

7. `SAVEMAST RANGE/22,26/,DONTDO/SALTRIVER/`

   Files and catalogs on fixed devices for all users whose names hash to the 22nd through 26th sections, except the user named, are saved.

8. `SAVEMAST RANGE/0,15/,DEVICE/181PK,P3715/`

   Files and catalogs on the named pack for all users whose names hash to the first 16 (0 through 15) sections are saved.

9.
   ```
   SAVEMAST DOONLY/CHICAGO,CORPORATE,DOMESDAY,
   FIEF,INSTALL,J.P.JONES,OPERA-1,PRALINE/,LISTOPT/YES/
   ```

   Files and catalogs on fixed devices for the named users are saved, and a list is produced showing each file saved.

10. `SAVEMAST SINCE/TODAY/`

    Catalogs on fixed devices and all files on fixed devices that have been written to today are saved.

11.

```
SAVEMAST  SINCE/070673,17.500/,DONTDO/SALTRIVER/
```

Catalogs on fixed devices and all files on fixed devices that have been written to since 5:30 p.m., 6 July 1973 are saved except those for the named user.

12.

```
SAVEMAST  INCR,LISTOPT/YES/
```

Catalogs on fixed devices and all files on fixed devices that have both been written to since the last save and have specified or assumed for them INCRSAVE/YES/ are saved. A listing of each such file is also produced.

13.

```
SAVEMAST  DOONLY/CORPORATE,J.P.JONES/,
DEVICE/181PK,P3715/, SINCE/070673,17:30/
```

Catalogs on the named pack and files on that pack that have been written to since 5:30 p.m., 6 July 1973 for the users named are saved.

14.

```
SAVEMAST  LISTOPT/YES/
```

Files and catalogs on fixed devices for all users are saved and listed.

Master Restore

Format

RESTOREMAST option(s)

Required

None

Options

1. LISTOPT/$\left\{\begin{matrix}\text{YES}\\\text{NO}\end{matrix}\right\}$/

   YES is assumed. If YES is assumed or specified, names of all restored files are listed. If NO is specified, only names of restored files are listed that are restored from a save marked write busy while saved, abort locked, with defective space, or wholly or partially unreadable.

2. DONTDO/user name,.../

   Files for named users are not restored.

3. DOONLY/user name,.../

   Only files for the named user are restored. DONTDO and DOONLY options must not be specified together in one use of the directive.

4. SINCE/$\left\{\begin{matrix}\text{TODAY}\\\text{date}\end{matrix}\right\}$/

   Date is given in mmddyy format. Only files are restored that are marked on the save as having been allocated or modified since the date specified. Allocated means that a request to use the file was granted; modified means the file description was modified by means of a File Modify directive. Notice that allocation or modification after the save is not considered: only dates of last allocation and modification at the time of the save are considered.

5. DEVICE/device name,.../

   Only files are restored that existing catalogs show to be assigned wholly or partially to one or more of the named device(s).

6. DEVICE/pack type, pack serial number/

   Catalogs and files are restored to the specified pack, thus overriding the pack type and pack serial numbers from which the save was taken. Note that the save must have been taken from a removable structured pack.

7. RANGE/32nd,32nd/

Files are restored for users identified on the master catalog listing as belonging to 32nds (into which user names are hashed) in the inclusive range specified. For example, RANGE/0,5/ specifies that files for users hashing to any of the first six 32nds, those numbered 0,1,2,3,4, or 5, are to be restored.

8. RESET/ $\left\{\begin{array}{l}\text{DEVICE}\\ \text{device name,...}\\ \text{RFILES}\end{array}\right\}$ /

If a device name or device class was specified when files were created, Restore attempts to put the files back on the same devices or device classes again. However, it is possible that these devices or classes do not exist in the system to which the restore is being done and consequently such files are not restored. The RESET option overcomes this problem. The option forces the restore to remove all device constraints from catalogs and selects available devices in the system to which files are restored. DEVICE removes all device constraints or constraints are removed from only these devices named.

The RFILES option sets the file restore indicator in the off position. In the normal condition the file restore indicator is in the on position.

9. RENAME/device name,...=device name:device name,...=device name:.../

Files constrained to device names will be constrained to equated device names after the restore. The colon separates groups of device names. A maximum of 200 device names can be specified, including the colons (i.e., colons are included in the device name count).

Notes

1. In addition to the selection criteria expressed in RESTOREMAST options, it is obvious that only files that are contained on the save(s) being presented can be restored. Of the files so contained, only those meeting criteria expressed by the options are restored. If the save was only of files on specified devices or in specified SMC sections, for example, and the Restore specifies DOONLY, then only files cataloged under the user names specified and present on the save will be restored. There may, of course, be no files both meeting the criteria and present on the save as when the save is of SMC sections not including DOONLY specified names. When inconsistencies between save contents and restore options are obvious as in the example above, the error is reported so that a correct save or correct restore specifications can be provided.

2. If restoring is being done for a user, then ordinarily all catalogs for the user (except those on structured removable disk packs) are also restored.

3. When the restore is done under the DEVICE option, however, no catalogs are restored from the save. Catalogs must exist on other devices, or if some of the devices to be restored contain catalogs, duplicates of these catalogs must exist on other devices. If such is the case, a copy of existing catalogs is made to the restored devices.

The save used for restoring may or may not have been done under a DEVICE option, but if it was, the device named on the restore obviously must have been included in those named on the save.

4. If the restore is from a save done under the DEVICE/pack/option, only catalogs on that pack are restored, and existing catalogs on fixed device(s) and other packs are not affected. If references to catalogs on the restored for pack do not exist on fixed device(s) or have been destroyed, they must be created before files on the pack can be used. The pack from which the save was done must be mounted at the time of the restore.

5. Restoring from save(s) performed under the INCR option requires that the installation present first the most recently performed incremental save, then less recently performed saves in reverse chronological order, and finally a total (not incremental) save. If catalogs are to be restored from the saves, the version of catalogs contained on the first save presented is the one that is restored. File content is restored from this sequence of saves only if the catalog shows the file existed at the time of the most recent save and if the file content has not already been restored from a previously presented member of the sequence of saves.

   When the DEVICE option is used, the existing or duplicate catalogs control restoring of file content.

   In any case, file content is restored only for files existing either currently or at time of most recent save and only the most recent version of the file is restored.

6. Restoring from save(s) performed under the SINCE option are similarly restored. The installation should present first the most recent (or only) SINCE save, followed by those less recently performed (if any), and finally a total save.

7. If the DEVICE option is used and there currently is an existing file description that shows the file to be duplicated, the file content is restored from the duplicate and the one on the save is not used unless the duplicate file content is null or unavailable. A file description currently exists, however, only if catalogs are duplicated or a DEVICE restore is performed for device(s) that did not contain catalogs.

8. When file content is restored from a save and the file description that is used specifies the file is to be duplicated, space is obtained for two copies of the files and the file content is written to both spaces.

9. Device specifications in whatever file description is used - currently existing or restored from save - control where the file content for that file is restored. If the specifications cannot be met, the file content is not restored and a report is issued.

   Files restored from a device save are assigned space only on devices saved. That is, if the save included the option DEVICE/DS1,DS2,DS3/, for example, a restore from this save assigns space only on DS1, DS2, and DS3.

When file content is not restored, the file description shows no space assigned to the file, which prevents any type of allocation to the file. Users can determine, by listing the file description, the cause of the no restore, can delete and create again the file description with a new device specification, and then can restore the file.

10. If a file is abort locked when it is saved, the save is so marked. When file content for such a file is restored the file description is marked to show the file is abort locked. If the file description shows the file is abort locked but the save from which the file content is done does not, the abort lock is removed. Similarly, a file with one or more parts marked defective when saved causes its file description to be marked accordingly, and a file description is so marked only when the save from which the restore is done is also done.

    A report is made of such exceptions in case earlier saves are to be used or recovery action is to be initiated or discontinued.

11. When a file is restored using the DEVICE option, all of the file is restored including parts on devices not specified, if any. An exception is made when FMS journaling was specified for the file at the time of the save. Then only the part of the file on specified device(s) is restored since rollforward using journals can be relied on to update the part restored. If a duplicate of the file is available, only the part of it duplicating the part on the specified device(s) is restored.

12. When rollforward by means of journals can be done to bring the file up to date, a report of any such files restored is made. The report shows whether the rollforward is optional or required because only part of the file was restored or the file was restored from a save made while the file was write busy. If rollforward is required, the file description is marked abort locked if any ABORT option is specified.

    If a file description for a restored file currently exists (as when catalogs are duplicated or a device restore is performed for device(s) without catalogs), the date and time of last change to the file as recorded in the currently existing file description is reported for use in controlling rollforward. If the file description as well as file is restored from the save, however, the date and time of last change to the file is assumed to be the current time, to be safe, and the current time is so reported.

13. Notice that saves done under INCR or SINCE options are not made while the file is write busy unless journaling is specified for the file. Files restored from saves not done under these options, however, can be made while the files are write busy even though no journaling is specified for them. A report of such restores is made in case restore from an earlier save is to be performed or recovery action initiated, and the file description is marked abort locked if any ABORT option is specified.

14. A RESTOREMAST uses only saves performed by means of SAVEMAST. If more recent saves have been performed by means of a SAVE directive or another save program, it is the responsibility of the user to note when restores from these are required and to initiate them. The date and time of last write to the file at the time of the save from which the file has been restored is set as date and time of last change in the file description to indicate what version is restored.

15. The RESET/RFILES/ option should be used when performing a total system restore. Use of this option causes succeeding, SINCE or INCR saves not to save the restored files unless they are updated.

16. Restoring from a save of a removable disk pack to a permanent disk pack requires that the structures to be restored be released and the SMC on the permanent disk pack be recreated, followed by a user restore.

Examples

1. RESTOREMAST

   Files and catalogs on fixed devices for all users are restored. Only exceptions are reported.

2. RESTOREMAST DOONLY/CORPORATE/

   Files and catalogs on fixed devices for the named user are restored.

3. RESTOREMAST DONTDO/SWEETWATER/

   Files and catalogs on fixed devices for all users except the user named are restored.

4. RESTOREMAST LISTOPT/ALL/

   Files and catalogs for all users are restored and a list of each file restored is produced.

5. RESTOREMAST DEVICE/ST3/

   Files shown in existing catalogs for all users to be assigned to the named device are restored. If the device has catalogs, duplicates of these catalogs must exist on another device and these are copied to the device.

6. RESTOREMAST DEVICE/DP1/,DONTDO/SWEETWATER/

   Files shown in the existing catalogs for all except the named user to be assigned to the named device are restored. If the device has catalogs, duplicates of all these catalogs except the named user must exist and are copied to the device.

7. RESTOREMAST SINCE/070673/

   Files and catalogs that describe these files on fixed devices for all users are restored only if the saved file description shows the file has been allocated or the file description modified since 6 July 1973.

8. RESTOREMAST RANGE/0,15/

   Files and catalogs for users whose names hash to the first sixteen sections are restored. Ordinarily, the save from which restoring is done specifies the range of sections, but sometimes it may be desirable to restore only a subset of the ranges on a save.

Replace Defective Pages of Files

Format

```
REPLACE file name,...
```

Required

At least one qualified file name. A $ USERID control card or USERID directive must be provided with the name of the creator of the file or one with RECOVERY permission. If a $ PRIVITY control card is present, however, no $ USERID control card or USERID directive need be given.

Options

Additional qualified file names. Unless a $ PRIVITY control card is present, a user name provided must be that of the creator of, or a user with RECOVERY permission for, all named files.

Notes

1. The file description for each named file is consulted for numbers of pages marked defective in each file. The version of each such page is read from the presented save and written to the file. If the file is duplicated and a good copy is available, this copy is read and written to the file.

2. If necessary, the space the page is assigned to is replaced by a newly assigned space to ensure that the page written to the file can subsequently be read.

3. The pages continue to be marked defective in the description, for it is expected that the replacement will be followed by a rollforward from a journal to obtain more recent versions of pages that have been changed since the save. However, if the date and time of the save from which replacement is made is not earlier than the date and time of last change to the file as recorded in its file description, then the pages will not be marked defective.

4. A report is made showing pages replaced and whether rollforward is required.

Examples

1. REPLACE CORPORATE$SEPT25/CUSTOMER$NOW

   The file named CUSTOMER cataloged under the user name CORPORATE is to have defective parts restored from the save.

2. REPLACE CORPORATE$SEPT25/CUSTOMER$NOW,
   CORPORATE$SEPT25/EQUIP,INSTALL/SYSA/DETAIL

   Files named CUSTOMER and EQUIP cataloged under user name CORPORATE and file named DETAIL, under user name INSTALL, are to have defective parts restored from the save.

Remake Changes to Files Using FMS Journal

Format

```
ROLLFWD   option(s)
```

Required

None. In this case, all file changes encountered on presented journal volume(s) are made to their respective files.

Options

1. DOONLY/qualified file name,qualified file name,.../

   Only changes for named files are made.

2. DONTDO/qualified file name,qualified file name,.../

   Changes for all files except the files named are made. DOONLY and DONTDO options should not be used together with one use of the directive.

3. DEVICE/device name,.../

   Only changes for parts of files assigned to named devices at time of journaling are made.

4. DEVICE/pack type,pack serial number/

   Only changes to parts of files assigned to specified pack at time of journaling are made. The specified pack must be mounted.

5. DEFECTIVE

   If file names are given, this option can be specified to indicate that only changes made to pages marked defective in the file description for each named file are to be re-made. This option is used after the REPLACE directive has been used to restore pages marked defective from a saved version of the files. By using the REPLACE directive first, pages not changed since the save will have been replaced from the save, and ones that have changed since the save will be replaced from the journal.

6.

```
       ( TODAY )
THRU/ {  date   } [, time]  /
       (       )
```

Date is given in mmddyy format and time is in hhmm, hh:mm, or hh.hhh format. Only changes shown on the journal to have been made on or before the specified data card time, are made to their respective files. The effect of this specification is to restore files to their version current at the specified time.

7.

```
       ( TODAY )
FROM/ {  date   } [,time]  /
       (       )
```

Date is given in mmddyy format and time is hhmm, hh:mm, or hh.hhh format. No changes shown on the journal to have been made on or before the specified time are made to the respective files.

8.

```
          ( YES  )
LISTOPT/ {  ONLY  } /
          ( NO   )
```

YES is assumed. If YES is assumed or specified, all activities appearing on the journal are listed. If ONLY is specified, only activities causing the file to be rolled forward are listed. If NO is specified, no activities are listed. Diagnostic messages are always listed regardless of the option selected.

Notes

1. All files for which rollforward is to be done must be currently available. Restoring of the files from their latest save should be performed first, followed by rollforward using each journal volume produced after or at the same time as the save. The earliest journal volume required is reported by the FMS Restore Program, and each volume includes the serial number of the next journal volume.

2. The last journal volume which should be provided can be determined by the THRU date, if it is specified, and by dates and times reported by the FMS Restore Program as last changes to files requiring rollforward, or by the date and time when restore was completed.

3. If FMS protection against incomplete update is specified for a file, the Restore Program abort locks the file. The Rollforward Program unlocks the file when it has again made changes to the file that were journaled since the save of the file. The date and time of last change to the file is obtained from the journal and recorded in the file description to show which version of the file has been produced by means of rollforward.

4. If some changes to the file cannot be read from the journal, the parts are marked defective. If later changes to these parts are found, the defective marks are erased. But if no later changes are found for some that were unreadable, rollforward leaves the file with these parts marked defective. If the pages were left marked defective by the Replace Program, when rollforward under the DEFECTIVE option is used, though, the defective marks are erased for pages unless the Rollforward Program found the last changes to these pages unreadable or found no changes.

Examples

1.
   ```
   ROLLFWD
   ```

   All file changes encountered on the journal volumes are made again to their respective files. This would be used after a full restore to bring files up-to-date.

2.
   ```
   ROLLFWD  THRU/TODAY,0800/
   ```

   All file changes encountered on the journal volumes made on or before 8 a.m. today are made again to their respective files. This would be used after a full restore to bring files to their versions current at eight o'clock.

3.
   ```
   ROLLFWD  DEVICE/DS3/
   ```

   All file changes encountered on the journal volumes for parts of files assigned to the device name DS3 are made again to their respective files. This would be used after a restore of files assigned to this device to bring the parts restored up-to-date.

4.
   ```
   ROLLFWD  DOONLY/CORPORATE/CUSTOMER,CORPORATE/EQUIP,
   INSTALL/SYA/DETAIL/,DEFECTIVE
   ```

   Files named CUSTOMER and EQUIP, subordinate to the user named CORPORATE, and DETAIL, subordinate to the user named INSTALL, are to have parts marked defective in their file descriptions replaced by the most recent change on the journal. This would be used after a REPLACE directive naming the same files to obtain the most recent version of each defective part in these files.

5.
   ```
   ROLLFWD  DONTDO/CLASS21/PROB1INPUT/
   ```

   All file changes encountered on the journal volumes except any for the named file PROB1INPUT subordinate to the user CLASS21 are made again to their respective files. This would be used to avoid bringing up-to-date a file for which the version from the save is desired.

6.
   ```
   ROLLFWD  DOONLY/SWEETWATER/POWER/ACCOUNT,SWEETWATER/
   POWER/IRRIGATION,SWEETWATER/POWER/WELLS/,DEVICE/DS9/
   ```

   Changes for parts of the named files on the named device are made again. This would be used after a restore of the parts of the files on this device to bring these parts of the files up-to-date with parts on other devices.

USER SAVE AND RESTORE DIRECTIVES

File and Catalog Save

Format

SAVE catalog or file name,option(s)

Required

Qualified catalog or file name. If a catalog is named, files subordinate to it are considered for saving. A $ USERID or USERID directive must be supplied, giving the name and log-on password of the creator of the catalog or file. If a $ PRIVITY control card is present, however, no user name need be given.

Options

1. $$\text{LISTOPT/}\begin{Bmatrix}\text{YES}\\\text{NO}\end{Bmatrix}\text{/}$$

   YES is assumed. If YES is assumed or specified, names of all files saved are listed. If NO is specified, only names of saved files that are write busy, abort locked, with defective space, or wholly or partially unreadable are listed.

2. INCR

   Only files for this named user are saved that have:

   a. INCRSAVE/YES/ specified or assumed for them.

   b. Been written to since their last save.

   c. Not been eliminated because of DOONLY, DONTDO, SINCE, or DEVICE option.

   d. Not remained write busy during the save unless journaling is specified for the file.

3. $$\text{SINCE/}\begin{Bmatrix}\text{TODAY}\\\text{date}\end{Bmatrix}\text{[, time] /}$$

   Date is in format mmddyy, and time is in hhmm, hh:mm, or hh.hhh. Only files for the named user are saved that have been written to since the date (and time) specified. Files that remain write-busy during the save are not saved when the SINCE option is used unless journaling is specified for the file.

   The difference between a SINCE and INCR save is as follows: The SINCE date and time can be that of the last total save so that incremental saves are accumulated into one SINCE save. Furthermore, only files for which INCRSAVE/YES/ is specified or assumed are saved using the INCR option while the SINCE option causes files to be saved regardless of the INCRSAVE specification. Both SINCE and INCR options can be used, but this is not often of value.

4. DEVICE/ [ONLY,] device name,.../

Files are saved that are assigned to the named devices. Files assigned partially to devices named and partially to devices not named are saved entirely unless the optional word ONLY is present, or the file specifies duplication or journaling. In such cases, only the portion of a file on a named device is saved. Catalogs are saved only if one of the devices named is ST1 (regardless of whether ST1 actually contains catalogs or not).

5. DEVICE/pack type,pack serial number/

Only files are saved that are assigned to the specified pack which must be mounted.

6. DONTDO/catalog or file name,.../

Names of catalogs or files subordinate to the catalog named after the directive can be given. If a catalog name is given, files subordinate to it are not saved.

7. DOONLY/catalog or file name,.../

Names of catalogs or files subordinate to the catalog named after the directive can be given. If a catalog name is given, files subordinate to it are saved. Both DONTDO and DOONLY options cannot appear in one use of a directive, nor can either appear when the name given after the directive is that of a file.

*

8. NEWNAM/user name/

Catalog structures and files are moved from one user name to another user name. The user name of the qualified file name is changed to the new named user for all catalogs and files that are saved. No catalog names, file names, and passwords are changed. When the NEWNAM option is used this directive requires PRIVITY.

Notes

1. If a file with code XS is present, it is written to at the same time the primary save file, file code PS, is written to. File PS can be assigned one unit or two to allow unit switching, depending on the number of volumes expected. If file XS is present, it should be assigned the same number of units as file PS.

2. All content of space currently assigned to each file is saved, except that if the file description indicates the file has never been written to, no contents are saved. In no case, however, is a search made for an end of file mark to detect the last space written to.

3. Files are saved whatever their condition - abort or security locked, write busy, or with one or more defective or unreadable spaces. If a file is write busy, however, an attempt is made to save the file later, but when this attempt also finds the file write busy, the file is saved as encountered. When the save is under the INCR or SINCE option and journaling is not specified for the file, the file is not saved if write busy. If the file is saved while write busy, a warning of the condition is provided, though, in case it is considered desirable to specially save or recover such files, and the save is so marked.

4. If the catalog named after the Save directive is on a removable structured disk pack, the catalog and its subordinate (subject to DOONLY or DONTDO restrictions) are saved if the pack is mounted. Similarly, if a file is named after the Save directive that is on a removable structured disk pack, that pack must be mounted. If on the other hand, the catalog named after the directive is not on a pack, no subordinates on a pack are saved, even though the pack is mounted. Naming such subordinates in DOONLY or DONTDO specifications is regarded as erroneous.

5. Files for the named user that are on nonstructured media - tapes or disk packs - are not saved whether the volume(s) are mounted or not.

6. A file is regarded as having been written to, for the purposes of INCR or SINCE options, if the allocation has been marked by Input-Output Service to show that at least one write was issued during the allocation, or if on system restart the file description shows a write allocation to the file to have been interrupted by the system failure. A test mode allocation is not interpreted as a write allocation.

7. Files subordinate to a catalog on a removable structured disk pack are saved when the name after the Save directive is of that catalog. Similarly, if the name of a file on a pack is given, that file is saved. In these cases, no DEVICE option may be supplied since a specific pack is implied by the use of the file or catalog name.

8. When a user has several level 1 catalogs on the same pack, however, the UMC can be named after the Save directive and the DEVICE/pack/ option given to specify that all files for the user on that pack are to be saved, subject to DOONLY or DONTDO constraints if these are given. This cannot be done when the UMC is itself on that pack, as note 7, makes clear.

9. Multiple SAVE directives must not be used within the same activity. Only the last SAVE directive in the input stream is processed.

10. User identification for the NEWNAM option must be the same as that of the user name of the qualified file name being saved.

11. Do not use more than one SAVE directive in a given activity.

Examples

1. `SAVE  CORPORATE$SEPT25`

   Catalogs and files on fixed devices for the user named CORPORATE are saved.

2. `SAVE  J.P.JONES/RMVBL,DONTDO/B/`

   The catalog RMVBL for the user named J.P.JONES and all files and catalogs subordinate to it are saved except the file whose full name is J.P.JONES/RMVBL/B. If B was a catalog, that catalog and any files and catalogs subordinate to it would not be saved. Because RMVBL is on a removable structured disk pack, all subordinates to it are also on that pack (see Example 3 under the Catalog Create directive).

3. `SAVE  SALTRIVER,DOONLY/ONE,THREE/`

   The catalog SALTRIVER and the two files ONE and THREE subordinate to it, but no others, are saved.

4. `SAVE  CLASS21,INCR,LISTOPT/YES/`

   All catalogs for the user named CLASS21 and files for this user for which INCRSAVE/YES/ has been assumed or specified and that have changed since the most recent save are saved. A list is provided of all files saved.

5. `SAVE  SWEETWATER,DEVICE/DS1,DS2,DS3/,DOONLY/POWER/`

   Any files subordinate to the catalog POWER with space assigned to any of the named devices are saved. If a file is partly on a named device, all of it is saved unless FMS journaling is specified for it.

6. `SAVE  CLASS21,SINCE/070673/`

   All catalogs for the named user and any files for this user that have been changed since the beginning of 6 July 1973 are saved.

7. SAVE ACE/POWER, NEWNAM/SWEETWATER/

The catalog POWER for the user named ACE and all files and catalogs subordinate to it are saved as if they were subordinate to the user named SWEETWATER.

File and Catalog Restore

Format

```
RESTORE  catalog or file name,option(s)
```

Required

Qualified catalog or file name. If a catalog name is given, all files subordinate to it are ones considered for restoring subject to restrictions expressed by options, if any. If a file name is given, only that file is restored. A $ USERID control card or USERID directive must be provided with the name of creator of the catalog or file name given and with that user's log-on password. If a $ PRIVITY control card is present, however, no user name or password need be given.

Options

1. LISTOPT/ $\left\{\begin{matrix}\text{YES}\\ \text{NO}\end{matrix}\right\}$ /

   YES is assumed. If YES is assumed or specified, names of all restored files are listed that are restored from a save marked write busy, abort locked, or with defective or unreadable content at the time of the save.

2. DONTDO/catalog or file name,.../

   If a catalog name was given after RESTORE, names of catalogs or files immediately subordinate to it may be given to indicate that files subordinate to the named catalogs and named files are not to be restored.

3. DOONLY/catalog or file name,.../

   If a catalog name was given after RESTORE, names of catalogs or files immediately subordinate to it may be given to indicate that only files subordinate to the named catalogs and named files are to be restored.

4. SINCE/ $\left\{\begin{matrix}\text{TODAY}\\ \text{date}\end{matrix}\right\}$ /

   Date is given in mmddyy format. Only files subordinate to the named catalog that are marked on the save as having been allocated or modified since the date specified are restored. Allocated means that a request to use the file was granted; modified means the file description was modified by means of a File Modify directive. Notice that allocation or modification after the save is not considered; only dates of last allocation and modification at the time of the save are considered.

5. DEVICE/device name,.../

Only files subordinate to the named catalog are restored that are shown in the existing catalog to have been on the named device(s). No catalogs are restored.

6. NEWNAM/user name/

Catalog structures and files are moved from one user name to another user name. The user name of the qualified file name is changed to the new named user for all catalogs and files that are changed. No catalog names, file names, or passwords are changed. When the NEWNAM option is used this directive requires PRIVITY.

7.
```
RESET/  ┌DEVICE          ┐/
        │device name,... │
        └RFILES          ┘
```

If a device name or device class was specified when files were created, Restore attempts to put the files back on the same devices or device classes again. However, it is possible that these devices or classes do not exist in the system to which the restore is being done and consequently such files are not restored. The RESET option overcomes this problem. The option forces the restore to remove all device constraints from catalogs and selects available devices in the system to which files are restored. DEVICE removes all device contraints or constraints are removed from only these devices named.

8. RENAME/device name,...=device name:device name,...=device name:.../

Files constrained to device names will be constrained to equated device names after the restore. The colon separates groups of device names. A maximum of 200 device names can be specified, including the colons (i.e., colons are included in the device name count). The RENAME option is only effective for files not existing prior to the RESTORE.

Notes

1. If a file is named after RESTORE, the DONTDO, DOONLY, SINCE, and DEVICE options are inapplicable, and an error is returned if they are specified. Both DONTDO and DOONLY options cannot be submitted in one use of the directive. Any other combination of options is possible; however, it is obvious that only files that are contained on the presented save(s) can be restored. If there should be no files both satisfying the selection criteria imposed by the restore options and contained on the presented save, an error is returned. An example is a save done under a device option that does not include any of the devices specified in the restore option.

2. For a user restore, currently existing catalogs, file descriptions, and file content space are used when available. Only when there is no currently existing catalog or file description for a file to be restored is the catalog or file description from the save used. When the currently existing are used, the following adjustments are made:

   a. The date and time of last write to a file recorded in the file description for the file is changed to the one in the saved file description.

   b. If the restored file content is larger than existing file content space, more space is obtained up to the maximum specified in the existing file description. *

   c. If a catalog contains no reference to a subordinate, a reference is entered.

3. When the currently existing file or catalog is not the one that is wanted, it must be deleted prior to the restore. This will cause the catalog and file description from the save to be used and new file content space to be assigned. If both the saved and existing versions are desired, the name of the existing may be modified prior to the restore to prevent it from being replaced by the saved version.

4. The fact that an existing version of file description will be used, however, enables the user to decrease the size of the file, change the device or device type the file is on, or cause the file to be duplicated. These changes to the version on the save can be effected by deleting existing files and then re-creating them.

   The decrease in size can be effected by recreating the file with maximum size less than the saved version. In this case, the file content is truncated on restore, but no mark is made in the file content. A report of the truncation is made in case a new end-of-file mark needs to be supplied or the restore should be done again after modifying the maximum size.

5. Device specifications in the currently existing file description or superordinate catalogs are used to control where the file is assigned. If there are no existing file description and catalogs, though, the specifications, if any, at the time of the save are applied. Thus to cancel specifications effective at the time of the save - either in superordinate catalogs or in the file description - the file description must be in existence at the time of the restore.

   If space is not available on specified devices, the file is not restored, and a report is made in case the specifications have to be changed or space made available on the devices.

Files restored from a device save are assigned space only on devices saved for on this save. That is, if the save included the option DEVICE/DS1,DS2/, for example, a restore from this save assigns space only on DS1 and DS2. With these device assignments, the installation can continue to use a device save taken prior to the restore when another restore is needed. The device save for DS4, DS5, and DS6, for example, can be used to restore from when there is a failure of DS4, because no file will have been restored to DS4 that is not saved on the DS4, DS5, and DS6 save.

The restrictions on device files can be restored to, both those resulting from specifications in catalogs and those required because the restore is from a device save, can be removed by use of the option RESET/DEVICE/.

* No space is released when a device restore is performed. It is assumed that if a new pack is provided for the device, it will have been initialized.

If no specifications - either in existing or saved catalogs - control device assignment, the restore uses the assignment procedure used at file create that assigns space on any devices with suitable allocation unit and available space.

6. If no file description exists for the file and the saved version specifies duplication, space for both original and duplicate is assigned, using device specifications for both original and duplicate, if any, in the saved file description. Then the saved version is written to both spaces.

   If the saved file description specifies duplication but an existing file description for the file does not, only one write of the file content occurs, of course, since the existing file description controls the restoring.

7. If the restore is from a save done under the DEVICE/pack/option, references on fixed devices to catalogs on the restored pack are not created when they do not currently exist.

8. Restoring from save(s) performed under the INCR option requires that the user present first the most recently performed incremental save, then less recently performed saves in reverse chronological order, and finally a total save.

   File restoring is under control either of currently existing catalogs or the catalogs on the first presented incremental save. Only the most recent version of the file content for a file is restored.

9. Restoring from save(s) performed under the SINCE option similarly requires use of the most recent save first, less recent saves (if any) next, and finally a total save.

10. If a file was abort locked or had defective or unreadable space at the time of the save from which it is restored, the file description is marked to show the file is abort locked or has defective space. If the currently existing file description shows the file abort locked or has defective space, such an indication is removed unless the save is also so marked. A report is made of any such exceptions in case earlier saves are to be used or recovery action initiated or discontinued.

11. When a file is restored using the DEVICE option, all of the file is restored including parts on devices not specified, if any. An exception is when FMS journaling or duplication was specified for the file at the time of the save. Then only the part of the file on specified device(s) is restored since rollforward using journals can be relied on to bring up to date the part restored or duplication ensures that the part is up to date.

12. When rollforward by means of journals can be done to bring the file up to date, a report of any such files for the user that have been restored is made. The report shows whether the rollforward is optional or is required because only part of the file was restored from a save mode while the file was write busy. If rollforward is required, and the file description shows any ABORT option was specified, the abort lock is set.

13. Notice that saves done under INCR or SINCE options are not made while the file is write busy unless journaling is specified for the file. Files restored from saves not done under these options, however, can be made while the files are write busy even though journaling is not specified for them. A report is made of restores for the user of such files in case a restore from an earlier save is to be performed or recovery action initiated, and the file description is marked abort locked if any ABORT option is specified.

14. A restore uses a save produced by either a user save or an installation SAVEMAST. Presumably, only the installation has access to the saves produced by SAVEMAST so that user restores from such saves can be made on request.

15. The date and time of last write to the file at the time of the save from which the file is restored is set as the date and time of last change to the file. Hence it can be relied on to indicate what version of the file content is provided by the restore.

16. Multiple restore directives should be avoided if at all possible. Options such as DOONLY or DONTDO should be used in their place. If multiple restore directives are used and the activity is processing a multireel volume containing "save" information, it becomes necessary for the operator to remount reel one for each directive. If alternate logical unit designators are used in the $ TAPE control card, it is likely that initiate interrupt errors will occur since unit switching logic uses the rewind/standby command.

17. When the restore is done under the DEVICE option, catalogs for the user must currently exist because no catalogs will be restored from the save. Note the distinction between a device restore and a restore from a device save.

18. The user identification of the user name of the qualified file name of the save must be used with the NEWNAM option. If the NEWNAM option was used on the save but not on the restore, then the user identification of the user name of the qualified file name being restored should be used as in a normal restore.

19. Catalog or file names in the directive and in DOONLY and DONTDO clauses must also be present in the save file to be restored; otherwise the restore aborts.

20. The RESET/RFILES/ applies only to the files that are not already present in the systems. This option forces the program to set the file restore indicator in the off position. Care should be exercised when using this option because it may cause the file not to be saved on a succeeding since type of save.

21. Do not use more than one RESTORE directive in a given activity.

Examples

1. RESTORE CORPORATE$SEPT25,DONTDO/CUSTOMER/,
LISTOPT/YES/

All files for the user named CORPORATE except the user named CUSTOMER are restored. A listing of restored files is provided. If CUSTOMER was a catalog, files subordinate to it would not be restored. Catalogs and file descriptions are restored and file spaces assigned only if those do not exist at the time of the restore, otherwise the existing catalogs, file descriptions, and files spaces are used.

2. RESTORE CLASS21/PROB1INPUT

The file PROB1INPUT cataloged under the user named CLASS21 is restored. If the restore is from a save that used the NEWNAM option, then the user identification must be the same for CLASS21 as in a normal restore.

3. RESTORE INSTALL/SYSA,DOONLY/ACT,EQUIP,PERS/

The three files named ACT, EQUIP, and PERS subordinate to the catalog INSTALL/SYSA are restored.

4. RESTORE CLASS21,SINCE/070673,1200/,LISTOPT/YES/

Files for the user named CLASS21 that at the time of the save had been allocated or whose file descriptions had been modified since 6 July 1973 are restored and a list is provided of each that is restored.

5. RESTORE SWEETWATER/POWER$GEN4,DEVICE/DS3/,DONTDO/WELLS/

Files for the user named SWEETWATER that are subordinate to the catalog POWER except the file named WELLS, that were assigned space on the device named DS3 at the time of the save are restored. If a file is journaled, only the part on the device is restored (and ROLLFWD is expected to be used to bring it up-to-date with parts on other devices). Otherwise, all of the file is restored including parts on other devices.

6. RESTORE J.P.JONES/RMVBL

Files subordinate to the catalog RMVBL for the user named J.P.JONES are restored. These files are on a structured removable pack that must be mounted at the time of the restore. (Refer to Example 3 under the Catalog Create directive.)

7. RESTORE SALTRIVER,NEWNAM/CORPORATE/

All catalogs and files are restored to the user named CORPORATE. The user identification for SALTRIVER should be used.

SECTION XII

MME GEFSYE USE FOR SERVICE CALLS

SERVICES PROVIDED

Any job, privileged or not, can call on the File Management Supervisor to provide cataloging or protection services in the course of execution of the job. A MME GEFSYE is used for each such call for service, with a function code in the MME calling sequence used to distinguish a request for one service from another.

The function codes and their corresponding cataloging and protection service requests are as follows:

Cataloging

2 = catalog create

3 = file create

9 = file delete (purge)

10 = catalog modify

11 = file modify

22 = file delete (release)

23 = file query

28 = security lock

29 = abort lock

38 = TDS lock

Protection

30 = read from duplicate set and reset

30 = reserve pages

30 = specify TDS functions

34 = identify defective space

35 = replace and/or re-mark defective space

40 = cancel changes

41 = regard changes as complete

43 = provide allocated file information

44 = user supplied befores

46 = file query for allocated file

MMEs for cataloging services always include the name of the catalog or file for which some service is being requested. MMEs for protection services, by contrast, identify a file to be serviced by referencing the allocation for that file. Setting and resetting of security or abort locks is provided in both ways; the file can be identified either by name or allocation.

When the file is identified by name, all of the permission checking performed for an allocation must be performed to determine if the service requested by the MME is authorized. Checking of permission requires that user making the request be identified. When the request is received from a non-time sharing job, the user name appearing on the $ USERID submitted with the job is used. In addition, when the UMC name of the file name field has -1 in the first of the two words, the user name on the $ USERID card submitted with the job is used as the UMC name.

All catalog names, file names, passwords, user names, and volume names should be in BCD, left justified and blank filled.

The MMEs for cataloging services are issued with the same calling sequences by the Time Sharing Executive, on behalf of subsystems that may call the Executive with a corresponding DRL. In this case, the user name to identify the user is the one provided by the terminal operator on log-on, and its address is inserted by the Time Sharing Executive in the MME calling sequence.

TSS System Programmer's Reference Manual should be consulted for the formats of DRLs to the Time Sharing Executive, or commands to Time Sharing subsystems that issue these DRLs, which in turn cause the Time Sharing Executive to issue the corresponding MME.

RETURN WORDS

A two-word return space is referenced in each MME calling sequence. On call, the first bit of the first word is off. On return, that bit is set on to serve as an indicator that the service has been performed.

A MME GEROAD is issued by FMS on behalf of the caller. This prevents the program from being dispatched to and hence being able to access the buffer, if one is provided with the call, until the requested service has been done or an error reported. FMS can read records into the buffer containing passwords the caller should not see. FMS then clears the buffer before returning. No MME GEROAD is issued for recognized privileged programs, however, so these must either GEROAD themselves or test the return word for completion. Courtesy calls specified in MME GEFSYE calling sequences are honored only for privileged programs. Courtesy call addresses are ignored for nonprivileged programs.

If an error was encountered in trying to perform the service, a binary code is returned in bits 1-11 of the first return word to show what error was encountered (see Appendix F). If this field is zero, there was no error.

For cataloging services, if an error in the file or catalog name was encountered, the address of the part of the name in error is returned in bits 18-35 of the first return word.

The format for the return words, then, is:

```
Word 0, bit 0 = 0: if service not yet finished
              or
              = 1: if service finished

        bits 1-11 = 0: if no error
                  or
                  = 1: error code

        bits 18-35: address of part of name in error

Word 1, Data Control Word that points to buffer and gives length of
        BCD error message text returned in user buffer; or bits 0-5
        contain device type, bits 6-35 contain pack serial number.
```

CATALOGING MMES

NOTE: Devices to be used with these MMEs are in some cases restricted:

For Series 6000 Only	For Series 60 Only	For Series 6000 and Series 60
DSS167	MS0310	DSS180
DSS170	MS0400	DSS181
DSS270	MS0500	DSS190
167PK	310PK	DSS191
170PK	400PK	MS0450
BSS001	500PK	180PK
	MTS67	181PK
	MTS69	190PK
	TAPE27	191PK
	TAPE29	450PK
		TAPE
		TAPE7
		TAPE9
		MTS57
		MTS59

Common Codes for Cataloging MME GEFSYEs

1. General permissions code (in octal); use one of these or OR together those being specified:

APPEND	=	1000
CREATE	=	0010
EXECUTE	=	0400
LOCK	=	0040
MODIFY	=	7754
PURGE	=	7604
READ	=	4400
RECOVERY	=	7404
WRITE	=	7400

2. Device class code (in octal); use one of these:

DSS167	=	42
DSS170	=	45
DSS180	=	46
DSS181	=	60
DSS190	=	61
DSS191	=	62
DSS270	=	50
MS0310	=	63
MS0400	=	64
BSS001	=	52
MS0450	=	65
MS0500	=	66

3. Volume type code (in octal); use one of these:

167PK	=	42
170PK	=	45
180PK	=	46
181PK	=	60
190PK	=	61
191PK	=	62
310PK	=	63
400PK	=	64
450PK	=	65
500PK	=	66
TAPE	=	10
TAPE7	=	11
TAPE9	=	12
TAPE27	=	15
TAPE29	=	16
MTS69	=	12
MTS57	=	11
MTS59	=	12
MTS67	=	11

4. Density code (in octal); use one of these:

DEN0	=	00
DEN2	=	04
DEN6	=	10
DEN8	=	20
DEN16	=	44
DENSTD	=	74

5. Common error code returns for cataloging MMEs (bits 1-11 of first return word (octal)):

1 Requesting user name not entered in System Master Catalog (although it was when job was submitted).

3 Requesting user does not have permission to perform requested service.

5 One of the names of a catalog or file cannot be located (address of name is in bits 18-35 of first return word).

6 Space for catalog or file description is required but not available.

11 Name specified for creation, or new name for modification, is already in use for a file or catalog subordinate to the catalog to which the created or modified is subordinate.

14 One of the passwords for a catalog or file is incorrect (perhaps at this time), missing, or given when none is specified (address of password is in bits 18-35 of first return word).

22 Space was required for catalog or file or was to be released but a checksum error in space availability table on device prevents the space assignment or release.

24 Same as 22 except checksum error is in table in memory.

25 Required removable structured disk pack is not mounted. (Second return word contains device type and pack serial number.)

52 Required fixed device is either not available or entry for device with specified name cannot be found.

Additional error codes associated with specific MMEs appear in on return section of individual MME descriptions.

6. The job is aborted with GCOS abort message, code 81, INVALID MME PARAMETER, for any of the following reasons:

a. invalid function code

b. zero buffer or option address

c. name or password with illegal character

File Create

```
          MME    GEFSYE
          ZERO   courtesy call or zero, arglist
          ZERO   3,buffer
```

arglist

```
     ZERO   return, user name
     ZERO   file name, permission
     ZERO   option, zero
```

buffer

```
     BSS    355
```

return

```
     DEC    0,0
```

user name

```
     words 0-1, user identification (name)
```

file name

```
     words 0-1, UMC name
     words 2-3, UMC password or blanks
     next 4i words, catalog name and password or blanks for 0,1,...,i
               intervening catalogs
     next 4 words, file name and password or blanks
         or
     word 4i+3, bits 0-17: number of entries in password list
                bits 18-35 = -1
     word 4i+4, bits 0-17: password list address
     last word, -1
```

password list

```
     word 0-1,  password
     word 2,  time when password starts applying
     word 3,  time when password stops applying

     (times in 1/64 milliseconds since midnight)
```

permission

```
     bits 0-11: general permissions code
     bits 18-20: access code

     6 = Concurrent
     6 = Monitor (also requires bit 24 = 1 in option -1)
     0 = Normal
     2 = Read while write
```

option -1 (This word required only when FMS protection is specified for file.)

bits 0-17: user specified device or device class for duplicate of file
bits 0-11: zero
bits 12-17: device class code
or
bits 0-17: device name (in BCD)
or
bits 0-17 = 0: no user specification

bit 18 = 1: RDERR/DUP/
bit 19 : reserved
bit 20 = 1: RDERR/JOURNAL/
bit 21 : reserved
bit 22 = 1: INCRSAVE/NO/
bit 23 = 1: ABORT/LOCK/
bit 24 = 1: ACCESS/MONITOR/
bit 25 = 1: VERIFY/YES/
bit 26 = 1: ABORT/ROLLBACK/
bit 27 : reserved
bit 28 = 1: AUDIT/DENIED/
bit 29 = 1: AUDIT/ALL/
bits 30-35: number of sectors in page; if zero, 5 is assumed

option

bit 0 = 1: reserved
bit 1 = 1: define file as random (hence disallow sequential access)
bit 2 = 1: file is TSS-created
bit 3 = 1: file is I-D-S/I
bit 4 = 1: sizes (in option +1) specified in llinks
bit 5 = 1: file already on tape or nonstructured disk pack
bit 7 = 1: attributes present
bit 8 = 1: user specified information present
bits 9-12: reserved
bit 13 = 1: FMS protection options specified (in option -1)
bit 14 = 1: duplicate device specified
bits 15-17: reserved
bits 18-35: user specified device or device class for original:
bits 18-29 = 0
bits 30-35: device class
or
bits 18-35: device name
or
bits 18-35 = -1: no user specification
or when bit 5 = 1 in option word
bits 18-35: volinfo address

option +1

bits 0-17: initial file size
bits 18-35: maximum file size (zero if unlimited)
or if bit 5 = 1 in option word
bits 0-5: volume type (code)
bits 6-35: volume serial number (in BCD, left-justified, blank-filled)

option +2

triplets of words for 0,1,... specific permissions
word 0-1, user name
word 2, specific permissions for user:

bits 0-11: general permissions code
bits 12-16: reserved
bit 17 = 1: EXCLUDE
bits 18-35: reserved

option +2 + triplets

(after 0,1,...triplets): -1

next word (only if bit 8 = 1 in option word)

```
2 words:
   word 0, bits 1-35: user supplied information
   word 1, -1
```

next word (only if bit 7 = 1 in option word)

```
attributes for I-D-S/I files, 5 words:
   word 0, bits 0-17: first page number in range (binary)
           bits 18-35: last page number in range (binary)
   word 1, bit 0 = 1: MULTIUSER/YES/
           bits 12-17 = 1: COEXISTENCE/YES/
           bits 18-35: number of words in page (binary)
   word 2, maximum number of lines per page (binary)
   word 3, inventory action (BCD percent or NO)
   word 4, base size (binary)
```

last word

-1

volinfo (when bit 5 = 1 in option word)

```
word 0-1, volume name
word 2, bits 0-5: density code (for magnetic tape only)
```

on return:

If the return word shows successful completion, a file description will have been created, and except for a file already on magnetic tape or nonstructured disk pack, file space will have been assigned. An allocation for the file, however, has not been provided. A MME GEMORE is required to provide the allocation necessary before any writing to the file can occur.

On unsuccessful completion, in addition to any of the common cataloging MME errors, the following specific to File Create directive errors may occur.

10 NO SPACE FOR FILE ON DEVICE YY
File space not available.

13 SPACE REQUEST GR THAN ALLOWED
File space requested together with what is already assigned for files and catalogs for user exceed user's allowance on all fixed devices or on removable structured disk pack.

In addition, the job is aborted with GCOS abort message, code 81, INVALID MME PARAMETER, if the initial size is zero or greater than a specified maximum size.

The calling sequence is altered in processing. In the option word, bit 15 is set on if a superordinate catalog specifies a default device for files, and in option+1, bits 0-17 are set to the actual space assigned that may differ from that requested because of file duplication and/or allocation in units other than llink on devices used.

File Modify

```
          MME     GEFSYE
          ZERO    courtesy call or zero, arglist
          ZERO    11, buffer
```

arglist

```
     ZERO   return, user name
     ZERO   file name, permission
     ZERO   option, new name
```

buffer

```
     BSS    355
```

return

```
     DEC    0,0
```

user name

words 0-1, user identification (name)

file name

```
     words 0-1, UMC name
     words 2-3, UMC password or blanks
     next 4i words, catalog name and password or blanks for 0,1,...,i
               intervening catalogs
     next 4 words, file name and password or blanks
     last word, -1
```

permission

```
     bits 0-11: new general permissions code
      or
     bits 0-17: -1 if general permissions not modified

     bits 18-23: new access mode code
      or
     bits 18-35: -1 if access mode not modified
```

option -2 (This word required only when new FMS protection is specified for file.)

```
     bits 0-17: 777777
     bits 18-35: ANDed product of codes to show which protection
                 options are changed (in octal):
```

ABORT:	566777
ACCESS:	773777
AUDIT:	777477
INCRSAVE:	757777
PAGE:	777700
RDERR:	277777
VERIFY:	775777

option -1 (This word required only when new FMS protection is specified for file.)

bits 0-17: user specified device or device class for duplicate of file
bits 0-11: zero
bits 12-17: device class code
or
bits 0-17: device name (in BCD)
or
bits 0-17 = 0: no user specification

bit 18 = 1: RDERR/DUP/
bit 19 : reserved
bit 20 = 1: RDERR/JOURNAL/
bit 21 : reserved
bit 22 = 1: INCRSAVE/NO/
bit 23 = 1: ABORT/LOCK/
bit 24 = 1: ACCESS/MONITOR/
bit 25 = 1: VERIFY/YES/
bit 26 = 1: ABORT/ROLLBACK/
bit 27 : reserved
bit 28 = 1: AUDIT/DENIED/
bit 29 = 1: AUDIT/ALL/
bits 30-35: number of sectors in page; if zero, 5 is assumed

option

bit 1 (only if bit 9 is on) = 0: mode is sequential; = 1: mode is random
bits 2-3 : reserved
bit 4 = 1: new maximum size (in option +1) specified in llinks
bits 5-6 : reserved
bit 7 = 1: new attributes present
bit 8 = 1: new user specified information present
bit 9 = 1: new mode is specified
bit 10 : reserved
bit 11 = 1: set file not busy specified
bit 12 = 1: reset abort lock off specified
bit 13 = 1: new FMS protection options specified (in option -1)
bit 14 = 1: duplicate device specified
bits 15-17: reserved
bit 18 = 1: set primary file copy (COPYA) defective indicator (DUP option)
bit 19 = 1: set primary file copy (COPYA) not defective (DUP option)
bit 20 = 1: set secondary file copy (COPYB) defective indicator (DUP option)
bit 21 = 1: set secondary file copy (COPYB) not defective (DUP option)
bits 22-35: reserved

Note: Options specified by bits 11, 12, 18, 19, 20, and 21 are allowed only if the content of .SACTY is 'FILSYS'. If not, the attempt to use these options results in error code 3 (PERMISSIONS DENIED).

option +1

bits 0-17: reserved
bits 18-35: new maximum size (zero if no change)
or
bits 18-35: -1 if unlimited

option +2

triplets of words for 0,1,... specific permissions to be modified
word 0-1, user name
word 2, specific permissions for user:

bits 0-11: general permissions code
bits 12-16: reserved

```
bit 17 = 1: EXCLUDE
 or
bits 0-17 = 0 (remove name unless bit 18 = 1 in which case name is
              kept in anticipation that specific permissions for user
              will subsequently be provided)
bits 18-35 : reserved
```

option +2 + triplets

```
(after 0,1,... triplets): -1
```

next word (only if bit 8 = 1 in option word)

```
2 words:
word 0, bits 1-35: user supplied information
word 1, -1
```

next word (only if bit 7 = 1 in option word)

```
attributes, for an I-D-S/I file, 5 words:
word 0, bits 0-17: first page number in range (binary)
        bits 18-35: last page number in range (binary)
word 1, bit 0 = 1: MULTIUSER/YES/
        bits 12-17 = 1: COEXISTENCE/YES/
        bits 18-35: number of words in page (binary)
word 2, maximum number of lines per page (binary)
word 3, inventory action (BCD percent or NO)
word 4, bits  6-11: area number (binary)
        bits 18-35: base size (binary)
```

last word

```
-1
```

new name

```
words 0-1, new name for file
     or
word 0, -1 if name not to be modified

words 2-3, new password for file
     or
word 2, -1 if password not to be modified
     or
words 2-3, blanks if existing password to be removed
     or
word 2, bits 0-17: count of number of passwords
        bits 18-35: -1
word 3, bits 0-17: password list address
```

password list

```
words 0-1,  password
word  2,  time when password starts applying
word  3,  time when password stops applying

(times in 1/64 milliseconds since midnight)
```

on return

If the return word shows successful completion, the file description will have been modified.

On unsuccessful completion, in addition to the common cataloging MME errors, the following may occur:

4 FILE BUSY; TRY LATER
File is allocated to another job and since the modification involves a change in file protection, modification must wait until file is not allocated.

12 SIZE REQUEST LS THAN ALLOCATE
The new maximum file size specified is less than the current size.

Note the additional reasons given for File Create that can cause the job to be aborted.

File Delete

```
          MME     GEFSYE
          ZERO    courtesy call or zero, arglist
          ZERO    9 or 22, buffer
```

arglist

```
     ZERO    return, user name
     ZERO    file name, 0
```

function code (decimal)

```
      9 = purge (write zeroes in file space before release)
     22 = release
```

buffer

```
     BSS     355
```

return

```
     DEC     0,0
```

user name

words 0-1, user identification (name)

file name

```
     words 0-1, UMC name
     word 2-3, UMC password or blanks
     next 4i words, catalog name and password or blanks for 0,1,...,i
               intervening catalogs
     next 4 words, file name and password or blanks
     last word, -1
```

on return

If return word shows successful completion, file description is deleted and file is either released or purged. If the file is busy, however, the deletion is deferred until the file is no longer busy.

See "Common Codes for Cataloging MME GEFSYEs" for error codes and reasons for unsuccessful completion.

File Query

```
          MME     GEFSYE
          ZERO    courtesy call or zero, arglist
          ZERO    23, buffer
```

arglist

```
     ZERO    return, user name
     ZERO    file name, (info)
```

buffer

```
     BSS     355
```

return

```
     DEC     0
     VFD     1/1 (file written to), 35/user supplied information
```

user name

words 0-1, user identification (name)

file name

```
     words 0-1, UMC name
     words 2-3, UMC password or blanks
     next 4i words, catalog name and password or blanks for 0,1,...,i
               intervening catalogs
     next 4 words, file name and password or blanks
     last word, -1
```

on return

On successful completion, bits 18-35 of second arglist word have address, within 355-word buffer, of information about file. Format of this information is:

```
     word 0, bits 0-5: device class code of file
             bits 6-35: serial number of last save of file

     word 1, file creation date (in BCD, MMDDYY)
     word 2, date of last allocation to file (in BCD, MMDDYY)
     word 3, date of last change to file (in BCD, MMDDYY)
     word 4, bits 0-17: time of last change to file (in binary, 512/1000 second
                        since midnight)
             bits 18-35: number of allocations to file (modulo 262144)
     word 5, bits 0-17: low order 18 bits of current file size (in llinks)
             bits 18-35: maximum file size allowed (zero means unlimited)

     word 6, bits 0-8  : number of writers allocated
             bits 9-17 : number of readers allocated
             bits 18-27: general permissions
             bit 28 = 1: specific permissions present
             bit 29 = 1: timed passwords present
             bit 30 = 1: file definition catalog continuation
             bit 31 = 1: one or more space descriptors marked defective
             bits 32-35: high order 4 bits of current file size
```

```
word 7, bit 0     : reserved
        bit 1   = 1: random
        bit 2   = 1: ASCII file
        bit 3   = 1: I-D-S/I file
        bit 4   = 1: maximum file size is in llinks
        bit 5   = 1: file on nonstructured device (reel or pack)
        bit 6      : reserved
        bit 7   = 1: attributes present
        bit 8   = 1: user-specified information present
        bit 9   = 1: user-specified class of device for original file
        bits 10-11: access mode
                         0, NORMAL
                         1, READ WHILE WRITE
                         2, reserved
                         3, CONCURRENT

        bit 12 = 1: abort lock set by MME
        bit 13 = 1: abort lock on
        bit 14 = 1: security lock on
        bit 15 = 1: restore lock on
        bit 16 = 1: file in recovery allocation
        bit 17 = 1: file was saved while write busy
        bit 18 = 1: RDERR/DUP/
        bit 19    : reserved
        bit 20 = 1: RDERR/JOURNAL/
        bit 21    : reserved
        bit 22 = 1: INCRSAVE/NO/
        bit 23 = 1: ABORT/LOCK/
        bit 24 = 1: ACCESS/MONITOR/
        bit 25 = 1: VERIFY/YES/
        bit 26 = 1: ABORT/ROLLBACK/
        bit 27 = 1: file changed since last save
        bit 28 = 1: AUDIT/DENIED/
        bit 29 = 1: AUDIT/ALL/
        bits 30-35: number of sectors in page (zero = not defined)

words 8-33, mapping or mounting information

if nonstructured: (mounting information)
     words 8-9: volume name
     word 10:
        bits 0-5 : volume type
        bits 6-35: volume serial number

     word 11:
        bits 0-5: density code (if volume type is tape)
        bits 6-35: reserved

if structured: (mapping information, first descriptor must be device,
        others may be either space or device; first 26 descriptors given)

     device:
        bits 0-3   : 0101 binary for device descriptor
        bits 2-5   : reserved
        bits 6-35  : pack serial number
          or
        bits 6-17  : reserved
        bits 18-35: device name

    space:
       bits 0-1   : 10 or 00 (if last)
       bit 2      : 1 if defective
       bits 3-17 : number of llinks
       bits 18-35: starting llink number
```

Bit 0 of second return word is 0 if the file has not been written to, is 1 if file has been written to. In addition, bits 1-35 of second return word have user supplied information.

NOTE: A permission denied error is returned if the user name (specified in $ USERID card) is not the same name as the UMC name.

Catalog Create

```
          MME     GEFSYE
          ZERO    courtesy call  or zero, arglist
          ZERO    2, buffer
```

arglist

```
     ZERO    return, user name
     ZERO    catalog name, permission
     ZERO    option, 0
```

buffer

```
     BSS     355
```

return

```
     DEC     0,0
```

user name

```
     words 0-1, user identification (name)
```

catalog name

```
     words 0-1, UMC name
     words 2-3, UMC password or blanks
     next 4i words, catalog name and password or blanks for 0,1,...,i
               intervening catalogs
     next 4 words, catalog name and password or blanks
         or
     word 4i+3, bits 0-17: number of entries in password list
                bits 18-35 = -1
     word 4i+4, bits 0-17: password list address
     last word, -1
```

permission

```
     bits 0-11: general permissions code
```

password list

```
     word 0-1, password
     word 2, time when password starts applying
     word 3, time when password stops applying

     (times in 1/64 milliseconds since midnight)
```

option

```
bit 5 = 1 catalog to be assigned to removable structured disk pack
bits 18-35: (when bit 5 = 0) device name or class to assign subordinate
            files to in lieu of specification in subordinate catalog or
            file
     bits 18-29 = 0
     bits 30-35: device class code
       or
     bits 18-35: device name
       or
     bits 18-35 = 0: no specification if bit 5 = 1
```

option +1

```
(when bit 5 = 1)
bits 0-5: volume type code
bits 6-35: volume serial number
  or
bits 0-35 = 0
```

option +2

triplets of words for 0,1,... specific permissions.

```
words 0-1, user name
word 2, specific permissions for user:
     bits 0-11: general permissions code
     bits 17=1: EXCLUDE
```

option +2 + triplets

(for 0,1,... triplets): -1

on return

On successful completion, catalog is created.

The following error, in addition to those listed as common for cataloging MMEs, may occur on unsuccessful completion:

01 USER-ID NOT IN MASTER CATALOG
User name not entered in Pack Master Catalog for removable structured disk pack catalog is to be created on.

Catalog Modify

```
          MME     GEFSYE
          ZERO    courtesy call or zero, arglist
          ZERO    10, buffer
```

arglist

```
     ZERO    return, user name
     ZERO    catalog name, permission
     ZERO    option, new name
```

buffer

```
     BSS     355
```

return

```
     DEC     0,0
```

user name

```
     words 0-1, user identification (name)
```

catalog name

```
     words 0-1, UMC name
     words 2-3, UMC password or blanks
     next 4i words, catalog name and password or
             blanks for 0,1,...,i intervening catalogs
     next 4 words, catalog name and password or blanks
     last word, -1
```

permission

```
     bits 0-11: general permissions code
       or
     bits 0-17: -1 if general permissions not to be modified
```

option

```
     words 0-1, not used
```

option +2

```
     triplets of words for 0,1,... specific permissions to be modified

     word 0-1, user name
     word 2, specific permissions for user:

          bits 0-11: general permissions code
          bit 17 = 1: EXCLUDE
           or
          bits 0-17 = 0 (remove name unless bit 18 = 1 in which case name is
                        kept in anticipation that specific permissions for user
                        will subsequently be provided)
```

option +2 + triplets

(for 0,1,... triplets): -1

new name (the UMC name cannot be modified)

```
words 0-1, new name for catalog
          or
word  0, -1 if name not to be modified
words 2-3, new password for catalog
          or
word  2, -1 if password not to be modified
          or
words, 2-3, blanks if existing password to be removed
          or
word  2, bits 0-17: count of number of passwords
         bits 18-35: -1
word  3, bits 0-17: password list address
```

password list

```
words 0-1, password
word  2, time when password starts applying
word  3, time when password stops applying
```

(times in 1/64 milliseconds since midnight)

on return

On successful completion, catalog will be modified.

See "Common Codes for Cataloging MME GEFSYEs" for error codes and reasons for unsuccessful completion.

Security or Abort Lock Set and Reset

```
MME     GEFSYE
ZERO    courtesy call or zero, arglist
ZERO    28, 29, or 38, buffer
```

arglist

```
ZERO    return, user name
ZERO    catalog or file name, 0, 1, or 2 (subfunction code)
ZERO    option, 0
```

function code (decimal)

28 = security lock
29 = abort lock
38 = TDS lock

subfunction code

0 = bits 0-17 have address of catalog or file name
1 = bits 0-17 have address of PAT pointer
2 = bits 0-17 have address of word with file code in bits 24-35

buffer

```
BSS     355
```

return

```
DEC     0,0
```

user name

words 0-1, user identification (name)

catalog or file name (only file name can be provided for abort lock)

words 0-1, UMC name
words 2-3, UMC password or blanks
next 4i words, catalog name and password or blanks
    for 0, 1,...,i intervening catalogs
next 4 words, catalog or file name and password or blanks
last word, -1

option

bit 0 = 1: for setting lock on
or
bit 0 = 0: for resetting lock off

on return

Lock will be set or reset on successful completion.

See "Common Codes for Cataloging MME GEFSYEs" for error codes and reasons for unsuccessful completion.

Common Error Code Returns for Protection MMEs

Error code in bits 1-11 of first return word.

(octal)	
17	SEEK ERROR ON DEVICE XXX SA = NNN......NNN Allocation has bad seek address for file description.
42	INVALID FILE CODE OR PAT PTR File with file code not located.
43	INVALID CATALOG BLOCK ADDRESS Address in allocation is not of a file description.
61	$FSYS HAS BEEN ENABLED File with file code is not a protected file.
62	ILLEGAL SUBFUNCTION CODE Incorrect subfunction code.
63	FILE NOT BEING MONITORED File is not being monitored.
64	DEADLOCK ON PAGE REQUEST One of the pages cannot be reserved.
65	PAGE CURRENTLY BUSY One of the pages cannot be reserved now. Repeat request later.
66	FILE NOT DUPLICATED File with specified file code is not duplicated.
67	TDS MON ALLOC ERROR File with file code has been deallocated by means of MME GERELS.

Additional error codes associated with specific MMEs appear in on return section of MME descriptions.

Identify Defective Space

```
          MME     GEFSYE
          ZERO    courtesy call or zero, arglist
          ZERO    34, buffer
```

arglist

```
     ZERO    return, 0 or 1 (subfunction code)
     VFD     6/0,H12/file code,18/0
                 or
     VFD     18/0,18/address of PAT pointer
```

buffer

```
     BSS     355
```

return

```
     DEC     0,0
```

subfunction code

0 = identify defective spaces
1 = identify all spaces

on return

On successful return, bits 18-35 of first arglist word have address in buffer of the list of defective spaces in file as well as information pertinent to file recovery.

space list

each defective space has three words:

```
word 0, bits 0-3 = 0101
        bits 4-17 = 0
        bits 18-35: name of device
          or
        bits 4-5 = 0
        bits 6-35: pack serial number
word 1, bits 0-2 = X0Y (X=0 if last descriptor, Y=1 if defective)
        bits 3-17: number of defective llinks
        bits 18-35: starting llink number
word 2, initial page number in defective space
```

last word

(after 0,1,... triplets of defective space): -1 next word

6 words of recovery information:

```
word 0, bits 0-5: volume type
        bits 6-35: volume serial number of last save of file
word 1, date of last change to file (mmddyy)
word 2, bit 0 = 1: abort lock set by MME
        bit 1 = 1: abort lock set by job or system or protection failure
        bit 2-9   : reserved
        bit 10 = 1: last save was installation save
        bit 11    : reserved
        bit 12 = 1: file has changed since last save
        bits 13-15: reserved
        bit 16 = 1: file is on structured pack
        bit 17 = 1: file was save on write busy
        bits 18-35: time last job writing to file deallocated
                    (milliseconds since midnight divided by 512)
word 3, bits 0-17: reserved
        bit 18 = 1: RDERR/DUP/
        bit 19    : reserved
        bit 20 = 1: RDERR/JOURNAL/
        bit 21 = 1: duplicate file written to
        bit 22 = 1: INCRSAVE/NO/
        bit 23 = 1: ABORT/LOCK/
        bit 24 = 1: ACCESS/MONITOR/
        bit 25 = 1: VERIFY/YES/
        bit 26 = 1: ABORT/ROLLBACK/
        bit 27    : reserved
        bit 28 = 1: AUDIT/DENIED/
        bit 29 = 1: AUDIT/ALL/
        bits 30-35: number of sectors per page
word 4, date file created (mmddyy)
word 5, bits 0-23: time at initialization (milliseconds since
                   midnight divided by 8) and overflows of creation
                   ordinal
        bits 24-35: prior creations since this restart
```

Security or Abort Lock Set and Reset

```
          MME     GEFSYE
          ZERO    courtesy call or zero, arglist
          ZERO    28, 29, or 38, buffer
```

arglist

```
     ZERO    return, user name
     ZERO    catalog or file name, 0, 1, or 2 (subfunction code)
     ZERO    option, 0
```

function code (decimal)

```
     28 = security lock
     29 = abort lock
     38 = TDS lock
```

subfunction code

```
     0 = bits 0-17 have address of catalog or file name
     1 = bits 0-17 have address of PAT pointer
     2 = bits 0-17 have address of word with file code in bits 24-35
```

buffer

```
     BSS     355
```

return

```
     DEC     0,0
```

user name

words 0-1, user identification (name)

catalog or file name (only file name can be provided for abort lock)

```
     words 0-1, UMC name
     words 2-3, UMC password or blanks
     next 4i words, catalog name and password or blanks
             for 0, 1,...,i intervening catalogs
     next 4 words, catalog or file name and password or blanks
     last word, -1
```

option

```
     bit  0  = 1: for setting lock on
       or
     bit  0  = 0: for resetting lock off
```

on return

Lock will be set or reset on successful completion.

See "Common Codes for Cataloging MME GEFSYEs" for error codes and reasons for unsuccessful completion.

Common Error Code Returns for Protection MMEs

Error code in bits 1-11 of first return word.

(octal)

17 SEEK ERROR ON DEVICE XXX SA = NNN......NNN
Allocation has bad seek address for file description.

42 INVALID FILE CODE OR PAT PTR
File with file code not located.

43 INVALID CATALOG BLOCK ADDRESS
Address in allocation is not of a file description.

61 $FSYS HAS BEEN ENABLED
File with file code is not a protected file.

62 ILLEGAL SUBFUNCTION CODE
Incorrect subfunction code.

63 FILE NOT BEING MONITORED
File is not being monitored.

64 DEALLOCATION PAGE REQUEST
One of the pages cannot be reserved.

65 PAGE CURRENTLY BUSY
One of the pages cannot be reserved now. Repeat request later.

66 FILE NOT DUPLICATED
File with specified file code is not duplicated.

67 TDS MON ALLOC ERROR
File with file code has been deallocated by means of MME GERELS.

Additional error codes associated with specific MMEs appear in on return section of MME descriptions.

Identify Defective Space

```
          MME     GEFSYE
          ZERO    courtesy call or zero, arglist
          ZERO    34, buffer
```

arglist

```
     ZERO    return, 0 or 1 (subfunction code)
     VFD     6/0,H12/file code,18/0
                 or
     VFD     18/0,18/address of PAT pointer
```

buffer

```
     BSS     355
```

return

```
     DEC     0,0
```

subfunction code

0 = identify defective spaces
1 = identify all spaces

on return

On successful return, bits 18-35 of first arglist word has address in buffer of the list of defective spaces in file as well as information pertinent to file recovery.

space list

each defective space has three words:

```
     word 0, bits 0-3 = 0101
             bits 4-17 = 0
             bits 18-35: name of device
               or
             bits 4-5 = 0
             bits 6-35: pack serial number
     word 1, bits 0-2 = X0Y (X=0 if last descriptor, Y=1 if defective)
             bits 3-17: number of defective llinks
             bits 18-35: starting llink number
     word 2, initial page number in defective space
```

last word

(after 0,1,... triplets of defective space): -1

next word

```
6 words of recovery information:

word 0, bits 0-5: volume type
        bits 6-35: volume serial number last save of file
word 1, date of last change to file (mmddyy)
word 2, bit 0 = 1: abort lock set by MME
        bit 1 = 1: abort lock set by job or system or protection failure
        bit 2-9   : reserved
        bit 10 = 1: last save was installation save
        bit 11    : reserved
        bit 12 = 1: file has changed since last save
        bits 13-15: reserved
        bit 16 = 1: file is on structured pack
        bit 17 = 1: file was save on write busy
        bits 18-35: time last job writing to file deallocated
                    (milliseconds since midnight divided by 512)
word 3, bits 0-17: reserved
        bit 18 = 1: RDERR/DUP/
        bit 19 = 1: reserved
        bit 20 = 1: RDERR/JOURNAL/
        bit 21 = 1: duplicate file written to
        bit 22 = 1: INCRSAVE/NO/
        bit 23 = 1: ABORT/LOCK/
        bit 24 = 1: ACCESS/MONITOR/
        bit 25 = 1: VERIFY/YES/
        bit 26 = 1: ABORT/ROLLBACK/
        bit 27    : reserved
        bit 28 = 1: AUDIT/DENIED/
        bit 29 = 1: AUDIT/ALL/
        bits 30-35: number of sectors per page
word 4, date file created (mmddyy)
word 5, bits 0-23: time at initialization (milliseconds since
                   midnight divided by 8) and overflows of creation
                   ordinal
        bits 24-35: prior creations this restart
```

Replace And/Or Re-mark Defective Space

```
          MME    GEFSYE
          ZERO   courtesy call or zero, arglist
          ZERO   35, buffer
```

arglist

```
     ZERO   return, space
     VFD    6/0,H12/file code, 18/0
```

buffer

```
     BSS    355
```

return

```
     bits 0-17 = 0
     bits 18-35: subfunction code (octal)
               = 0: replace and re-mark not defective
               = 1: re-mark not defective
               = 2: replace and mark defective
               = 3: mark defective
```

return +1

```
     0
```

space

```
     word 0, bits 0-3 = 0101
             bits 4-17 = 0
             bits 18-35: name of device
               or
             bits 4-5 = 0
             bits 6-35: pack serial number
     word 1, bits 0-2 = X0Y (X=0 if last descriptor, Y=1 if defective)
             bit 3-17: number of defective llinks
             bits 18-35: starting llink number
     word 2, initial page number in defective space
```

on return

On successful completion, specified space will have been replaced and/or re-marked as no longer defective.

On unsuccessful completion, in addition to one of the errors common to protection MMEs, one of the following may be returned:

3 PERMISSIONS DENIED
Recovery allocation to file is required but file is not so allocated.

45 INVALID SPACE IDENTIFIER
The specified space is not marked defective in file description.

Read From Duplicate/Original

```
        MME     GEFSYE
        ZERO    zero, arglist
        ZERO    30,0
```

arglist

```
    ZERO     return, file code
    ZERO     2 or 3 (subfunction code), 0
```

return

```
    DEC      0,0
```

file code

```
    VFD      24/0,H12/file code
```

subfunction code

```
    2 = set to read from duplicate
    3 = reset to read from original
```

on return

On successful completion, next read will be from duplicate or original according to set or reset subfunction requested. If one copy is marked defective, the read from duplicate will attempt to read the defective copy and the read from original will read from the non-defective copy.

On unsuccessful completion, in addition to the errors listed as common to protection MMEs, one of the following may occur:

65 PAGE CURRENTLY BUSY
Request to switch to reading from duplicate or original when file I/O is outstanding or request to reserve pages of file when one of the pages is in concurrent use by a job that may complete before the requesting job.

66 FILE NOT DUPLICATED
File with specified file code is not duplicated.

NOTE: Even though the type of access to the file is specified (on the $ PRMFL control card or by a time sharing system command) or assumed (in the absence of a specification) to be sequential, random access is provided to the duplicate. Hence, when reading from the duplicate, the user must calculate initial and subsequent seek addresses rather than rely on the current position in the file being known and IOS providing seek addresses for "get next" or skip forward and backward commands.

Moreover, the current position as recorded for the original file, if access to it is sequential, is not affected by reading from the duplicate. Hence, when resetting to read from the original, the user can resume reading from the original at the position in the file last attained, prior to setting to read from the duplicate, again assuming that access to the file is sequential.

Regard Changes as Complete

```
          MME     GEFSYE
          ZERO    courtesy call or zero, arglist
          ZERO    41, buffer
```

arglist

```
     VFD   18/return,6/0,H12/file code
     DEC   -1
          or
     ZERO  return,-1 (to signify all files)
     DEC   -1
```

buffer

```
     BSS   (words per page + 35) (minimum of 355)
```

return

```
     DEC   0,0
```

on return

On successful completion, pages written only to collection file are written to protected file. If befores are kept in collection file, they are discarded. If ACCESS/MONITOR/ concurrency control is specified for the file, all files allocated to the activity that have ACCESS/MONITOR/ control have their changes regarded as complete.

On unsuccessful completion, in addition to errors common to protection MMEs, one of the following may occur:

17 SEEK ERROR ON DEVICE XXX SA = NNN......NNN
The seek error occurred on the collection file.

32 COLLECTION FILE ERROR
FMS cannot write to the collection file.

47 UNACCOUNTABLE ERROR
An illegal condition which the program logic cannot handle has occurred.

Cancel File Changes

```
        MME     GEFSYE
        ZERO    courtesy call or zero, arglist
        ZERO    40,buffer
```

arglist

```
    VFD    18/return, 6/0,H12/file code
    DEC    -1
           or
    ZERO   return, -1 (to signify all files)
    DEC    -1
```

buffer

```
    BSS    (words per page + 35) (minimum of 355)
```

return

```
    DEC    0,0
```

on return

On successful completion, if befores (ABORT/ROLLBACK/) are present, they are written to the file to roll back the changes made to the file. If afters (RDERR/JOURNAL/) are present with befores, then the afters are discarded. If afters are present and befores are absent, this function is equivalent in effect to function code 41 (Regard Changes as Complete). If TEST allocation is specified, the changes made since the beginning of the activity or prior function code 41 will be ignored. If ACCESS/MONITOR/ concurrency control is specified for the file, all files allocated to the activity that have ACCESS/MONITOR/ control have their changes cancelled.

On unsuccessful completion, in addition to errors common to protection MMEs, one of the following may occur:

02 I/O ERROR ON DEVICE XXX SA = NNN......NNN
An unrecoverable I/O error occurred.

17 SEEK ERROR ON DEVICE XXX SA = NNN......NNN
Error return on a seek calculation.

27 FILE IN DEFECTIVE STATUS
Can not write to the data file or its duplicate if present - ABORT/ROLLBACK/ option.

32 COLLECTION FILE ERROR
Can not write to the collection file - RDERR/JOURNAL/ option.

40 NEED MORE PAT SPACE
Can not build a PAT body - ABORT/ROLLBACK/ option.

42 INVALID FILE CODE OR PAT PTR
Can not locate data file PAT - ABORT/ROLLBACK/ option.

47 UNACCOUNTABLE ERROR
An illegal condition which the program logic cannot handle.

73 INVALID ARG LIST PARAMETER NO. 000
GEFSYE buffer exceeds upper address limits of program.

Reserve Pages from Concurrent Use

```
          MME    GEFSYE
          ZERO   courtesy call  or zero, arglist
          ZERO   30,0
```

arglist

```
     ZERO   return, file code
     ZERO   1,1
     VFD    12/0,24/page number in binary
```

return

```
     DEC    0,0
```

file code

```
     VFD    24/0,H12/file code
```

on return

On successful completion, the page is reserved to this job. The page numbers for a file start at page zero.

On unsuccessful completion, in addition to the errors common to protection MMEs, one of the following may occur:

63 FILE NOT BEING MONITORED
File is not being monitored (either job has exclusive allocation or file does not specify monitoring).

64 DEALLOCATION PAGE REQUEST
Request to reserve pages on file when one of the pages is in concurrent use by a job that cannot complete until requesting program completes.

65 PAGE CURRENTLY BUSY
Request to switch to reading from a duplicate or original when file I/O is outstanding or request to reserve pages of file when one of the pages is in concurrent use by a job that may complete before the requesting job.

Reserve Page for Shared Use

```
          MME       GEFSYE
          ZERO      0, arglist
          ZERO      30,0
```

arglist

```
     ZERO     return, file code
     ZERO     5(subfunction code),1
     DEC      page number
```

return

```
     DEC     0,0
```

file code

```
     VFD       24/0,H12/file code
```

on return

On successful completion, the page is reserved for shared use (read) to this job. Otherwise, the return is the same as for "Reserved Pages from Concurrent Use."

Release Non-Updated Page

```
          MME     GEFSYE
          ZERO    0, arglist
          ZERO    30, 0
```

arglist

```
     ZERO    return, file code
     ZERO    6(subfunction code),1
     DEC     page number
```

return

```
     DEC     0,0
```

file code

```
     VFD     24/0,H12/file code
```

on return

A successful return will be completed, unless invalid arguments are present. If the page was reserved for shared use, the reservation will be released so other programs may access the page.

Provide Allocated File Information

```
     MME   GEFSYE
     ZERO  courtesy call or zero, arglist
     ZERO  43,buffer
```

arglist

```
     ZERO   return, information return
     DEC    -1
```

buffer

```
     BSS   99
```

return

```
     BSS   2
```

on return

On successful completion, bits 18-35 of the first arglist word (information return) have the address, within the 99-word buffer, of information about the mass storage files allocated to the job. The format of this information is:

word 0,	bit 0:	reserved
	bit 1 = 1:	update file with RDERR/JOURNAL/ present
	bit 2:	reserved
	bit 3 = 1:	file with RDERR/DUP/ present
	bit 4 = 1:	file with ACCESS/MONITOR/ present
	bit 5:	reserved
	bit 6:	reserved
	bit 7 = 1:	update file with ABORT/ROLLBACK/ or TEST allocation present
	bit 8 = 1:	update file with VERIFY/YES/ present
	bit 9 = 1:	temporary random files allocated
	bit 10 = 1:	temporary sequential files allocated
	bit 11 = 1:	files with TEST allocation present
	bit 12 = 1:	files with TEST allocation present and update PRMFL's without TEST allocation present
	bit 13 = 1:	user-supplied befores should not be taken for efficiency reasons
	bits 14-29:	reserved
	bits 30-35:	number of mass storage files allocated to this activity

words 1-n, where n = bits 30-35 of word 0

bit 0:	reserved
bit 1 = 1:	file has update allocation with RDERR/JOURNAL/
bit 2:	reserved
bit 3 = 1:	file has RDERR/DUP/ protection
bit 4 = 1:	file has ACCESS/MONITOR/ protection
bit 5:	reserved
bit 6:	reserved

bit 7 = 1: file has update allocation with ABORT/ROLLBACK/
bit 8 = 1: file has update allocation with VERIFY/YES/
bit 9 = 1: file is permanent
bit 10 = 1: file is not cataloged
bit 11 = 1: file has random allocation
bits 12-17: sectors per page (for protected files only)
bits 18-21: allocation type (from PAT pointer)

bit 18	19	20	21	
0	0	0	1	Execute
0	0	1	0	Append
0	1	1	0	Load
0	1	1	1	Recovery
1	0	0	0	Read
1	0	0	1	Read, concurrent
1	0	1	0	Read and append
1	0	1	1	Query
1	1	0	0	Write
1	1	0	1	Write, concurrent
1	1	1	0	Test
1	1	1	1	Test, concurrent

bits 22-23: reserved
bits 24-35: file code

No error returns are made from this function.

User Supplied Befores

```
          MME   GEFSYE
          ZERO  courtesy call or zero, arglist
          ZERO  44,buffer
```

arglist

```
     VFD   18/return,6/0,12H/2-character file code
     DEC   relative block address of start of page
     ZERO  DCW list, zero
     DEC   -1
```

NOTE: File code must reference a file with ABORT/ROLLBACK/ or RDERR/JOURNAL/ protection.

buffer

```
     BSS   99
```

return

```
     BSS   2
```

DCW list

```
     IOTD   data address, number of words
```

(IOTPs,IONTPs, and IOTDs can be used in the DCW list. TDCWs are not permitted.)

on return

On successful completion of this function, the data described by the DCW list will have been saved as a before image for the specified page of the file. Only one page can be described in the DCW list on a single GEFSYE request. This function can reduce the number of I/O required to update a file with ABORT/ROLLBACK/ protection, as FMS will not need to re-read the page before a write is made.

On unsuccessful completion, the following error completions can occur, in addition to the common error returns for protection MMEs:

02 I/O ERROR ON DEVICE XXX SA = NNN......NNN
An unrecoverable I/O error occurred.

17 SEEK ERROR ON DEVICE XXX SA = NNN......NNN
Error return on a seek calculation - collection file.

64 DEALLOCATION PAGE REQUEST
One of the pages cannot be reserved. (Returned only if file has ACCESS/MONITOR/.)

65 PAGE CURRENTLY BUSY
One of the pages cannot be reserved now. Repeat request later. (Returned only if file has ACCESS/MONITOR/ protection.)

71 ILLEGAL DCW SPECIFIED
An illegal DCW string was specified.

72 IMPROPER PROTECTION OPTION
Improper file protection option. (File has neither ABORT/ROLLBACK/ nor RDERR/JOURNAL/ option.)

73 INVALID ARG LIST PARAMETER NO. 000
PAT for file code not found on file does not have write allocation.

When an I-D-S/I file with ABORT/ROLLBACK/ protection is updated, this GEFSYE should be used to collect before images if bit 13 of word 0 in the return from function code 43 is 0 at the time of the I-D-S/I OPEN. If bit 13 is-1, less I/O will be required if FMS issues the re-read of the page than if the before image is furnished with the GEFSYE.

File Query For Allocated File

```
     MME     GEFSYE
     ZERO    courtesy call, arglist
     ZERO    46, buffer
```

buffer

```
     BSS    355
```

arglist

```
     ZERO   status, filcd
```

status

```
     BSS    2
```

filcd

```
     filcd VFD   24/X,H12/fc

                 X = any information, ignored by file system

                 fc = file code (two BCD characters)
```

or, if time sharing (.PNTSS)

```
     filcd ARG aft

            aft = relative address, in SSA, of files PAT pointer
```

on return

On unsuccessful completion, in addition to errors common to protection MMEs, the following may occur:

05 INCORRECT CAT/FILE DESCRIPTION AT AAA......AAA
Not mass storage PAT or not perm file.

42 INVALID FILE CODE OR PAT PTR
File code not found in SSA.

43 INVALID CATALOG BLOCK ADDRESS
File control block error

51 CHECKSUM ERROR - DEVICE XXX SA = NNN......NNN

NOTE: No error message text is returned, thereby leaving buffer information intact. Error code returned in bits 1-11 of status word 0.

information return

buffer +	35		PAT pointer image
	36	0-17	Number words in PAT
		18-35	Reserved
	37 . . . 98		Image of PAT
	99 . . . 162		Image of file description (FMS type 02 block) if a perm file
	163 . . . 226		Image of PIT if protected perm file allocation
	227 . . . 290		Image of FIT if protected perm file allocation
	291 . . . 354		Reserved

Notes:

1. If error code 05 is returned, the PAT image is available if the file is on mass storage. Word 36 of the buffer is nonzero in this case.

2. The buffer is not reset either before or after processing the function. The user may therefore wish to initialize the area prior to issuing the MME GEFSYE.

3. The password in the file description block is set to zero.

4. The PIT and FIT information is present only if the current allocation requires protection; e.g., QUERY access does not use protection services. Bit 0 of word 2 of the PAT image indicates whether or not protection is present.

5. Since formats are subject to change with new software releases, it is suggested that the standard FMS symbol reference be used to access data in the FD, PIT, and FIT.

6. Formats are as described in the System Tables manual.

7. If this function is used by a privileged slave, the buffer area should not be changed until after the function is complete (bit 0 of status word 0=1).

APPENDIX A

DESCRIPTIONS OF REPORTS PRODUCED BY FILSYS ACTIVITY

FILE OR CATALOG LIST

Heading Line

1. user number (MASLST only): first user on page is identified by five-digit number composed of two-digit section number to which the user name hashes and three-digit ordinal of users within that section

2. user name (MASLST only)

3. directive (both MASLST and LIST): card number and directive that produced this listing

4. page number (both MASLST and LIST): page number of previous page, if any, on which is described a file or catalog that is abort, restore, or security locked

Main Line

1. name: file or catalog name

2. password: shown only if listing is by job that has $ PRIVITY and does not request passwords to be hidden; legend /time/ used to indicate timed passwords specified

3. general permissions: abbreviations for permissions, F abbreviates recovery

4. device type, name or pack serial number: for catalog, device or pack holding the catalog, for file, first device file is assigned to (file description is always on same device or pack as catalog it is subordinate to)

5. creation date: date when file catalog was created, in format mmddyy

6. date of last modification: date when file description or catalog was most recently modified (on creation, modify date is same as creation date), in format mmddyy

7. date of last allocation: for file only, date when file was most recently allocated (on creation, allocation date is same as creation date), in format mmddyy

8. date of last change: for file only, date when file content was most recently changed (on creation, change date is same as creation date), in format mmddyy

9. time of last change: for file only, time of completion of most recent allocation during which at least one writer to the file occurred, in format hh.hhh. (test mode write does not count) (on creation, change time is same as time of creation)

10. maximum llinks: specified limit on growth of file, in llinks

11. current llinks: number of llinks currently assigned to file

12. last save volume number: serial number of first volume of most recent FMS save or restore of file; if prefixed by U, the save was a user save, otherwise it was a master save

13. changed since save indicator: N for file content not changed, Y for changed since most recent FMS save of file, W for saved while write busy

14. allocation count: number of allocations to file, modulo 262144. Note that the Peripheral Allocator may allocate and deallocate a file several times before all required resources are available

15. file status: abbreviation to indicate condition of file when it was listed

NULL	-	file not written to since it was created
BUSY	-	file is marked in use by time sharing task(s)
ALOCK	-	file was abort locked when listed
SLOCK	-	file or catalog was security locked when listed
RSTL	-	file description but no file content restored (this occurs when restore is still in execution or restore or save is truncated, space on specified device or device class (if any) is unavailable, tape read errors on restore prevent content records from being recognized, or file is deleted after file description but before file content is saved)

If multiple statuses apply, only the last in the list above is displayed.

16. file type: abbreviation to indicate specified type of file

IDS - file created as I-D-S/I file

SEQ - either random or sequential access to file permitted

RAND - sequential access to file not permitted

TSS - file created by time sharing task

Special Legends in Main Line

1. for file already assigned to nonstructured disk pack or magnetic tape: volume serial number and type and, for magnetic tape, density are displayed

2. for catalog: if a device type or name is specified as a default to assign files to, the type or type and name are displayed

3. on master listing: at beginning of listing for each user, a line is produced preceded and succeeded by a banner of asterisks containing SMC section number, user name, log-on password, TSS resource limit and use, fixed device space limit and use

4. When there are several levels of subordinate catalogs, those immediately subordinate to the most inclusive are called level 1, those subordinate to it are called level 2, and so forth. For each level, the level number and qualified catalog name is provided, followed by list of subordinates. If there are no subordinates, a line with that message is provided.

5. When a catalog is listed that is on a structured removable disk pack, a message is provided indicating that subordinate and detail information for the catalog is not provided. A list directive naming the catalog will provide that information if pack is mounted.

Extra Lines

1. created by: name of user creating file or catalog displayed only when file or catalog is created by user with name different than that of UMC it is subordinate to

2. specific permissions: types of permission and name of user for each specific permission for catalog or file

3. timed passwords: passwords and times between which they are applicable, displayed only when job has $ PRIVITY and does not request passwords to be hidden; times are in format hhmm

4. protection: for files, specifications of file protection different than RDERR/NONE/, ABORT/NONE/, ACCESS/NORMAL/, AUDIT/NONE/. Page size is in words, rounded up to an integral number of 64-word sectors.

5. extents: when file is assigned to more than one device, to a device or device type specified by user, has 15 or more extents or any defective extents, the number of extents on each device is shown by giving device type and name or number followed by parenthesized number of extents and number of llinks in these extents followed by number of defective extents and defective llinks included in total on the device.

6. duplicate extents: same as above except always produced and describes space assigned duplicate of file. Also, if either copy of the file is defective and not being used, the defective copy is noted.

7. monitor extents: same as in item 5 except always produced if ACCESS/MONITOR/ is specified and space assigned to the monitor file is described. The monitor file has one llink for each 1550 pages in the monitored file, and it is used to keep track of which concurrently accessing jobs have accessed which pages of the file.

8. I-D-S/I attributes: specified or assumed attributes for I-D-S/I file. Page size is in words.

SYSTEM MASTER CATALOG LIST

1. user number: 5-digit number to identify user, composed of 2-digit section number to which user name hashes and 3-digit ordinal of users within that section

2. name: name of user for which SMC entry is prepared

3. password: log-on password, unless passwords are requested to be hidden

4. TSS authorizations: given as Y for yes and N for no for CARD, TALK, LODX, and LODS in that order. Urgency is given as blank or 2-digit number between TALK and LODX.

5. llink limit: maximum llinks the user can have assigned to files and catalogs cataloged on fixed devices under this entry

6. TSS use limit: maximum dollars of TSS usage user may spend

7. dollars used: dollars of TSS usage spent at time of listing

8. catalog llinks: space assigned to catalog cataloged under the SMC entry on fixed devices, at time of last restore of these catalogs

9. file llinks: space currently assigned to files cataloged under the SMC entry on fixed devices

10. change date: date in mmddyy format of most recent change in file llink assignment

11. llink days: space x time change for user at time of listing

12. catalog device: name of device or serial number of pack containing all catalogs for entry; if catalogs are duplicated, name of device containing duplicates of all catalogs for entry

13. function: if the line is produced as a result of system or pack master catalog directives, the type of directive is identified: CRE for SMC create, for example, or PACK DEL for PMC delete.

14. busy: If the SMC entry for a user is marked as being modified by time sharing and/or by FILSYS, the legends TB and/or BB appear.

SYSTEM MASTER CATALOG SUMMARY

1. section: SMC section number that can be referenced on SAVEMAST

2. number of users: number of SMC entries that hash to this section number

3. llinks: llinks currently assigned to files and catalogs on fixed devices for entries hashing to this section number; llinks for duplication are not included

4. llinks accounted for: sum of llinks used for both files and catalogs for entries hashing to this section; llinks required for duplicate files and/or catalogs are included; if sum for a user exceeds 262,143 llinks, the excess is not included

DEVICE UTILIZATION REPORT

1. name: name given at startup to device, (in case device is a disk pack, to the spindle)

2. type: device type

3. address: IOM number, channel number, unit number as entered on configuration cards at Startup

4. class: classes of devices are distinguished

FIXED	not removable
RMV-S	removable, structured
RMV-N	removable, nonstructured
STRNG	removable, stranger
RELEASED	operator has released
DITTO	another name of device described above
SHARE	shared between systems
UNSHR	not shared between systems

5. pack number: serial number of pack mounted on spindle

6. llink capacity: capacity of device in llinks (not including alternate tracks)

7. available for temporary use: capacity of device less llinks marked defective or assigned permanently

8. available for permanent use: capacity of device less llinks marked defective, assigned permanently, or reserved for growth

9. permanent use: llinks assigned permanently (to cataloged files, catalogs, label, directories, uncataloged but permanent files)

10. marked defective: llinks withdrawn from assignment because suspected of being defective at startup, RSIP, or by operator selecting W option on read error

11. available for growth: llinks reserved for growth of permanent files

12. llinks per allocation unit: unit by which space is assigned on device as specified on configuration cards at startup

13. connects: connects issued to device divided by 1024, since startup

14. errors: errors encountered on device since startup

15. used catalog: llinks used for catalogs on device or pack. Includes system or Pack Master Catalogs, catalogs for users, and duplicates of these, if any. Note that these llinks are included in the USED PERM llinks.

16. totals: totals are shown in links (llinks divided by 12) except for number of devices, defective llinks, connects, errors, and catalog llinks.

FILE ROLLFORWARD

Heading Line

1. indicates a rollforward activity

2. page number of previous page containing last diagnostic message; if no diagnostics have occurred, the word NONE appears

3. subtitle lines indicating the output format for each activity on the journal file

Main Line

1. sequential numbers associated with each activity on the journal file to be listed

2. qualified file name, without passwords, if it exists in the system; if not, a message indicating file cannot be located appears

3. $ SNUMB and activity number of the job that wrote to the file

4. date and time of day of last change to the file; note that this is same date and time that appears in the listing of the file

5. number of pages updated, including duplicate file and pages on which attention alerts occur; if no pages are updated, the word NONE appears

6. message indicating results of file update:

   a. COMPLETE
      file has been successfully updated

   b. SKIPPED****
      file was not updated and reason is given (see Appendix B)

   c. INCOMPLETE
      file has been partially updated and reason is given (see Appendix B)

Summary Report

1. lists total number of update activities completed

2. lists total number of update activities skipped

3. lists total number of incomplete update activities

Defective File Summary Report

This report appears only when file(s) in defective status is updated; indicates status of file(s) at completion of reading a journal file.

1 FILE n REMOVED FROM DEFECTIVE STATUS

2 FILE n TO REMAIN IN DEFECTIVE STATUS

where n is sequential number listed in main line.

FILE RECOVERY PROGRAM

Heading Line

MAIN PROGRAM

1. description of current system and directive, date, time of day, and page number 1 (all directives)
2. indicates type of activity, date, time of day, and page number 2 (for all directives except PATCH)
3. termination notes (for all directives except PATCH)
4. When recovery is completed or no more tables are available for files to be recovered, the recovery files (same as files obtained) and respective catalogs are listed and deleted (for all directives except PATCH and RECDUMP). If recovery is not complete, that is, tables are available for files to be recovered, the program indicates that no files are deleted and another attempt to recover the files should be made.

OVERLAY PROGRAM

1. indicates recovery program as follows:

   a. recovery activity, GCOS number, date, time of day, and page number (note that the GCOS number for initial recovery is always FSREC 00)

   b. subtitles for output

2. indicates for RECDUMP directive:

   dump activity, GCOS number, date, time of day, and page number

3. indicates for REMOVE directive:

   remove activity, GCOS number date, time of day, and page number

4. displays absolute sector number for PATCH directive

MAIN PROGRAM

1. indicates system number on which job is executing, current system boot sequence number, and directive type (all directives)

2. contents of GCOS3/RCVYINFO (for all directives except PATCH) as listed:

   (CHECKSUM ERROR appears when program execution terminates because of checksum error.)

   a. system number

   b. system boot sequence number

   c. recovery type needed or executing, including current job

   d. devices initialized and restored

   e. when only the current sequence number is present, the following message appears:

      NO PRIOR BOOT IN RCVYINFO FILE

   f. recovery files obtained

OVERLAY PROGRAM

1. indicates GCOS number, program number, qualified catalog of file name of protected file, file code, number of pages recovered, and file status (initial recovery)

   The number of pages recovered includes duplicate file pages and pages on which attention alert occurs. If no pages are recovered, the word NONE appears and if 1000 or more pages are recovered > 999 appears. File status code may be any of the following:

   a. COMPLETED
      file was successfully recovered for this job.

   b. SKIPPED
      file was not fully recovered as indicated. File cannot be recovered by recovery program.

   c. INCOMPLETE
      file was partially recovered as indicated.

   d. DEFERRED
      file recovery was deferred as indicated. Cause of deferral should be corrected before attempting recovery again.

2. RECDUMP protection tables (see System Tables manual for formats):

   a. dump of the contents of file GCOS3/RCVYINFO

   b. for each GCOS number, activity number, and program number of files requiring recovery, the catalog or file name and protection options are listed.

      protection table addresses are listed in relative and absolute mode for cross reference with the protection tables.

   c. dump of the recovery protection tables file with relative and absolute addresses for cross reference with general information of heading line; if this file is not present the following message appears:

      *** RPTBLS DOES NOT EXIST ***

      dump of the alternate recovery protection tables file identified by the GCOS number; if this file is not present, the following message appears:

      *** ALTBLS DOES NOT EXIST ***

      if none of the recovery files is present, the following message appears:

      *** THERE WEREN'T ANY PROTECTION TABLES ***

      if no programs require recovery, the following message appears:

      *** THERE WEREN'T ANY JOB(S) TO RECDUMP ***

   d. GCOS number and file names precede the dump of the collection file to which they pertain. If the collection file is empty, the following message appears:

      *** COLLECTION FILE EMPTY ***

      otherwise, the collection file header is dumped with the page header and page images.

3. indicates for REMOVE directive:

   GCOS number, program number, activity number, and catalog or file name for each set of tables that are removed

   if no jobs are present in the protection tables the following message appears:

   *** THERE WEREN'T ANY JOB(S) TO REMOVE ***

4. indicates for PATCH directive:

   a. device name or disk pack serial number

   b. word FROM

   c. dump of the sector with upward arrows pointing to the words to be changed

   d. word TO

   e. dump of the sector with upward arrows pointing to the words to be changed

APPENDIX B

ERROR MESSAGES

ABNORMAL TERMINATION

FILSYS terminates with an FS abort if errors are encountered in processing directives.

Allocation requests for files SR, SS, and FD result in a FC abort unless the following conditions are met:

1. Request the file by means of a $ FILE control card.
2. Assign an R (release) disposition to the file.

CATEGORIES OF ERROR MESSAGES

Error messages contained in the appendix are categorized into the following groups:

1. General error messages.
2. Error messages produced by save directives.
3. Error messages produced by restore directives.
4. Error messages produced by rollforward directive.
5. Error messages produced by recovery directives.

NOTE: The Time Sharing System also produces error messages relating to the use of FMS. Refer to the _TSS General Information_ manual for information pertaining to these error messages.

GENERAL ERROR MESSAGES

ACCESS GRANTED TO I-D-S/I FILE

Request to allocate an I-D-S/I file was successful and Peripheral Allocator was notified to restrict allocation of I-D-S/I files in order to establish a cleanpoint.

ALL MUST BE ON SAME PACK OR NONE MAY BE

If a catalog (Type 0) or file description (Type 2) is on a pack, all of the subordinate catalog structure must be on the same pack.

BAD SPACE INVENTORY ON DEVICE (name)

Space on named device cannot be withdrawn from or returned to permanent availability because the inventory of available space recorded on the device is unreadable or appears damaged.

CHARACTER STRING SIZE ERROR

The name field is larger than 12 characters.

CATALOG/FILE SECURITY LOCKED

Request denied because catalog or file is security locked and request is from user who is not the creator of catalog/file or does not have lock permission.

CATALOGS BUSY

Message used as a special time sharing indicator. When an FMS request is made and the required gate is already closed, this message is sent to time sharing instead of a GEWAKE.

CHECKSUM ERROR-DEVICE (name) SA = (address)

Request cannot be satisfied because sector read in attempting to get a catalog or file description has incorrect checksum. The sector read is identified by naming the device and supplying the device address. The seek address is given as a word location of the sector in a llink in the first six octal digits and starting sector number of the llink in the second six octal digits. Reference to the sector may be in error or the sector itself may be damaged.

COLLECTION FILE ERROR

Writing to a protected file cannot be completed, because a read or write error was detected on the collection file.

DEADLOCK ON PAGE REQUEST

Request to reserve pages cannot be satisfied and deadlock occurs, because one of the pages is in concurrent use by a program whose completion is pending on the completion of the requesting program.

DESCRIPTION TOO LONG

On a FILSYS directive, a qualified catalog or file description appears to contain more than 50 names. Or on SAVE or RESTORE directives, the specification includes more names (of users, catalogs, files, devices, etc.) than space permits. Or on a listing, more than 100 catalog names need be remembered.

DEVICE (name) RELEASED

Request cannot be granted because catalog or file description, or file, is on or is to be placed on the released device.

DIRECTIVE NOT IMPLEMENTED

A FILSYS directive is recognized but is rejected because implementation has not been provided for with this software release.

DUP CANNOT BE ON RMVBL DEVICE

Duplication cannot be specified for a file assigned to a removable structured disk pack, to a nonstructured disk pack, or to a magnetic tape.

ERR TDS SUBSET PAGES RELEASED

Request cannot be granted, because list of pages to be released is incorrect.

EXPECTING A DIRECTIVE

First word on input card to FILSYS after finishing with previous directive (or on first input card) is not recognized as a directive. The previous directive may have been incorrectly written.

EXPECTING AN IDENTIFIER

On a FILSYS directive, an identifier is required but is not given (password or name of catalog, file, user, device, etc.).

EXPECTING AN INTEGER

On a FILSYS directive, an integer value is required to complete a specification, but none is provided.

EXPECTING AN OPTION

On a FILSYS directive, an option is required, but none is given.

FAILURE IN NAME SCAN

Symptomatic of FMS implementation error.

FILE BUSY; TRY LATER

Request to allocate file refused because of present allocation(s). Or request to modify protection specified for file is refused until file is not allocated. Or file content has not yet been restored.

FILE IN DEFECTIVE STATUS

Request to allocate file is denied because requested allocation does not accept file in defective status nor is protection specified for the file that would detect and handle attempts to access defective parts of file. Allocation to a defective file is granted only to QUERY and RECOVERY requests when protection for the file is not specified. A REPLACE directive, using the latest save tape, can be used to remove defective file marks. If the file has journal specified, an additional ROLLFWD directive may be required to remove the defective file marks if the file has changed since the save tape was created.

FILE IS ABORT LOCKED

Request to allocate file is denied because requested allocation does not accept file in abort lock condition. Only QUERY and RECOVERY allocation are granted to an abort locked file.

FILE IS TDS LOCKED

A monitor controlled file was being used by a TDS user when TDS aborted or the system failed. The lock remains in effect until the TDS and FMS journal synchronization. Note that file recovery is not possible on a file locked in this manner.

FILE NOT BEING MONITORED

Request to reserve pages was denied, because file does not have monitoring protection or is not allocated to the program.

FILE NOT CAT NAMED OR CONVERSE

On a FILSYS directive, a file name is given where a catalog name is required or vice versa.

FILE NOT DUPLICATED

Request to perform duplicate file processing is denied, because file is not duplicated or is not allocated to the program.

FILE "PD" SIZE EXCEEDED

When an error occurs while writing to a temporary file on a MASLST directive, the data has exceeded the file size.

FILE RESTORE LOCKED

Request to allocate file is denied; file is in restore lock condition.

FILE WRITE IN BATCH ONLY

A request to allocate file for writing is denied because protection against incomplete update is specified for file and such protection is not implemented for time sharing use.

$FSYS HAS BEEN ENABLED

A request for allocation to a protected file was denied, because the FMS Executive was not in memory. The FMS Executive is invoked and a message is issued to indicate that a GEWAKE is required.

ILLEGAL DCW SPECIFIED

The DCW word specified does not have a valid data address or the required action code type, or the word count exceeds the allowable limit.

ILLEGAL OPTIONS COMBINATION

Contradictory options have been selected; e.g., specification of both ABORT/ROLLBACK/ and ABORT/DELAY/ or same device specified for both original and duplicate files.

ILLEGAL PACK TYPE

A pack identification has unrecognized pack type, or the pack referenced as removable structured is labeled as nonstructured.

ILLEGAL SUBFUNCTION CODE

On a GEFSYE function code 30, the subfunction code is incorrect or not available to the user.

ILLEGAL USAGE OF OPTION

On a FILSYS directive, an option appears that is illegal for this directive. Or an inconsistent or invalid combination of options is used.

IMPROPER PROTECTION OPTION

The file does not have protected allocation or does not have the proper type of protection.

INCORRECT CAT/FILE DESCRIPTION AT (name)

The qualified catalog or file name does not match an existing name. The first non-matching name is given. Or the name of a catalog is given when a file name is required or vice versa.

INCORRECT OR MISSING PASSWORD AT (name)

Request denied because request gives a password for the named catalog or file where catalog or file has specified another password or no password at all; or request denied because password is specified and none has been supplied.

INVALID ARG LIST PARAMETER NO. (octal number)

The nth parameter of the user's MME GEFSYE calling sequence is invalid.

INVALID CATALOG BLOCK ADDRESS

Request identifying file by file code cannot be granted because PAT for file does not contain address of file description.

INVALID CHARACTER IN STRING

Alpha or numeric character expected but not supplied.

INVALID FILE CODE OR PAT POINTER

Request for file identifying file by file code or PAT pointer location cannot be granted because PAT pointer is invalid or cannot be located.

INVALID INTEGER VALUE

On a FILSYS directive, an integer appears that is larger than limit specified.

INVALID MME PARAMETER

Part of the MME GEFSYE calling sequence is invalid; that is, the address overlaps, the argument list is in the buffer, or there is an invalid character in the catalog or file name.

INVALID OPTION

On a FILSYS directive, prior punctuation indicates an option follows, but word encountered is not recognized as an option.

INVALID SPACE IDENTIFIER

Request to replace defective space cannot be granted because space is not properly described, not assigned to this file, or not shown to be defective.

I/O ERROR ON DEVICE (name) SA = (address)

Request cannot be satisfied because a required catalog or file description cannot be read from or written to the named device. The seek address is given as a word location of the sector in a llink in the first six octal digits and starting sector number of the llink in the second six octal digits.

LAST DIRECTIVE NOT FINISHED

A file is required on a FILSYS directive, but the system cannot comply.

LOG-ON PASSWORD INCORRECT

On a USERID directive, the log-on password given is not the correct password for the named user.

MASTER CATALOG ERROR

On a MASLST directive, no users have been found. On a SAVEMAST directive, a given user has not been found.

NAMED DEVICE NOT CONFIGURED

On a SAVE or ROLLFWD directive, a name is given as a device name, but that name does not appear in the list of dedicated mass storage devices.

NEED MORE PAT SPACE

Request to allocate file denied because file duplication is specified and space provided for FMS to construct a PAT is insufficient for both original and duplicate file descriptors.

NO SPACE FOR CATALOG ON DEVICE

Catalogs for user have been started on one device and are not allowed to continue to another. Deleting a catalog or a file assigned space on the initial device should make space available. This error also occurs if the PACK List option is attempted on a pack that has no Pack Master Catalog space.

NO SPACE FOR FILE ON DEVICE (name, type, of ANY TYPE, or PK)

A file cannot be created because sufficient space is unavailable on named device, on devices of given type, on any device, or in pack holding catalog to which file is subordinate, or an attempt has been made to use a file that is restore locked, in which case the name appears in the message.

NO OR INVALID USERID

On a FILSYS directive, the user name given on the $ USERID control card or on the most recent USERID directive does not match user name required. Or no $ USERID card or USERID directive is present

NO PROTECTION SPACE AVAILABLE

An allocation request for a file specifying protection cannot be granted because space for control tables is not available in the FMS Executive. When other protected file allocations finish, space should be available.

NONSTRUCTURED FILE ENTRY

This message is sent to Peripheral Allocator to notify the Allocator that a nonstructured disk pack or magnetic tape is required to allocate a file.

NON-UNIQUE NAME

A request to create a catalog or file cannot be granted because the qualified catalog or file name is already in use.

OPTION NOT IMPLEMENTED

On a FILSYS directive, an option is recognized but is rejected because implementation has not been provided for in this software release.

PACK MASTER CATALOG ERROR

On a MASLST directive, where device pack option is used, no user is found on the structured pack.

PAGE CURRENTLY BUSY

Request to reserve pages was denied, because the pages are currently being operated upon by another program.

PERMISSIONS DENIED

Request cannot be granted because the request is from a user other than the creator, or a user excluded by specific permissions, or the catalog or file does not have general permissions and the user lacks specific permissions, or an attempt has been made to read from a null file.

POSTING IN PROCESS ON THIS FILE

A MME GEFSYE to indicate changes complete or to cancel changes made to a file has been issued before a prior MME GEFSYE has finished. The MME GEFSYE should not be repeated until the status return word shows that the previous MME GEFSYE has finished.

PRIVILEGED DIRECTIVE

A FILSYS directive that requires that a $ PRIVITY control card appear in the job has been encountered in a job that does not contain the control card.

PUNCTUATION MISSING OR INVALID

FILSYS directive not properly punctuated.

READ ERROR - FILE CODE "PD"

On a MASLST directive, an error has occurred in reading the temporary file.

REQUESTED ENTRY NOT ON-LINE

Request cannot be granted because a removable structured disk pack is required that is not presently mounted.

*** RESERVED FOR FMS ***

Message error code reserved for future use.

TDS MON ALLOC ERROR

Request was denied for allocation of a monitored file. The file is either being used by another TDS user or no TDS initialization GEFSYE was issued.

SEEK ERROR ON DEVICE (name) SA = (address)

Request cannot be granted because a required catalog or file description cannot be read from or written to the named device. The seek address is given as a word location of the sector in a llink in the first six octal digits and starting sector number of the llink in the second six octal digits. Reference to the sector has been damaged.

SIZE REQUEST LS THAN ALLOCATED

A request to modify the maximum size of a file cannot be granted because the newly specified maximum size is less than space already assigned to the file.

SPACE REQUEST GREATER THAN ALLOWED

A request to create a file cannot be granted because the total space assigned to all files cataloged under the user name would then exceed space allowed that user on all fixed devices or on the disk pack.

STATEMENT INCOMPLETE

On a FILSYS directive, a specification has not been completed.

SYSTEM JOURNAL NOT CONFIGURED

The site has chosen not to allow allocation of files with the RDERR/JOURNAL/ option.

TROUBLE GETTING TEMP FILE

To process SAVE, RESTORE, and ROLLFWD directives, temporary mass storage is required to save main FILSYS program or to extend calling sequence for SAVE, and none is available.

TOO MANY NAMES

On SAVE or RESTORE directives, names given exceed 200 in number. On ROLLFWD directive, names given exceed 100 in number.

UMC CANNOT BE RENAMED

A CMOD directive specifies a new name for the UMC. Because the UMC name must always be the same as the user name, the specification is in error.

*** UNACCOUNTABLE ERROR ***

An error occurred, because a collection file device name or statistical collection table was not defined.

```
                  TYPE   type
UNDEFINED DEVICE  NAME   device
```

Completion of a requested function is not possible, because the specified device type or name is non-existent. If message occurs on a delete request (purge or release), two more file purge requests are needed to remove the entry.

UNRECOVERABLE FILE - FC "PD"

On a MASLST directive, an unrecoverable I/O error has occurred.

USER-ID NOT IN MASTER CATALOG

An entry in the SMC is not found for the first name in the specified catalog or file descriptor on the request. This first name should be that of the user under which the catalog or file sought is cataloged.

VALIDATION ERROR x...x SHOULD BE y...y

This message appears if the data pattern in the PATCH DEVICE directive (x...x) does not correspond to the content of the disk location being changed. No changes are made to the disk via the directive once this message appears.

WRITE ERROR - FILE CODE "PD"

An unrecoverable I/O error occurs on a MASLST directive when attempting to write MASLST data to the temporary PD file.

(name) DEVICE NAME INCORRECT

The name must be a three-character, acceptable device name.

(name) NAME NOT IN PMC

An entry in the Pack Master Catalog is not found for the first name in the specified catalog or file descriptor on the request.

nnnnnn CATALOG/FILE-CATALOG POINTERS FOUND

nnnnnn is the number of pointers to catalogs or files encountered before recovery is performed.

nnnnnn CATALOGS/FILE-CATALOGS SAVED

nnnnnn is the number of catalogs or files encountered before recovery is performed.

CHECKSUM ERROR IN SMC LLINK.
CLASS nn CIT?

The installation is given the option to continue, ignore, or terminate the error. If continue, this class is bypassed and save continues with next class. If ignore, save continues as if error never occurred. However, the ignore can cause further problems, depending upon condition of SMC llink.

CONTENT SAVE ERROR ON FILE x...x
PROCESSING TERMINATED ON THIS FD.

I/O error occurred when FD was read from disk. Save continues with next file in candidate list.

*** END STRUCTURE RECOVERY ***

Self-explanatory.

*** ERROR READING JOURNAL (J1) FILE ***

An error occurred in trying to read the journal control file. Save continues.

FILE x...x READ ERROR. DEV/xxx.
LLINK# n...n THRU n...n SKIPPED.
SAVE CONTINUED

I/O error occurred when one link or less file content was read from disk. Whatever is in buffer after the read is written to save media. Questionable llinks are indicated in message (these are llink number, not sector addresses). Save of contents is continued.

FOLLOWING LLINK ssssss POINTED TO BY ADDRESS IN LLINK pppppp RECOVERED

ssssss is the sector number of recovered llinks and pppppp is the sector number of a llink containing a pointer to a recovered llink.

*** INTERNAL INTERFACE ERROR - CODE n ***

User program bug. Save terminated.

*** LLINK ssssss HAS FOLLOWING INVALID CATALOG ***

ssssss is the sector number of a llink that contains an invalid sector. Llink is snapped.

nnnnnn LLINKS WERE NOT RECOVERED

nnnnnn is the number of llinks that could not be recovered because of a read error or a checksum error.

nnnnnn LLINKS WERE RECOVERED THIS USER

Self-explanatory.

*** MORE MEMORY REQUIRED TO PERFORM POSSIBLE RECOVERY ***

The allotted memory must be increased on the limits card to provide for buffer space of up to 20,480 catalogs per user.

*** NO PRIMARY SAVE DEVICE FURNISHED ***

$ TAPE PS,... card missing. Save terminated. Insert card and rerun save.

*** PACK x...x UNDEFINED or RELEASE ***

Savemast from removable pack is terminated.

*** PRIMARY/COPY SAVE HDR WRITE ERROR ***

An unrecoverable I/O error occurred when a write to task block to tape was tried. Save is terminated.

PRIOR LLINK POINTER = ssssss

ssssss is the sector number of a llink pointing to a recovered llink. This message is followed by names of catalogs or files found in a recovered llink and a snap of the recovered llink if a NOP is executed in the module SRSR with the location defined in the symdef SNAP used as the NOP operand.

READ ERROR ON FD x...x CONTINUATION BLOCK.
PROCESS ON THIS FD TERMINATED

Save continues with next file (FD) in candidate list.

{READ
CHECKSUM} ERROR IN LLINK ssssss POINTED TO BY ADDRESS IN LLINK pppppp .LLINK SKIPPED

Self-explanatory.

READ ERROR ON FD x...x.
PROCESSING TERMINATED ON THIS FD

Save continues with next file (FD) in candidate list.

SAVE LIST ENTRY NOT FD.
SKIPPED = ssssssssssss/nnnnnnnnnnnn

An FD read from candidate list was not correct type. Save continues with next file in candidate list.

```
ssssssssssss = bits 0-5, class number of FD
             = bits 6-17, sector offset within llink
             = bits 18-35, sector address of llink in which FD resides
nnnnnnnnnnnn = bits 0-17, SCT address of device
             = bits 18-35, device name on which FD resides
```

*SV-S#sssss-SAVE LIST ENTRY=m...m,n...n:CKSM CIT?

A checksum error occurred in a user's file descriptor

```
mmmmmmmmmmmm  = bits 0-5, class number of FD
              = bits 6-17, sector offset within llink
              = bits 18-35, sector address of llink in which FD resides
nnnnnnnnnnnn  = bits 0-17, SCT address of device
              = bits 18-35, device name on which FD resides
```

Installation has option to continue, ignore, or terminate. If continue, save skips to next file. If ignore, an attempt is made to continue the save of the entry causing the error; however, further problems may rise.

SAVE TERMINATED, PRI/DUP SMC
DEVICE NOT FOUND

Self-explanatory.

SAVE TERMINATED, QUERY SMC FAILURE

Self-explanatory.

SAVE TERMINATED. NOT "RMVBL" PACK x...x

Pack x...x is not on a removable spindle.

SAVE TERMINATED. "RMVBL" PACK x...x
IS RELEASED

Self-explanatory.

SAVE TERMINATED. WRITE ERROR ON SAVE MEDIA

Self-explanatory.

SAVE TERMINATED. UNDEFINED "RMVBL"
PACK NAME x...x

Self-explanatory.

SAVE TERMINATED. UNSUCCESSFUL UMC
READ

Self-explanatory.

```
*** START STRUCTURE RECOVERY USER=userid DEVICE=ddd  BLOCK#=bbb ***
```

ddd is the name of the device on which the structure resides and bbb is the block sequence number of the recovered structure on the save tape.

```
WRITE-BACK ERROR ON FD BLOCK x...x
PROCESSING CONTINUED
```

Error occurred after FD had been updated at end of contents save.

UMC x...x LLINK CHECKSUM ERROR IT?

Installation has option to ignore error or terminate. If ignore, save program continues as if error never occurred. However, for ignore, further errors may occur depending upon the condition of the UMC llink.

UMC x...x LLINK CHAIN CHECKSUM ERROR CIT?

Checksum error occurred in one of user's catalog llinks. Installation has option to continue, ignore, or terminate. If continue, save skips to next user. If ignore, save contines as if error never occurred. However, for ignore, further problems may arise, depending on condition of catalog llink.

UMC x...x LLINK CHAIN READ ERROR CT?

I/O error occurred when a read to a catalog llink was tried. Installation has option to continue or terminate. If continue, save continues with next user.

UNSUCCESSFUL SMC LLINK READ.
CLASS nn CT?

I/O error occurred in reading an SMC llink. Installation has option to continue or terminate. If continue, class is bypassed and save continues with next class.

UPDATE-READ ERROR on FD BLOCK x...x.
PROCESSING CONTINUED

Contents have been saved but catalog is not updated.

ERROR MESSAGES PRODUCED BY RESTORE DIRECTIVES

(A)BORT, (C)ONTINUE

This message is always preceded by a specific error message and gives the installation the option to abort or continue the restore.

(A)BORT, (C)ONTINUE, COPY #

This message is always preceded by a specific error message and gives the installation the option to abort, continue, or enter the copy tape number if one exists.

(A)BORT, (C)ONTINUE, (I)GNORE AND CONTINUE

This message is always preceded by a specific error message and gives the installation the option to abort, continue, or ignore the error and continue the restore. If continue, the erroneous record is bypassed. If ignore, restore continues as if the error never occurred.

The ignore option should be used with care as it may cause further errors; e.g., a record with a checksum error can possibly be processed, but it can also precipitate a condition wherein it is impossible to proceed.

*** ATTEMPTED TO OPEN SECTION THAT WAS NOT SHUT ***

Self-explanatory. Restore is terminated.

ATTEMPTING TO RESTORE CATALOG TO FILE x...x

A type 0 catalog on the save tape was a type 2 (FD) catalog in the system. Restore is terminated.

ATTEMPTING TO RESTORE FILE TO CATALOG x...x

A type 2 (FD) catalog on the save tape was a type 0 in the system. Restore is terminated.

*** BELOW FILE ALREADY RESTORED OR CAT NOT RESTORED ***

A restoremast from a non-ST1 device save encountered a multidevice file that has already been restored from another device save or the catalogs have not been restored at all (have not been restored from the ST1 device save). If the latter is the case, it is an error and the restores must be rerun, starting with the ST1 device save tape.

*** BELOW FILE NOT RESTORED. NOT ENOUGH ON TAPE ***

A device restore was attempted on a given multidevice file, but there was not enough data on the save tape. The entire file must be restored again. Restore continues with the next file.

*** CANNOT RESTORE BOTH SMC DEVICES (xxx,xxx) ***

Both SMC devices have been specified in a device restore; illegal. Abort.

*** CAN'T RESTORE. NO CATALOGS EXIST ***

A user restore was attempted from a non-ST1 device save, but failed because no catalogs exist for the user. Prior to this restore, the user's catalog must be restored from the ST1 device save. Abort.

CAN'T RESTORE ANOTHER USER'S FILES

The job userid does not agree with the CAT/FILE string UMC name on the restore directive. Restore is terminated.

CAN'T RESTORE UMC-SAVE WAS PACK # x...x

A restore was attempted to a removable structured pack at the UMC level. However, the UMC existed on a perm device. Restore is terminated. Correct the qualified CAT/FILE name in the restore directive and rerun job.

*** CATALOG COMPRESSION ERROR ***

An illegal type continuation catalog was found on the save tape. Check save tape and/or MASLIST/CLIST reports for discrepancies. The installation is given the option to abort or continue the restore. If continue, restore continues with the next userid. A message is generated identifying the userid for which the error occurred.

CATALOG OR FILE x...x NOT ON SAVE TAPE

The user specified a catalog or file name in the restore directive that is not contained on the save tape. Restore is terminated. Correct the error and rerun job.

*** CATALOG OR FILE x...x NOT RESTORED ***

This message is always preceded by another message explaining why the catalog could not be restored. Restore continues.

*** CATALOG RELOCATION ERROR ***

User program bug. The installation has the option to abort or continue. If continue, a message is generated identifying the user not restored. Restore continues with the next user.

*** CATALOG SPACE TABLES OVERFLOWED ***

If this message is given, the disk packs are fragmented (more than 20 descriptors to map the user catalog structure). The installation has the option to abort or continue. If continue, another message is generated identifying the user not restored. Restore continues with the next user.

*** CHECKSUM ERROR ***

Checksum error in SMC llink read from system. Installation has the option to abort or continue the restore.

*** DENIAL ON GETTING SPACE. BELOW FILE NOT RESTORED

Request to get new spaces in replace was denied. Space is left defective and replace continues with the next file.

*** DESCRIPTOR MIXTURE ***

A device restore was attempted and a file was encountered with meaningless space descriptors. Abort.

*** DEVICE DIRECTORY CHECKSUM ERROR ***

A checksum error occurred on the device directory being read. The installation has the option to abort or continue. If continue, the restore proceeds as if the error never occurred. However, the condition of the device directory is uncertain and further errors are possible.

*** DEVICE NAME xxx NOT FOUND IN SYSTEM ***

This message may be generated for one of two reasons:

When releasing existing catalogs and files at the beginning of the Restoremast, a device name that does not exist in the system was found in a catalog/file string. The file was not restored in a previous restore and therefore the device description contains the device name the file was saved from. Restore continues.

A device name was found in the device directory for which there is no entry in the System Configuration Table. Software bug. Restore is terminated.

*** DEVICE NAME xxx NOT FOUND ***

User program bug. Restore is terminated.

*** DEVICE xxx NOT ONLINE/RELEASED; FILE x...x ***

Specified device not online/released. File was restored. Restore continues with next file.

*** DEVICE xxx NOT ON PERM SPINDLE; FILE x...x ***

A file had a device constraint not on a perm spindle. File is bypassed and restore continues with the next file.

*** DEVICE xxx NOT SHARED. BELOW FILE NOT RESTORED

Restore encountered a file to be restored on a user specified device is not declared shared in a shared system. Restore continues with the next file.

*** DISC WRITE ERROR ***

An unrecoverable I/O error occurred in writing catalog structure to disk. Restore is terminated.

$\left\{\begin{array}{l}\text{DOONLY}\\ \text{DONTDO}\end{array}\right\}$ NAME x...x NOT ON TAPE

The specified catalog or file name was not found on the save file. The user restore terminates. Remove or correct the specified name and rerun job.

*** ERROR DURING SINCE PROCESSING ***

User program bug. Restore is terminated.

*** ERROR IN DESCRIPTOR PROCESSING ***

The Mass Store Allocator denied a request to mark space unavailable on disk. User program bug. Restore is terminated.

*** ERROR IN READING SMC LL FROM DEVICE xxx. RESTORE DISCONTINUED ***

A device restore was attempted on an SMC device and an I/O error occurred when reading the other SMC device. Abort.

*** ERROR SPLITTING DESCRIPTORS FOR ABOVE FILE ***

In getting new space for defective space in replace, too many extents were obtained to fit in catalog. File must be entirely restored; is left defective. Installation has option to continue with the next file.

*** ERROR WHILE CHECKSUMMING RCVYINFO FILE ***

The checksum is incorrect for the recovery information file.

*** ERROR WHILE WRITING DISC ***

An unrecoverable I/O error occurred in writing catalogs or data to disk. Restore is terminated.

*** ERROR WHILE $\left\{\begin{array}{l}\text{WRITING}\\ \text{READING}\end{array}\right\}$ TO RCVYINFO FILE ***

An I/O error occurred either during writing or reading to the recovery information file.

*** ERROR WHILE TRYING TO GEMORE RCVYINFO FILE *** (R)ETRY, (C)ONTINUE ?

Restore could not access the recovery information file, because the file is either busy or non-existent.

*** FD EQUATE LIST ERROR ***

User program bug. The installation is given the option to abort or continue the restore. If continue, another message is given identifying the userid not restored. The next userid is then processed.

*** FILE x...x FD MISSING. CONTENT NOT RESTORED ***

A candidate list catalog was read but was not an FD catalog. While performing the restore, another restore has restored the file or the file has been deleted. Restore continues with next file.

*** FILE x...x BEING DELETED; CANNOT RESTORE

The file cannot be restored and deleted from the system simultaneously. The file is not restored and processing continues with the next field. The file may be restored after delete is completed. If diagnostic persists, delete file again and rerun the restore.

*** FILE x...x IS BUSY; CANNOT RESTORE ***

A file was busy during a user restore. File is bypassed and restore continues with next file.

*** FILE x...x NOT COMPLETELY RESTORED ***

File was not completely restored because of I/O error in reading content from save tape. Restore continues with next file.

FILE x...x NOT ON SAVE TAPE

Save user program bug. Restore is terminated.

*** FILE SIZE ERROR IN FD FOR FILE x...x ***

An FD catalog on the save tape showed zero size file. File was restore locked when saved. Check save report and if available, list report taken at save time. Restore continues with next file.

*** FILE x...x WAS NOT RESTORED. COULD
NOT OBTAIN SPACE PROPERLY ***

There is not room on any device, specified or not, to hold file. Restore continues with next file.

*** FILE x...x WAS TRUNCATED ***

There was more data on the save tape than the maximum size field in the FD allows for. This could happen in the user restore when the structure already exists on disk. The size of the file is truncated to the maximum size allowed. Restore continues with the next file.

*** FILES ARE LEFT RESTORE LOCKED. CHECK MASLST ***

This is an informative message at the completion of content restore. For some reason, all files have not been restored. Check MASLST (taken after restore) to find restore locked files and check restore execution report for reasons why those files were not restored.

*** FIRST ENTRY IN FD EQUATE LIST IS NOT A HEADER ***

User program bug. Restore is terminated.

*** FMS RECOVERY MAY FAIL THIS BOOT ***

The recovery information file is not in order and could cause the recovery program to fail.

*** FOLLOWING FILE NOT RESTORED. READ DISC ERROR (DUP)

When restoring from duplicate file in replace or device restore, an I/O error occurred. File is left restore locked and next file is processed.

*** FOLLOWING FILE NOT RESTORED. TAPE ERROR

A tape read error occurred during replace or device restore. File is left restore locked and the next file is processed.

*** GET CURRENT ERROR ***

Catalogs with bad pointers found on save tape. Restore is terminated. Check save and list reports for discrepancies.

*** GEMORE DENIAL-FILE CODE FD ***

Not enough temporary disk space for file candidate list. Restore is terminated.

*** GEMORE DENIAL-FILE CODE SS ***

Forty links of temporary disk space is not available. Restore is terminated. Try again.

*** GET CURRENT SECTOR ERROR ***

The UMC for a user on the save tape contains names for which no catalogs exist (on the save tape). These names are deleted (with an identifying message) and good catalogs of the user are restored. Check the save report and MASLST/CLIST (if available) taken at save time for discrepancies.

*** ILLEGAL CATALOG BLOCK ***

A catalog of illegal type was found on the save tape. The illegal catalog is swapped and then deleted. An attempt is made to restore the remainder of the user catalogs.

INCORRECT CATALOG TYPE ON SAVE TAPE

Self-explanatory. Restore is terminated. Check save and list reports for possible discrepancies.

INCORRECT CAT/FILE STRING AT x...x

In the CAT/FILE string specified in the restore directive, a level other than the last level was found to be a non-catalog type. The last name in the string must be a catalog or file name. Correct string and rerun the restore.

*** INPUT BUFFER TABLE OVERFLOW ***

A user's structure contained more than 240 catalog llinks. The installation is given the option to abort or continue with the next userid.

*** INSUFFICIENT CATALOG SPACE***

The catalog portion of the disk is full and no other catalog can be added to this device. A user structure cannot be assigned to more than one device.

*** JRNL NOT SPECIFIED FOR BELOW FILE AND ONLY PIECE ON TAPE.
FILE NOT RESTORED ***

A device restore was attempted on a multidevice file, for which there was only a piece on the save tape (device save). Journaling was not specified for this file; it is inconsistent. Entire file must be restored. Restore continues with the next file.

*** MASS STORAGE ALLOCATOR DENIAL *** (A)BORT, (C)ONTINUE

The message can be given under the following circumstances:

When releasing existing catalog and file space at the beginning of a Restoremast, the Mass Storage Allocator finds some of this already marked available because of a previously aborted restore (aborted before some of the catalog or file space was marked unavailable) or because one or more disk packs (but not all) containing files (but not the catalog) were initialized by the installation prior to the Restoremast.

No space is available on any catalog devices for the user's structure. Restore is terminated.

A request was made to the Mass Storage Allocator to obtain space for an overflow SMC llink and it was denied because no space is available on the disk.

If the user answered with (A)BORT, the files may be left restore locked. Specific files can be restored by a user restore.

*** NO DATA FOR BELOW FILE ON TAPE (REPLACE)

An attempt was made to perform a replace from a restore locked file on tape. File is left defective and replace continues with the next file.

*** NO DATA ON TAPE BEYOND LL# xxx FOR FILE

A replace or device restore was attempted from a device save that did not contain all the data necessary to restore the required piece. File is left restore locked and the next file is processed.

*** NO DEF SPACE FOUND ON BELOW FILE ***

A replace was attempted but the specified file had no defective space. The file is ignored and replace continues with the next file.

*** NO DEVICE FOUND FOR CATALOGS ***

Restore program could not find any catalog device in the device directory. Restore is terminated.

*** NO DEVICE FOUND FOR DUPLICATE CATALOGS ***

There was only one catalog device configured in the system. If CATDUP is specified at startup, at least two catalog devices must be configured. Restore is terminated.

*** NO DUP SMC DEVICE. RESTORE TERMINATED ***

A device restore was attempted on ST1 but required duplicate SMCs do not exist in system. Abort.

*** NO ENTRIES IN FD EQUATE LIST ***

No files are candidates to be restored. Restore is terminated.

NO FILES MEET SINCE CRITERIA

Self-explanatory. Restore is terminated.

NO FILES RESTORED

A DOONLY user restore was run but none of the names specified were present on the save tape. Restore is terminated.

*** NOT ENOUGH MEMORY ALLOCATED FOR RESTORE ***

Restore utilizes unused allocated memory for buffer space. At least 3840 words of such memory must be available. Restore is terminated. Ask for more memory on $ LIMITS card and rerun restore.

PACK x...x NOT ON LINE

A restore was attempted to a removable structured disk pack that was not on line. Restore is terminated.

*** PACK x...x UNDEFINED OR RELEASED. RESTORE DISCONTINUED ***

Restore is attempted to an inactive pack. Restore is terminated.

*** PREVIOUS CATALOG NOT FOUND ***

A superordinate catalog was missing on the save tape. Check save report and MASLST/CLIST report (if available) taken at save time for discrepancies. Installation is given the option to abort or continue the restore. If continue, an additional message is given identifying the userid bypassed. Restore continues with the next user.

*** READ DISC ERROR ***

This message can be given under the following circumstances:

An unrecoverable I/O error occurred in reading the catalog structure from disk. Restore terminated.

An unrecoverable error occurred in reading user catalogs from disk or reading the device directory. Restore is terminated. To circumvent the problem, the installation could try to release the user structure first and then rerun the restore. If the error persists, the device directory cannot be read.

An error occurred in reading an SMC record from disk. The installation has the option to abort or continue the restore. If continue, processing of the next SMC entry is attempted and the entry on which the error occurred is not restored.

*** READ ERROR DURING COPY POSITIONING ***

A read error occurred in trying to position copy tape. Restore is terminated.

*** READ ERROR - FILE CODE FD ***

An error occurred in reading a llink from the candidate list file. Restore is terminated.

*** READ ERROR - FILE CODE SS ***

An unrecoverable error occurred in reading a structure record from temporary file SS. Restore is terminated. Attempt to rerun.

READ ERROR TRYING TO LOCK FD FOR RESTORE

An error occurred in reading an FD catalog in restore lock processing. Restore is terminated.

*** READ OR CHECKSUM ERROR OF DUP/*C CAT FOR USER x...x, FILE x...x ***

In trying to read the duplicate or monitor file catalog for above user and file in device restore, an error occurred. Installation has option to continue with the next user.

*** READ SMC ERROR ***

Self-explanatory. Installation has the option to abort or continue the restore.

*** READ TAPE ERROR *** COPY # IS $\left\{\begin{array}{l}\text{NNNNN}\\ \text{UNKNOWN}\end{array}\right\}$

Save tape cannot be read. If a copy of the save was made at save time, the restore will request to have copy tape NNNNN mounted. If no copy is available, the message will state UNKNOWN.

*** REL LL's xxx-xxx NOT FOUND ON TAPE FOR

A replace or device restore was attempted from a device save that did not contain all the data necessary to restore the required piece. File is left restore locked and the next file is processed.

*** SAVE DEVICE xxx NOT CONFIGURED ***

A restore was attempted from a device save where at least one specified device name (xxx at save time) was found not to be configured in the system to restore to. If the system wishes to continue restoring the file, the most available device will be picked upon which to put the data (unless other device constraints have been put upon the file, in which case the file must be restored again with the RENAME or RESET options specified).

*** SMC FOR x...x BUSY-WAS RESET AND IGNORED ***

An SMC entry was being updated by either a time sharing user or non-time sharing user. SMC entry was ignored. Restore is continued.

SMC x...x NOT FOUND IN SYSTEM

When a user restore is run, the SMC must already exist for that user. If the restore is to a removable structured pack, a Pack Master Catalog entry must exist as well. Restore is terminated. Correct situation and rerun the restore.

*** SPECIFIED DEVICE OR DEVICE CLASS NOT PRESENT; FILE x...x ***

The file had a device or device class constraint when it was saved that caused restore to put the file back on the device or device class that did not exist in the system the restore was being made to. Two methods for circumventing this problem are: (1) be sure the device names are configured in the system, and (2) use the RESET/DEVICE/ option in the restore directive to erase (not ignore) all device constraints and permit restore to pick the device. Restore continues with the next file.

*** STACK OVERFLOWED ***

A user structure had more than 100 levels of catalogs. The installation has the option to abort or continue the restore. If continue, the userid in question is not restored and is identified by a message. Restore continues with the next userid.

*** SUB-TASK DESCRIPTOR ERROR ***

The subtask code in the task block passed from FACT to restore does not show Restoremast and yet is called. User program bug. Restore is terminated.

SUBORDINATE POINTS TO RMVBL

An attempt was made to restore catalogs/files to a removable structured disk pack from a perm device save tape. Restore is terminated.

*** TASK BLOCK ERROR ***

Interface error between FACT and restore. User program bug. Restore is terminated.

*** TAPE FORMAT ERROR ***

An invalid task block was found on the save tape. Restore is terminated.

*** TOO MANY FD EQUATE ENTRIES FOR FILE FD ***

User attempted to restore more than 3840 files in one restore. Restore is terminated.

*** WORKING FILE NOT PRESENT ***

User program bug. Restore is terminated.

*** WRITE DISC ERROR ***

This message can occur under the following circumstances:

An error occurred in writing an SMC entry to the disk. The installation has the option to abort or continue the restore. If continue, processing of the next SMC entry is attempted but the condition of the erroneous SMCs are uncertain.

An unrecoverable error occurred when a user's catalog was written to disk or device directory. For disk, the installation has the option to abort or continue. If continue, the message is given identifying user not restored and next user subroutine is processed. For device directory, the write causes restore to terminate.

*** WRITE DISC ERROR DURING FD SEARCH FOR USER x...x ***

In searching for files to restore for user x...x during device restore, an error occurred when writing the user structure back to disk. Installation has the option to continue with next user.

*** WRITE ERROR - FILE CODE FD ***

An error occurred in writing a llink to the candidate list file. Restore is terminated.

*** WRITE ERROR - FILE CODE SS ***

An unrecoverable error occurred in trying to write a structure record to the temporary file SS. Restore is terminated. Attempt to rerun.

*** WRITE SMC ERROR ***

Self-explanatory. Installation has the option to abort or continue the restore.

WRITE ERROR TRYING TO LOCK FD FOR RESTORE

An error occurred in writing an FD catalog in restore lock processing. Restore is terminated.

UMC NOT FOUND ON SAVE TAPE

Self-explanatory. Restore is terminated. Check save library for discrepancies.

*** UMC SECTOR NOT ON LL BOUNDARY ***

User program bug. Restore is terminated.

*** UNABLE TO UPDATE SMC ***

SMC could not be updated because of read or write error (occurs in a user restore if file space is released or if file grows). File is already restored and restore continues with the next file.

*** UNABLE TO POSITION COPY TAPE ***

Restore program was unable to find the last "good" record before the I/O error occurred on the other tape (caused by wrong copy tape being mounted). Restore is terminated.

*** UNEXPECTED EOF ***

EOF record was read from save tape unexpectedly. Restore is terminated. Check the save report for possible errors.

*** UNKNOWN ENTRY NOT RESTORED ***

This message is preceded by another message identifying the error condition. Restore continues.

*** UNKNOWN SAVE OPTION - DISCONTINUE ANALYSIS ***

Save tape does not have updated software release format. Restore is terminated.

*** UNRECOVERABLE ERROR ***

An error has occurred in a device restore that makes it impossible to continue. This message is usually accompanied by another, more descriptive message.

*** UNRECOVERABLE ERROR - USERID xx...x ***

This message is preceded by another message further specifying what kind of error occurred. The installation has the option to abort or continue the restore. If continue, the userid is not restored and restore continues with the next userid.

*** USER HAS NO DUP CATALOGS TO RESTORE xxx ***

A device restore was attempted on device xxx and user x...x had catalogs on this device, but required duplicate catalogs do not exist. Installation has option to continue with next user.

*** USER x...x TRUNCATED EXCEEDED ALLOWED SPACE ***

During a restore the space used in the SMC exceeded the maximum allowed value. Restore terminates. If the maximum size is increased then the restore can be completed.

*** USERID x...x NOT FOUND IN SYSTEM ***

User catalogs were found on the save tape for which there was no SMC entry (all SMCs have been restored at this point). Restore continues with the next user. Check both save and restore reports for errors in SMC processing of the user; this user is not restored.

*** USERID *** x...x *** NOT RESTORED ***

This message is preceded by message(s) identifying the problem. Restore continues with the next userid.

x...x OPTION INVALID FOR FILE

User attempted to restore a single file with DOONLY, DONTDO, or SINCE. Restore is terminated. Remove option and rerun the restore.

*** xxx AND xxx ARE BOTH CAT DEVICES. USER x...x ***

A device restore was attempted on devices xxx and xxx and user x...x had original duplicate catalogs on these devices. Installation has option to continue with next user.

ERROR MESSAGES PRODUCED BY ROLLFORWARD DIRECTIVE

BAD TAPE FORMAT

After a file has been updated, the next record expected from the journal file does not agree with what the program expected. The update is marked as incomplete if any pages were written to the file. Otherwise it is marked as being skipped. If the journal file can be repositioned successfully, processing will continue.

CANT ROLLFWD AFTER FR/FC

The file has been deleted and recreated in the time between the actual update by the user job and the present rollforward update. In order for rollforward to update the file, it must first be restored from the latest save file. Until this is done, the rollforward update activity is skipped.

DEVICE NAME UNKNOWN

The file is presently described as residing on a device not configured in the system. The update is marked as incomplete if any pages were written to the file before the condition occurred. Otherwise it is marked as being skipped.

DIFFERENT PAGE SIZES

The page size of the file does not agree with the page size on the journal file. The update activity is skipped.

DISQUALIFIED BY OPTION

The rollforward update activity for the listed file is skipped because of an option selected on the ROLLFWD directive; i.e., the file does not appear in a DOONLY list.

ENTRY IS NOT A FILE

The entry named on the journal file is no longer defined as a file in FMS. The update activity is skipped.

FILE IN DEFECTIVE STATUS

While performing the update activity, write errors occurred and the file has been placed in defective status. The final status of the file appears in the defective file summary report. Finally, the update is marked as being incomplete.

FILE IS BUSY, TRY LATER

Request by rollforward update refused because of present allocation or allocations. The rollforward requests L (load) permission for the update. The activity is skipped; it is not put in a delay list for later processing. It is the site responsibility to run the update at a time when it is known that the file is not busy.

FILE NOT JOURNALED

The file currently does not have journalization specified. The update activity is skipped.

MAXIMUM NUMBER OF DEFECTIVE FILES EXCEEDED, AUTO RESTART ENGAGED

The limit of 50 defective files has just been exceeded. A special indicator is set and the updates on files not in defective status or one of the 50 files already found to be in defective status continues. After the current journal file is processed, and the defective file summary report is produced, those files removed from defective status are removed from the table. The current journal file is then reprocessed to locate and update only those files in defective status. If the DEFECTIVE option was not selected on the directive, WITH DEFECTIVE OPTION is added to the message.

PAGE NO. OUTSIDE OF FILE LIMITS

A page number has been found on the journal file that is larger than the largest file page number. The update is marked as incomplete if any file pages were updated; otherwise the update activity is skipped.

***** ROLL FORWARD ACTIVITY COMPLETE *****

This message indicates the end of file has been reached on the current journal file. The summary reports for roll forward and defective files are produced.

START PROCESSING ON JOURNAL FILE fffff, REEL rrrrr, SEQ ss

fffff is the current journal file serial number, rrrrr is the current journal reel serial number and ss is the reel sequence number. This message indicates when the first and following journal tapes are mounted and read by the program.

*** TAPE FORMAT ERROR ***

An invalid record identifier was read from the journal file. Processing continues with the next directive but the current activity terminates with an FS abort.

*** TASK BLOCK ERROR ***

Interface error between FILSYS and rollforward. This is a user program bug. Processing continues with the next directive but the current activity terminates with an FS abort.

TOO MANY DEFECTIVE FILES

The limit of 50 defective files has been exceeded. The automatic roll forward restart is engaged on the first occurrence of this message. The update activity is skipped for now but is processed later; i.e., the update is deferred till an auto restart run is initiated.

*** UNEXPECTED EOF ***

An unexpected EOF was encountered on the journal file. The roll forward activity goes to completion. check the rollforward and journal writer reports for errors.

UNRECOVERABLE xxxx ERROR

where xxxx may be TAPE or DISK. If TAPE, the journal file is unreadable and the rollforward activity goes to completion. If DISK, then the rollforward program cannot read or write the catalog sector describing the file. The update is marked as incomplete if any file pages were updated; otherwise the update activity is skipped.

USERID: zzz...z, CAN'T LOCATE FILE FOR ROLL FORWARD

A file has been found in an update activity on the journal file that is not currently in the system. The update is skipped. The update can be tried again after the file is restored.

$\left\{\begin{array}{l}\text{ALTBLS}\\\text{RPTBLS}\end{array}\right\}$ FILE USED FOR RECOVERY

This message indicates which recovery file remains usable. Another message preceding this message indicates whether a read or write error occurred on the other recovery file.

CANT ROLLBACK AFTER FR/FC

The file was deleted (FR) and recreated (FC) after the last update and before the recovery attempt. Recovery of the file is skipped.

*** CHECKSUM IS xxx...x SHOULD BE yyy...y ***

A checksum error was encountered upon validation of the checksum on the PATCH directive. Change the checksum option and rerun the PATCH directive.

DEVICE name INITED, MUST RECOVER FILE BY RESTORE

The device specified was initialized and contained part or all of a non-duplicated file. Recovery cannot recover file in this configuration. The file must be restored or reloaded for recovery.

DEVICE name INITED, ONLY $\left\{\begin{array}{l}\text{ORIGINAL}\\\text{DUPLICATE}\end{array}\right\}$ FILE RECOVERED

The specified device was initialized and contained part or all of one copy of a duplicated file. After the file is recovered a restore must be issued to update the "bad" copy from the "good" copy.

DEVICE name INITED, RESTORE DEVICE, THEN RECOVER

The specified device was initialized and contained a catalog structure that had not been restored with a device restore. After the restore is completed the recovery program should be rerun.

DEVICE name INITED, WAS COLLECTION FILE DEVICE

The specified device was initialized and contained the collection file for the specified file. The file cannot be recovered. The file must be restored or reloaded for recovery.

DEVICE name UNKNOWN

The specified device is unknown or undefined.

DIFFERENT PAGE SIZES

The page size of the file does not match the page size of the collection file. Recovery of the entry is skipped.

ENTRY IS NOT A FILE

The entry specified on the collection file is no longer defined as a file. Recovery of the entry is skipped.

$$\text{ERROR}\left\{\begin{array}{l}\text{READ}\\\text{WRIT}\end{array}\right\}\text{ING}\left\{\begin{array}{l}\text{COLLECTION}\\\text{PERMANENT}\\\text{TEMPORARY}\end{array}\right\}\text{FILE ,ONLY}\left\{\begin{array}{l}\text{ORIGINAL}\\\text{DUPLICATE}\end{array}\right\}\text{FILE RECOVERED}$$

An I/O error occurred. Recovery is deferred if the error occurred on a temporary file and may be rerun. Recovery is skipped if the error occurred on a collection file, a non-duplicated file, or a duplicated file in which both copies have been marked defective. Corrective action may be taken with a REPLACE directive followed by another recovery attempt. If journaling was specified, corrective action is taken with a REPLACE directive followed by ROLLFWD DOONLY/ file name/, DEFECTIVE and another recovery attempt. If the file is duplicated and one "good" copy is present, recovery continues on the copy indicated in the message. After recovery of the file is completed, replace and restore directives should be used to produce copies that are correct and consistent.

$$\text{ERROR}\left\{\begin{array}{l}\text{READ}\\\text{WRIT}\\\text{WAIT}\end{array}\right\}\text{ING FILE}\left\{\begin{array}{l}\text{RCVYINFO}\\\text{RPTBLS}\\\text{ALTBLS}\end{array}\right\}$$

An error occurred while the file was being read or written. If the error occurred on the recovery information file, program execution terminates. If the error occurred on the RPTBLS file, a message appears indicating that the ALTBS file is used for recovery and vice versa. If no other file is available for recovery, the program terminates.

*** USER SPECIFIED SNUMB IS WRONG ***

The SNUMB specified in the RECDUMP or REMOVE directive does not exist in the recovery tables files.

*** ERROR WAS FOUND ON READING C.F. ***

The RECDUMP directive attempted to read a collection file and encountered an error. Processing continues with the next entry in the recovery files.

FILE IN DEFECTIVE STATUS

A file in defective status was encountered. Corrective action may be taken with a REPLACE directive followed by another file recovery attempt. If the file is journalized, ROLLFWD DOONLY/filename/,DEFECTIVE should follow the REPLACE directive.

FILE IS TDS LOCKED

File was being used by TDS. The FMS recovery program will not act on this file because TDS is expected to perform the recovery.

FILE RECOVERED BY RESTORE

The recovery program encountered a file recovered by the restore program. No recovery is required.

FILE RECOVERED BY RESTORE &/OR USER

The recovery program encountered a file that was recovered by the restore program or a user program. No recovery is required.

FILE RECOVERED BY RESTORE &/OR USER&/OR JOURNAL

The recovery program encountered a file that was recovered by the restore program, a user program, or the FMS journal, i.e., the file also has RDERR/JOURNAL/. No recovery is required.

FILE RECOVERY DEFERRED, "FMS diagnostic message"

An attempt was made to recover a file that was deferred. The deferral is explained by the FMS diagnostic message (see Appendix F).

FILE RECOVERY NOT NEEDED

The file specified was being used with T or T/C allocation. No recovery is required.

GMOR DENIAL FOR TEMP FILE

Request for a 3-link temporary file was denied and recovery was deferred. The RECOVER directive may be rerun when space is available for the temporary file.

*** INTERNAL PROGRAM ERROR ***

A non-recoverable error was encountered during processing of the RECDUMP/REMOVE directives. Either the PAT could not be built for the collection file with the RECDUMP directive or the recovery tables could nct be found a second time with the REMOVE directive.

*** I/O ERROR ***

A read or write disk error occurred during processing of the RECDUMP or PATCH directives. Directive processing terminates.

NOT USABLE-ON INITED DEVICE

The recovery file obtained is unusable because it was on a initialized device. Recovery may continue if the other recovery file is usable.

PAGE NO. OUTSIDE FILE LIMITS

A page number that exceeds the highest page number in the file was encountered. No recovery is attempted and the file remains in abort lock status.

PROGRAM EXECUTION TERMINATES

Errors encountered in the recovery program caused premature termination. The errors should be corrected if possible and the recovery program rerun.

RECOVERY PERMISSION DENIED "FMS diagnostic message"

An attempt to allocate the RPTBLS file with recovery permission failed as explained by the FMS diagnostic message. (See Appendix F). The RPTBLS file is not usable when this message appears.

ROLLBACK NOT SPECIFIED

This file no longer has ABORT/ROLLBACK/ specified. Recovery of the file is skipped.

*** TASK BLOCK ERROR ***

An interface error exists between FILSYS and the recovery program or one of its overlays. The current activity terminates with an FS abort, but processing continues with the next directive. A snapshot of the buffer containing the error may be obtained.

UNRECOVERABLE DISC ERROR

A read or write occurred on the file catalog structure.

USERID: user name, CAN'T LOCATE FILE FOR RECOVERY

A collection file was encountered for a user identification that is not currently in the system. Recovery can be attempted after the user is restored.

APPENDIX C

$ FILSYS CONTROL CARD FORMAT AND DECK SETUP FOR $ FILSYS ACTIVITY

The $ FILSYS control card calls the FILSYS activity and has the following format:

```
1      8         16
$      FILSYS    options
```

The options are as follows, with standard options underlined:

DUMP - slave memory is dumped if the activity terminates abnormally and if the activity is submitted with a $ PRIVITY control card.

NDUMP - program registers, upper SSA, and slave program prefix are dumped if the activity terminates abnormally.

JREST - enable job restart

NJREST - do not restart this job

REST - enable activity restart

NREST - do not restart this job with current activity

The following deck setup is representative of a $ FILSYS activity for user directives described in Section IX.

```
1      8         16

$      SNUMB
$      IDENT
$      USERID                 (Optional;  convenient if only one user is to be
                              referenced.  If not used, directives must include
                              one or more USERID directives.)
$      FILSYS   options       (DUMP option honored only if  $ PRIVITY card  is
                              present.)
$      LIMITS                 (Optional)
               ⎧SR⎫
$      FILE    ⎨SS⎬ ,lud,R,nR (Optional, disposition must be R)
               ⎩FD⎭
          .
    directives
          .
          .
$      ENDJOB
***EOF
```

The following deck setup is representative of a $ FILSYS activity for privileged directives described in Section X.

```
1      8          16
________________________________
$      SNUMB
$      IDENT
$      FILSYS     options
$      PRIVITY
$      LIMITS                  (Optional)
       .
       .
       .
  directives
       .
       .
       .
$      ENDJOB
***EOF
```

The following deck setup is representative of a $ FILSYS activity for the save, restore, and rollforward directives described in Section XI. File codes for the $ TAPE control cards are:

PS	for save directives
XS	for save tape copying
PR	for restore directives
JF	for rollforward directives

```
1      8         16
_______________________________________
$      SNUMB
$      IDENT
$      USERID                           (Optional; convenient if for user
                                        save  and  restore  activity  not
                                        requiring PRIVITY.  Not  used  for
                                        master save and restore activity.)
$      FILSYS    options
$      PRIVITY                          (Not  required  for  user  save  and
                                        restore activity.)
$      TAPE      file code, options     (One required)
$      TAPE      file code, options     (Optional; file code XS  if  previous
                                        tape is file code PS; file code JF if
                                        previous tape is file code PR and the
                                        rollforward   directive   follows   a
                                        restore directive.)
$      LIMITS                           (Optional)
       .
  directives
       .
       .
       .
$      ENDJOB
***EOF
```

The following is an example of a list of FILSYS directives interspersed with some mode directives and comment cards as described in Section VIII under "Usage Notes". The comment cards with an asterisk comment on the mode directive. The mode directives start in column 2 and the FILSYS directives start in column 1.

```
*       PROCESS DIRECTIVES AFTER A DETECTED ERROR
 IGNORE ERRS
RELPAC (variable field)
RELMAS (variable field)
*       CANCEL THE EFFECT OF IGNORE ERRORS DIRECTIVE
 NOTICE ERRS
CRMAST (variable field)
CCREATE (variable field)
*       SUPPRESS THE PRINTING OF PASSWORDS
 HIDE PASSWDS
CLIST (variable field)
*       CANCEL THE EFFECT OF HIDE PASSWDS DIRECTIVE
 SHOW PASSWDS
*       LIST EACH USER ON A SEPARATE PAGE
 USER PER PAGE
MASLST (variable field)
```

For multi-$ FILSYS activities, the $ IDENT, $ USERID cards and $ ENDJOB, ***EOF cards bind the first and last activity, respectively. The implied SYSOUT limit for FILSYS is 1000 lines. This limit should be increased if the LISTOPT YES or ALL option is used in a SAVE, SAVEMAST, RESTORE, RESTOREMAST, CLIST, or MASLST directive, if a large structure is involved.

If a removable structured disk pack is required for a $ FILSYS activity, then a $ xxxPK control card is required in the deck setup after the $ PRIVITY control card. The ownership field must contain the parameter PUBLIC (see Control Cards Reference Manual).

APPENDIX D

FILE CONVERSION AND CATALOG RESTRUCTURING

INTRODUCTION

The File Management Supervisor (FMS) requires that all installations convert ASCII (media code 5 and 6) files from pre-Software Release (SR) E formats to FMS standard system format. In conjunction with the file conversion, catalogs must be also restructured.

Conversion of the ASCII files is performed by a system loadable program (FILCNV) by reformatting the file content on pre-SR E save tapes. Catalog restructuring is performed from the reformatted pre-SR E save tapes by a system loadable program (.SRFCT). A conversion from the SR E catalog and file structure to pre-SR E catalog and file structure is performed by the .SRFCT program.

.SRFCT contains module RFCT. FILCNV executes any release to perform file conversion; .SRFCT executes on SR E or later to reformat the catalogs and (if required) to convert FMS catalogs and files to pre-SR E formats.

The conversion process may reduce or expand the file size. Expansion is reflected in both the current and the maximum size fields on the conversion/analysis report. The user's SMC entry is adjusted to reflect the net change in file sizes.

GENERAL NOTES ON CONVERSION

1. FILCNV processes master save tapes only.

2. Files with conversion errors which "bridge" two reels of a multiple reel file are deleted from the output tape. It is suggested that multiple reel saves be avoided as a general practice.

3. Some save tapes may contain media code 5 files where it is desirable not to convert some of them to FMS format (e.g., Time Sharing FORTRAN data files). The following alternatives are suggested to accomplish the wanted conversions. Volume considerations will dictate the alternative used.

   a. The user can elect to skip the FILCNV conversion, and upon restoration to FMS, use ASCASC to convert selectively to FMS format.

b. The user can elect to convert the entire tape, and upon restoration to FMS, use ASCASC to reconvert selectively back to media code 5.

c. The user can elect to convert the entire tape and restore it to FMS. Using .SRFCT, a selective save could be done, naming only those files to be converted back to media code 5. This tape would then be restored to FMS, as no conversion takes place at conversion time.

4. Partitioned records are not converted by RCNV to media code 6 standard system format.

PRE-SR E TO SR E CONVERSION

Pre-SR E to SR E Conversion is performed by the FILCNV module. This program performs the conversion, provides an analysis/conversion report, and provides a converted tape. It can execute in an analysis-only mode or run in the conversion mode.

File Codes

The file codes and explanations of file codes used in the deck setups are as follows:

File Code	Meaning
D6	Input tape - SR D save tape
E7	Output tape - Converted save tape
AC	Work tape - Current converted file
A2	Work tape - Current user copied/converted files
SR	File/Report Collector

Save Tape Records

Type 1	SMC create record
Type 2	Catalog create record
Type 3	File create record
Type 4	File content record

Analysis Report

The analysis report describes file structures, type of file encountered, size, and conversion information. A sample report is included in this appendix.

Analysis Mode

When the analysis (report only) mode is requested, the user sets bit 18 on (see "Deck Setups"). Only files D6 and SR are required. Analysis is performed on each file encountered and a report is prepared as if conversion had been attempted. If no media code 5 or 6 files are encountered, the conversion run can be bypassed. If conversion is required, the program can be run in a conversion mode.

Conversion Mode

In the conversion mode, all five files (D6, E7, AC, A2, and SR) are required and bit 18 is not set. A report is prepared and conversion is attempted.

When FILCNV reads a Type 1 record, it is retained in memory for later updating. Type 2, catalog create, records are copied to file A2. Type 3, file create, records are retained in memory for possible updating. Type 4 records are inspected for conversion processing. Non-conversion candidates are copied directly to file A2 preceded by the associated Type 3 records. Type 4 records requiring conversion are written to file AC in a converted form. If no conversion errors exist on the file, the Type 3 record in memory is updated and written to file A2. The converted file on AC is then copied to file A2. If an error occurs in conversion while writing to AC, the error type is noted and conversion is stopped; file code D6 is backspaced to the beginning of the file in process; and the unconverted file is copied to file A2 preceded by the Type 3 record.

Conversion for a user is complete when another Type 1 record is read. The retained Type 1 record is updated to reflect new size requirements and written to file E7. User catalog and file data is copied from file A2 to file E7 to complete the process. Totals for the user are reported and the conversion process continued.

When the end-of-file occurs on file D6, a summary is prepared for all users. This reflects size requirements and errors encountered. File E7 can then be used to restore to FMS structures.

Size Changes

No size changes are made for random (RAND) or sequential (SEQ) files. A converted file (TSS05, TSS06 on sample summary) may be reduced or expanded from the current size. Any reduction is reflected in the Type 3 current size field and does not affect the maximum size. Any expansion modifies the current size field and increases the maximum size to accommodate the file. Net changes in size for all files affect the current size in the Type 1 record. Maximum size in the Type 1 is reflected in the analysis display.

Deck Setups

The deck setups to extract module FILCNV from a Secondary Software Object Library for SR E or a later release perform the conversion to FMS format, perform the analysis, and restructure the catalogs are as follows:

EXTRACT FILCNV FROM SECONDARY SOFTWARE OBJECT LIBRARY

The following deck setup can be used to extract the FILCNV module from the Secondary Software Object tape delivered with SR E. Before performing this activity, the user must create a permanent file space with a size of eight random links.

```
$       8       16
_________________________________
$       SNUMB
$       IDENT
$       USERID
$       SYSEDIT
INCLUDE/FILCNV/
$       LIMITS  ,48K
$       TAPE    R*,XID,,#####,,SEC-SOF-OBJ
$       PRMFL   Q*,R/W,R,USERID/FILENAME
$       ENDJOB
***EOF
```

= Secondary software object tape delivered with SR E.

NOTE: This step is required only if the user wishes to run FILCNV on a pre-SR E release.

ANALYSIS ONLY OF MASTER SAVE TAPE

The following deck setup uses the save tape (file code D6) to perform an analysis-only run. This setup is for a run on a pre-SR E release and assumes that the activity to extract FILCNV has been performed. For a run on SR E or a later release, omit the $ PRMFL control card.

```
$       8       16
_________________________________
$       SNUMB
$       IDENT
$       USERID
$       PROGRAM FILCNV
$       PRMFL   **,R,R,USERID/FILENAME
$       LIMITS  99,30K,,20K
$       SET     18
$       TAPE    D6,11DD,,99999,,INPUT
$       FILE    SR,11R,10L
$       ENDJOB
***EOF
```

CONVERSION AND ANALYSIS OF MASTER SAVE TAPE

The following deck setup uses the master save tape (file code D6) to perform a conversion and an analysis of the save tape. This deck setup is for a run on a pre-SR E release and assumes that the activity to extract FILCNV has been performed. For a run on SR E or a later release, omit the $ PRMFL control card.

```
$       8       16
_______________________________

$       SNUMB
$       IDENT
$       USERID
$       PROGRAM  FILCNV
$       PRMFL    **,R,R,USERID/FILENAME
$       LIMITS   99,30K,,20K
$       TAPE     D6,11DD,,99999,,INPUT
$       TAPE     E7,12DD,,99999,,OUTPUT
$       TAPE     AC,13R
$       TAPE     A2,14R
$       FILE     SR,11R,10L
$       ENDJOB
***EOF
```

Output from file code E7 can be restored under SR E or later catalog structures using .SRFCT (see "Reformat Catalogs in FMS Structures").

TIMING

Experience indicates that FILCNV will execute for an average of 0.11 hour in the analysis mode for one reel of tape; conversion averages 0.56 hour per reel of tape. Multiple conversions can be in execution concurrently.

CONVERSION ANALYSIS EXAMPLE

The following three pages contain a sample output of the conversion analysis from the FILCNV program. The page following the sample output contains an explanation of the items indexed.

```
RCNV  01  08-08-73   15.870   FMS CONVERSION/ANALYSIS PROGRAM     730712     REEL#  23243            PAGE     12

USERID------  C#  CATALOG-----  FILE--------  TYPE--  ERRORS  PRE-SZ  CVT-SZ  NET-CHG
OPNSUTIL①
                OPNSUTIL②                                   ⑥        ⑦         ⑧
                              TDUMP③        NOFILE④           5
                              CIM06LEJ      SEQ       COPY⑤   2
                              BMC08         SEQ       COPY    1
                              LBT11         SEQ       COPY    2
                              DUM10         SEQ       COPY    1
                              SINC1         SEQ       COPY    1
                              COMP1         NOFILE            1
                              UTC07         SEQ       COPY    1
                              RTP12         SEQ       COPY    1
                              BOD16         SEQ       COPY    2
                              CAT01         SEQ       COPY    1
                              SINC2         SEQ       COPY    1
                              CAT02         SEQ       COPY    1                                  ⑨
                              SSAVE         SEQ       COPY   11
                              TTYTEST       TSS05   YES-07   34                          PHYREC   481 BSN     27
                              LAT15         SEQ       COPY    2
                              REPPU         SEQ       COPY    1
                              SNOOP         SEQ       COPY    1
                              COUNT         SEQ       COPY   10
                              DMAPS         SEQ       COPY    7
                              UNPRG1        SEQ       COPY    4
                              355BT         SEQ       COPY   50
                              A             TSS05     CONV   45        3      -42
                              VIDEO         SEQ       COPY    1
                              HEALS         SEQ       COPY    1
                              UNPUR         SEQ       COPY    1
                              MMD           RAND      COPY  120
                              CORUS         SEQ       COPY    1
              ⑪OPNSUTIL/E7⑩  FILUS         SEQ       COPY    3

   MODIFIED FILE SIZE FROM    312 TO    270.  NET CHANGE    -42 LLINKS, FOR     29 FILES.    SMC MAX. CHANGE =   NONE.
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
          1  ERRORS IN THE CONVERSION OF FILES FOR THIS USERID.
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
 ⑫
CONVERSION/ANALYSIS ERROR CODE LEGEND:
   01-CHARACTER STRING DID NOT END WITH A 'CR'.                02-LAST BLOCK NOT A TERMINAL BLOCK OR NO END-OF-FILE FOUND.
   03-BAD CHARACTER COUNT.                                     04-BLOCK SERIAL NUMBER ERROR.
   05-INCORRECT RECORD CONTROL WORD.                           06-SUSPECTED PARTIONED RECORD.
   07-INCORRECT BLOCK WORD COUNT.                              08-A FILE SPANNING 2 REELS WAS NOT OUTPUT.
   09-FILE CURRENT SIZE EXCEEDED MAXIMUM SIZE.
```

① USERID - SMC CREATE RECORD

② UMC NAME - CATALOG CREATE RECORD

③ FILE NAME - FILE CREATE RECORD

④ FILE TYPE ENCOUNTERED

⑤ ACTION TAKEN BY RCNV
YES-07 INDICATES COPY BECAUSE OF ERROR CODE 7

⑥ CURRENT SIZE IN FILE CREATE RECORD

⑦ NEW CURRENT SIZE IN FILE CREATE RECORD

⑧ SIZE CHANGE ON CONVERTED FILE

⑨ TAPE INPUT RECORD ERROR LOCATION AND USER FILE BSN

⑩ CATALOG CREATE RECORD

⑪ SUMMARY OF USER CHANGES

⑫ ERROR CODES

⑬ SUMMARY DATA FOR ALL USERS

⑭ ERROR INDICATIONS BY USER

FILCNV ERRORS

The errors that can be detected by FILCNV during conversion to media code 6 standard system format are as follows:

Code	Error
1	Character string did not end with a carriage return (CR)
2	Last llink not a terminal llink or no end-of-file indicator on file
3	Bad character count (string extends beyond data llink)
4	Block serial number error (sequence)
5	Incorrect record control word (size or media code do not match expected format)
6	Suspected partitioned record (could be an error or a partitioned record for Time Sharing FORTRAN)
7	Incorrect llink word count (size error or does not match expected format on media code 5)
8	A file spanning two reels was not output (conversion error on file "bridging" two save reels)
9	File current size exceeded maximum size

Codes 1 through 7 force a file copy (unconverted). The maximum size field in the Type 2 record is adjusted if a code 9 error is encountered.

REFORMAT CATALOGS IN FMS STRUCTURES

Program .SRFCT reads pre-SR E save tapes converted with FILCNV and reformats catalog structure to FMS requirements. The following directives are recognized. All options (except LISTOPT) listed for these directives in the manual File System, BR38, are recognized.

```
RESTORE
RESTOREMAST
USERID
```

Program .SRFCT runs as a privileged slave program regardless of the directive employed.

The following deck setup can be used to reformat pre-SR E save tapes (converted or unconverted) to FMS structures. This activity is setup to run under SR E or a later release.

```
1      8       16
__________________________

$      SNUMB
$      IDENT
$      PROGRAM .SRFCT
$      LIMITS  99,32K,,10K
$      PRIVITY
RESTOREMAST
$      TAPE    PR,11DD,,99999,,D6SAVETAPE
$      ENDJOB
***EOF
```

PR = Primary restore tape

If bit 18 of the user switch word (.SWIT) is on (SET = 1 by a $ SET 18), and the DEVICE/name/ option is present for a catalog or file, the system name table is scanned for an equal name. If no match is found, the option is changed to permit the file system to select the device.

If the DEVICE/class/ option is present, the following test occurs:

1. If the device type is configured, leave the option as is.
2. If the device is a DSS181, change it to a MS0310 and repeat test.
3. If the device is a DSS190, change it to a DSS191 and repeat test.
4. If the device is a DSS191, change it to a MS0400 and repeat test.
5. If a permanent device was requested, change the option to allow the file system to select the device.
6. If structured or nonstructured removable device, output error message, "DEVICE TYPE xxxxxx NOT CONFIGURED", and discontinue processing on the current catalog or file.

SR E TO PRE-SR E CONVERSION

Program .SRFCT can be used to "save" FMS catalogs and files in pre-SR E save tape format. It is used when saving FMS files and structures for restoration to a pre-SR E software release. The following options are available: SAVEMAST, SAVE, USERID. If the $ PROGRAM control card is used without the ON6 option as indicated in the example, all media code 6 standard system format files are converted to pre-SR E media code 5 format. The user can convert all media code 6 standard system format files to pre-SR E media code 6 files by using ON6 on the $ PROGRAM control card. Media code 5 and pre-SR E media code 6 files are not converted if they are present on the files being saved.

After catalogs and files have been restored, the user can reconvert media code 6 standard system format files to media code 5 by the use of the time sharing ASCASC subsystem. The following deck setup can be used for the conversion:

```
1      8       16
$      SNUMB
$      IDENT
$      PROGRAM .SRFCT
$      LIMITS  99,32K,10K
$      PRIVITY
SAVEMAST
$      TAPE    PS,11DD,,99999,,RETREAT SAVE
```

PS = SR E primary save tape

.SRFCT Error Detection

The errors that can be detected during the conversion from FMS to pre-SR E are as follows:

Code	Error
1	Expected BSN 1 (on first record)
2	No header record on BSN 1 (media 8 record not present)
3	System option illegal (invalid conversion type request)
4	BSN error (sequence)
5	Record exceeds llink length (size error on control word)
6	BCW is greater than 319 words (size error on llink control word)
7	Non-media 6 in RCW
8	Illegal EOF indicator word (EOF, but improper format)
9	Zero size REC (record control word with no data words)

Errors, when detected, stop the conversion process. The error location on tape is identified for the user. Remaining blocks of the file are copied without converting. Partially converted files will be present on the save tape.

ALGORITHM FOR ADJUSTMENT OF SIZE

The following is the algorithm used to adjust the maximum llinks allowed field of a user created master record.

Definitions

U = Llinks used, before conversion

M = Maximum llinks allowed, before conversion

U_c = Llinks used, after conversion

M_c = Maximum llinks allowed, after conversion

B = Number of llinks increased during conversion

NOTE: If $1.125*U \geq M$, then a message of the number of llinks available is already output on the Time Sharing System log-on message.

After Conversion

Case 1:	If $1.125*U < M$, and $1.125*U_c < M$, then $M_c = M$	(No message) (No change)
Case 2:	If $1.125*U < M$, but $1.125*U_c \geq M$, then $M_c =$ $(U_c *1.125) + 1$	(No message) (Grow to insure no message)
Case 3:	If $1.125*U \geq M$, and $1.125*U_c \geq M$, then $M_c = M + B$	(Message) New = old + grow
Case 4:	If $1.125*U \geq M$, and $1.125*U_c < M$, then $M_c = M$	(Message) New = old

APPENDIX E

I/O STATUS RETURNS SENT BY FMS

The File Management Supervisor returns an exception status on requests to read from or write to a protected file on occasions when the request cannot be initiated. The status returned and its meaning for each are listed below.

<u>Status</u> — <u>Meaning</u>

Octal code in bits 0-14 of first I/O return word

47771 Illegal request to access a protected file occurred because:

a. I/O request was for more than one page.

b. seek address was not to a page origin.

c. I/O was issued at Courtesy Call level and neither can be delayed due to access conflict nor await completion of another I/O.

d. sequential access to a file using the ACCESS/MONITOR/ option was attempted.

e. I/O request was attempted which is not relative to the slave LAL.

f. ACCESS/MONITOR/ control tables (prepared by FMS) were destroyed.

g. abort request for the program was received prior to I/O request for processing.

h. I/O request to ACCESS/MONITOR/ protected file is denied when there is only a single user updating or reading the file and the request exceeds the maximum number of reserved pages (default is 300).

47772 I/O request to file using ACCESS/MONITOR/ is denied to avoid a deadlock.

47773 The FMS collection file for after images (RDERR/JOURNAL/ option or TEST allocation) cannot be expanded or has reached its maximum size of 4032 pages. The JOURNAL option requires that user checkpoint files (MME GEFSYE function 41, Regard Changes as Complete) at least once after 4032 pages are changed. A maximum of 4032 updates are allowed for a job using TEST allocation.

47774 A concurrent access conflict exists on a file allocated to TSS.

47775 I/O is attempted to a file using RDERR/DUP/ option and both file copies are marked unusable.

47776
47777 Reserved for future FMS use.

APPENDIX F

RETURN CODES AND MESSAGES

Return Code	Diagnostic Message
01	USER-ID NOT IN MASTER CATALOG
02	I/O ERROR ON DEVICE XXX SA = NNN......NNN
03	PERMISSIONS DENIED
04	FILE BUSY; TRY LATER
05	INCORRECT CAT/FILE DESCRIPTION AT AAA......AAA
06	NO SPACE FOR CATALOG ON DEVICE XXX
07	UNDEFINED DEVICE YYYY ZZZZZZ
10	NO SPACE FOR FILE ON DEVICE YYY
11	NONUNIQUE NAME
12	SIZE REQUEST LS THAN ALLOCATE
13	SPACE REQUEST GR THAN ALLOWED
14	INCORRECT OR MISSING PASSWORD AT AAA......AAA
15	FILE IS ABORT LOCKED
16	FILE WRITE IN BATCH ONLY
17	SEEK ERROR ON DEVICE XXX SA = NNN......NNN
20	FAILURE IN NAME SCAN (IMP.)
21	DUP CANNOT BE ON RMVBL DEVICE
22	BAD SPACE INVENTORY ON DEVICE XXX
23	INVALID MME PARAMETER
24	BAD SPACE TABLE FOR DEVICE ON XXX
25	REQUESTED ENTRY NOT ON-LINE
26	NON-STRUCTURED FILE ENTRY
27	FILE IN DEFECTIVE STATUS
30	ILLEGAL PACK TYPE
31	ACCESS GRANTED TO I-D-S FILE
32	COLLECTION FILE ERROR
33	CATALOG/FILE SECURITY LOCKED
40	NEED MORE PAT SPACE
41	NO PROTECTION TABLE SPACE AVAILABLE
42	INVALID FILE CODE OR PAT PTR
43	INVALID CATALOG BLOCK ADDRESS
44	ILLEGAL OPTIONS COMBINATION
45	INVALID SPACE IDENTIFIER
46	CATALOGS BUSY (SPECIAL TSS INTERFACE)
47	UNACCOUNTABLE ERROR
50	***RESERVED for FMS***
51	CHECKSUM ERROR - DEVICE XXX SA = NNN......NNN
52	DEVICE XXX RELEASED
53	***RESERVED FOR FMS***
54	***RESERVED FOR FMS***
55	***RESERVED FOR FMS***
56	SECURITY PARAMETER - REQUIRED
57	SECURITY PARAMETER - INVALID
61	$FSYS HAS BEEN ENABLED
62	ILLEGAL SUBFUNCTION CODE

63	FILE NOT BEING MONITORED
64	DEADLOCK ON PAGE REQUEST
65	PAGE CURRENTLY BUSY
66	FILE NOT DUPLICATED
67	TDS MON ALLOC ERROR
70	ILLEGAL CHECKPOINT REQUEST
71	ILLEGAL DCW SPECIFIED
72	IMPROPER PROTECTION OPTION
73	INVALID ARG LIST PARAMETER NO. 000
74	SYSTEM JOURNAL NOT CONFIGURED
75	FILE RESTORE LOCKED
76	FILE IS TDS LOCKED
77	ERR TDS SUBSET PAGES RELEASED
100	UNACCOUNTABLE ERROR

where	XXX	=DEVICE NAME(ST1,DS1,...)
	NNN......NNN	=OCTAL REPRESENTATION OF THE SEEK ADDRESS
	AAA......AAA	=12 BCD CHARACTERS OF THE CATALOG ELEMENT IN ERROR
	YYY	=TYPE/OR/NAME
	ZZZZZZ	=DEVICE NAME OR TYPE OF DEVICE
	000	=OCTAL NUMBER

NOTE: Meaning of messages is described in Appendix B.

APPENDIX G

FSAD UTILITY PROGRAM

The FSAD program is a privileged program designed to inspect catalog structures on permanent and removable packs, analyze and report specific errors, display catalog structures, and dump arbitrary portions of a disk pack. Directives to this program are entered either in card form or from the console typewriter. Output is a report containing device directory information, the SMC/PMC llink for the class being analyzed and the user's structure under analysis, and a list of the user's option settings and directives which may vary from input directives. Output of dump directives is in llinks or sectors.

PROGRAM DESCRIPTION

This program is edited on the software library, under the catalog name FSAD, and is called by a $ PROGRAM control card. A $ PRIVITY control card is required in the activity that calls FSAD. A $ LIMITS control card may be required for a large output. A list of the structure errors checked by the program that are reported includes the following:

a. Comparison of catalog structure descriptors and file descriptors against the AST or the device space table

b. Checksums and comparison against duplicate catalogs

c. Illegal catalog structure record types

d. Non-existent device names for files

e. Strange descriptors (unrecognizable formats)

f. Structure read errors

g. Directive errors and items not found

h. Locked files (not in error)

A bad structure may cause FSAD to relinquish the structure llink chain. This occurs when the prior llink pointer has an erroneous value or the five sectors of the llink have a checksum error.

The entire llink within which errors are encountered is printed during the error pass for options 20 and 21. Both the message "DUP LLINK NOT IDENTICAL" and the duplicate llink (not primary) are printed during error processing.

All options are directive input. The four options available (listed under card format) can be entered anywhere and apply to the entire directive. Generated directives are appended to the end of the directive list and function with options null.

Except for the option directives, all directives are device oriented. The device in structure directives is the device of the primary catalog structure. If an initial device card is missing, FSAD assumes the permanent structure. The dump can return to the permanent structure if so directed. Except for the assumed case of the permanent structure (or if directive input is null), any device specification with no specific directives following causes an ALL directive to be generated. Specification of the duplicate SMC device allocates that device to the dump directives and the primary SMC device to the structure directives.

A dump can be effected by llink or sector for a specific item or an inclusive range. Dump directives can contain seven octal character llinks or sector addresses with or without leading zeros. The second octal fields specify end range and are optional. The field boundaries for card input are 1-7 and 13-19. A value of less than seven characters may reside anywhere within the field.

The number of llinks examined by a directive is limited, in order to prevent errors in the structure from causing looping. For large structures, patching may be used to increase the limit of llinks to be examined. Normally, FSAD does not use FMS gating to prevent changes in the structure being examined, because this may cause false reports of errors. However, FSAD can use FMS gating by patching calls to the SHUT and OPEN routine from NOP to TSX1.

An optional dump of the AST can be generated when descriptor errors are encountered by making a patch in the ADES routine. The AST dump can be used as system debug information when descriptor errors are reported. Patches for these changes to FSAD can be applied either through the PATCH section of the startup deck or by means of the Extension Editor.

The job in card form consists of the following:

```
1      8       16
$      SNUMB
$      IDENT
$      PROGRAM FSAD
$      PRIVITY
$      FILE DF,....(needed if option 18 is used)
          .
       directives
          .
$      ENDJOB
***    EOF
```

To enter directives from console typewriter, enter card deck with no directives. A descriptive line will be followed by the message:

```
...S#- SNUMB -ENTER ONE OPTION
```

In response enter on console format directive. The ENTER line will repeat until EOM or the directive DONE is entered.

DIRECTIVES

The directives in card format are as follows:

1	8	13	Meaning
nn	OPT		18 - Write descriptor file 19 - Check available space table (AST) of device via .MSXA1 20 - Process for errors 21 - Process for errors and generate a userid directive for current userid of any error.
xxx	DVC		Shift analysis to device XXX
	ALL		Analyze all FMS structures
nn	CLASS		Analyze all FMS structures in section nn
userid (columns 1-12)			Analyze all FMS structures in the specified userid.
OCT	SECT	[OCT]	Dump sector for a specific item or range
OCT	LLINK	[OCT]	Dump llink for a specific item or range

The directives in console typewriter format are the following:

Directives	Meaning
abuts	
O:nn	Same as the OPT directive
D:xxx	Same as the DVC directive
ALL	Same as the ALL directive
C:nn	Same as the CLASS directive
U:userid	Same as the USERID directive
S:OCT[:OCT]	Same as the SECT directive
L:OCT[:OCT]	Same as the LLINK directive
DONE	Terminates the console typewriter directive inputs

INDEX

CUT ALONG LINE

HONEYWELL INFORMATION SYSTEMS
Technical Publications Remarks Form

TITLE	SERIES 60(LEVEL 66)/6000 FILE MANAGEMENT SUPERVISOR	ORDER NO.	DD45, REV. 1
		DATED	AUGUST 1976

ERRORS IN PUBLICATION

SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION

Your comments will be promptly investigated by appropriate technical personnel and action will be taken as required. If you require a written reply, check here and furnish complete mailing address below. □

FROM: NAME ______________________ DATE ____________

TITLE ______________________

COMPANY ______________________

ADDRESS ______________________

____

CUT ALONG LINE

PLEASE FOLD AND TAPE –
NOTE: U. S. Postal Service will not deliver stapled forms

FOLD ALONG LINE

FIRST CLASS
PERMIT NO. 39531
WALTHAM, MA
02154

Business Reply Mail
Postage Stamp Not Necessary if Mailed in the United States

Postage Will Be Paid By:

HONEYWELL INFORMATION SYSTEMS
200 SMITH STREET
WALTHAM, MA 02154

ATTENTION: PUBLICATIONS, MS 486

FOLD ALONG LINE

Honeywell

CUT ALONG LINE

HONEYWELL INFORMATION SYSTEMS
Technical Publications Remarks Form

TITLE SERIES 60(LEVEL 66)/6000 FILE MANAGEMENT SUPERVISOR, ADDENDUM A

ORDER NO. DD45A, REV. 1

DATED APRIL 1977

ERRORS IN PUBLICATION

SUGGESTIONS FØR IMPROVEMENT TO PUBLICATION

Your comments will be promptly investigated by appropriate technical personnel and action will be taken as required. If you require a written reply, check here and furnish complete mailing address below. □

FROM: NAME ____________________ DATE ____________

TITLE ____________________

COMPANY ____________________

ADDRESS ____________________

____

CUT ALONG LINE

PLEASE FOLD AND TAPE –
NOTE: U. S. Postal Service will not deliver stapled forms

FOLD ALONG LINE

FIRST CLASS
PERMIT NO. 39531
WALTHAM, MA
02154

Business Reply Mail
Postage Stamp Not Necessary if Mailed in the United States

Postage Will Be Paid By:

HONEYWELL INFORMATION SYSTEMS
200 SMITH STREET
WALTHAM, MA 02154

ATTENTION: PUBLICATIONS, MS 486

FOLD ALONG LINE

Honeywell

Honeywell

Honeywell Information Systems
In the U.S.A.: 200 Smith Street, MS 486, Waltham, Massachusetts 02154
In Canada: 2025 Sheppard Avenue East, Willowdale, Ontario M2J 1W5
In Mexico: Avenida Nuevo Leon 250, Mexico 11, D.F.

20478, 3478, Printed in U.S.A.