

UNISYS

A Series
Tape Management
**Programming and
Operations Guide**

Release Mark 3.9.2

January 1993

Priced Item

Printed in U S America
8600 0940-000

Product Information Announcement

● New Release ○ Revision ○ Update ○ New Mail Code

Title

A Series Tape Management Programming and Operations Guide

This Product Information Announcement announces the publication of the *A Series Tape Management Programming and Operations Guide* as part of the Mark 3.9.2 System Software Support Release, which will be available in the first quarter of 1993.

The *A Series Tape Management Programming and Operations Guide* documents enhancements to the A Series tape management software.

The guide discusses the following subjects in relation to these enhancements:

- New options for unloading tapes
- One new system command—HOLD (Hold Unit for Operator Use)—and a number of existing system commands that have been modified
- Programming considerations, including the following:
 - Two new file attributes (AUTOUNLOAD and SCRATCHPOOL)
 - One new task attribute (LABELFORMAT) and modifications to two existing task attributes (HISTORYCAUSE and HISTORYREASON)
 - Modifications to the SETSTATUS intrinsics
 - Modifications to the Work Flow Language (WFL)
 - New system log entries and modifications to existing log entries
 - A new tape volume security attribute (SECURITYLABELS)
- Data Management System II (DMSII) considerations, including a new recommendation concerning dump tape directory maintenance and ways of using the SCRATCHPOOL attribute
- TAPEMANAGER library implementation and operations
- Tape label formats

To order additional copies of this document

- United States customers, call Unisys Direct at 1-800-448-1424.
- All other customers, contact your Unisys Sales Office.
- Unisys personnel, use the Electronic Literature Ordering (ELO) system.

Announcement only:

Announcement and attachments:
AS213

System: A Series
Release: Mark 3.9.2 January 1993

Part number: 8600 0940-000

UNISYS

A Series
Tape Management
**Programming and
Operations Guide**

Copyright © 1993 Unisys Corporation.
All Rights Reserved.
Unisys is a registered trademark of Unisys Corporation.

Release Mark 3.9.2

January 1993

Priced Item

Printed in U S America
8600 0940-000

The names, places, and/or events used in this publication are not intended to correspond to any individual, group, or association existing, living, or otherwise. Any similarity or likeness of the names, places, and/or events with the names of any individual, living or otherwise, or that of any group or association, is purely coincidental and unintentional.

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THIS DOCUMENT. Any product or related information described herein is only furnished pursuant and subject to the terms and conditions of a duly executed agreement to purchase or lease equipment or to license software. The only warranties made by Unisys, if any, with respect to the products described in this document are set forth in such agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information in this document or software material, including direct, special, or consequential damages.

You should be very careful to ensure that the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with respect to which it is used.

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

RESTRICTED RIGHTS LEGEND. Use, reproduction, or disclosure is subject to the restrictions set forth in DFARS 252.227-7013 and FAR 52.227-14 for commercial computer software.

Correspondence regarding this publication may be forwarded using the Product Information card at the back of the manual, or may be addressed directly to Unisys, Technical Publications, 25725 Jeronimo Road, Mission Viejo, CA 92691-2792.

Page Status

Page	Issue
iii	-000
iv	Blank
v through xv	-000
xvi	Blank
xvii	-000
xviii	Blank
1-1 through 1-13	-000
1-14	Blank
2-1 through 2-64	-000
3-1 through 3-74	-000
4-1 through 4-7	-000
4-8	Blank
5-1 through 5-60	-000
A-1 through A-37	-000
A-38	Blank
B-1 through B-20	-000
C-1 through C-8	-000
Glossary-1 through 7	-000
Glossary-8	Blank
Bibliography-1 through 2	-000
Index-1 through 27	-000
Index-28	Blank

Unisys uses an 11-digit document numbering system. The suffix of the document number (1234 5678-xyz) indicates the document level. The first digit of the suffix (x) designates a revision level; the second digit (y) designates an update level. For example, the first release of a document has a suffix of -000. A suffix of -130 designates the third update to revision 1. The third digit (z) is used to indicate an errata for a particular level and is not reflected in the page status summary.

About This Guide

Purpose

The *A Series Tape Management Programming and Operations Guide* documents enhancements to the A Series tape management software.

The guide discusses the following subjects in relation to these enhancements:

- New options for unloading tapes
- One new system command – HOLD (Hold Unit for Operator Use) – and a number of existing system commands that have been modified
- Programming considerations, including the following:
 - Two new file attributes (AUTOUNLOAD and SCRATCHPOOL)
 - One new task attribute (LABELFORMAT) and modifications to two existing task attributes (HISTORYCAUSE and HISTORYREASON)
 - Modifications to the SETSTATUS intrinsics
 - Modifications to the Work Flow Language (WFL)
 - New system log entries and modifications to existing log entries
 - A new tape volume security attribute (SECURITYLABELS)
- Data Management System II (DMSII) considerations, including a new recommendation concerning dump tape directory maintenance and ways of using the SCRATCHPOOL attribute
- TAPEMANAGER library implementation and operations
- Tape label formats

Scope

This guide covers only those subjects related to the enhancements to A Series tape management software. It does not deal with aspects of this software that have not changed.

Audience

This guide is intended for use by experienced users of A Series systems, including system administrators, operations center managers, operators, and system programmers.

Prerequisites

The reader of this guide should possess some knowledge of the basics of A Series software and hardware.

In particular, the reader is presumed to be familiar with tape management procedures on his or her system as they relate to his or her responsibilities.

How to Use This Guide

Some readers might choose to use only parts of this guide. However, all readers should begin by reading Section 1, "Overview." This section provides an introductory summary of key points presented in the guide. When you have completed the overview section, refer to subsequent sections for additional information on topics related to your needs.

Much of the information presented in this guide has been extracted from other A Series documents and then revised to reflect the tape management enhancements. For example, Appendix B, "Standard Tape Formats," is based on Appendix G, "Standard Tape Formats," of the *A Series I/O Subsystem Programming Guide*. Section 2, "System Commands," presents revised versions of information from the *A Series System Commands Operations Reference Manual* as well as some text that is entirely new to that book.

When it presents information that is based on another A Series book, the *Tape Management Guide* follows the style and format of the base book. Appendix B is presented in the style and format of the *A Series I/O Subsystem Programming Guide*. Section 2 follows the style and format of the *A Series System Commands Operations Reference Manual*.

Each section (or subsection, as appropriate) of this guide begins with a note that indicates the base book of the text – that is, the book on which the material is based.

Organization

This guide provides an overview section that is intended for all readers, followed by four sections that each discuss a particular aspect of tape management on A Series systems. Appendixes provide related information.

This organization enables the reader to focus on information pertinent to how he or she uses the system. For example, system operators can concentrate on the section that deals with system commands and the appendix that presents new system messages, and avoid the discussions that relate to programming and database management. System administrators might choose to read the entire guide.

In addition, a glossary, a bibliography, and an index appear at the end of this guide.

Section 1. Overview

This section presents highlights of the topics discussed in this guide.

Section 2. System Commands

This section provides detailed information on commands that operators might use to operate the system.

Section 3. Programming Considerations

This section provides detailed information for programmers.

Section 4. DMSII Considerations

This section provides detailed information for users of DMSII.

Section 5. TAPEMANAGER Library Implementation

This section discusses the functions in a TAPEMANAGER library, required procedures of a TAPEMANAGER library, exported master control program (MCP) procedures used in tape management, and operations that interact with the TAPEMANAGER library.

Appendix A. System Messages

This appendix presents system messages that are generated by the MCP as a result of the enhancements to the tape management facilities.

Appendix B. Standard Tape Formats

This appendix presents a number of standard tape formats.

Appendix C. Understanding Railroad Diagrams

This appendix explains how to read the syntax diagrams used in the guide.

Related Product Information

A Series ALGOL Programming Reference Manual, Volume 1: Basic Implementation (8600 0098)

This manual describes the basic features of the Extended ALGOL programming language. This manual is written for programmers who are familiar with programming concepts.

A Series Binder Programming Reference Manual (8600 0304)

This manual describes the functions and applications of the Binder, an efficiency tool that reduces the need to recompile an entire program when only a portion of the program has been modified. This manual is written for programmers who are familiar with programming language concepts and terms.

A Series CANDE Operations Reference Manual (8600 1500)

This manual describes how CANDE operates to allow generalized file preparation and updating in an interactive, terminal-oriented environment. This manual is written for a wide range of computer users who work with text and program files.

A Series DMSII Data and Structure Definition Language (DASDL) Programming Reference Manual (8600 0213)

This manual provides instructions for defining and maintaining a Data Management System II (DMSII) database using DASDL. This manual is written for database administrators and staff.

A Series DMSII Technical Overview (8600 1641)

This overview describes the Data Management System II (DMSII) environment, which includes DMSII databases and software. This overview is written for end users, system programmers, database designers, and database administrators who need information on creating, changing, restoring, accessing, and maintaining a DMSII database.

A Series DMSII Utilities Operations Guide (8600 0759)

This guide shows you how you can use the Data Management System II (DMSII) utility programs. These programs allow you to perform such database management tasks as initialization, reorganization, backup, recovery, verification, and remote database backup. This guide is written for database administrators and programmers who are responsible for database integrity and recovery.

A Series File Attributes Programming Reference Manual (8600 0064)

This manual contains information about each file attribute and each direct I/O buffer attribute. The manual is written for programmers and operations personnel who need to understand the functionality of a given attribute. The *A Series I/O Subsystem Programming Guide* is a companion manual.

A Series GETSTATUS/SETSTATUS Programming Reference Manual (8600 0346)

This manual explains how to use the various GETSTATUS and SETSTATUS calls used in the DCALGOL programming language. This manual is written for experienced ALGOL programmers who are involved with data communications.

A Series I/O Subsystem Programming Guide (8600 0056)

This guide contains information about how to program for various types of peripheral files and how to program for interprocess communication, using port files. This guide is written for programmers who need to understand how to describe the characteristics of a file in a program. The *A Series File Attributes Programming Reference Manual* is a companion manual.

A Series Report Program Generator (RPG) Programming Reference Manual, Volume 1: Basic Implementation (8600 0544)

This manual describes the basic features of the RPG programming language, including both RPG I and RPG II dialects. This manual is written for application programmers who are familiar with programming concepts.

A Series RoboHost A Operation, Installation, and Administration Guide (8600 1740)

This guide describes how to install and operate the RoboHost A TAPESERVER library on an A Series host system. It includes information needed to oversee cartridge tape library (CTL) system operation from the RoboHost A software.

A Series RoboHost A Programming Guide (8600 1609)

This guide provides a detailed definition of the RoboHost A TAPESERVER library functions and is intended as a reference document for developers of tape management systems (TMSs) and user programs that access the cartridge tape library (CTL) database.

A Series Security Administration Guide (8600 0973)

This guide describes system-level security features and suggests how to use them. It provides administrators with the information necessary to set and implement effective security policy. This guide is written for system administrators, security administrators, and those responsible for establishing and implementing security policy.

A Series Security Features Operations and Programming Guide (8600 0528)

This guide describes the security features available to users and provides instructions for their use. This guide is written for users who are responsible for maintaining the security of their individual programs and data.

A Series System Commands Operations Reference Manual (8600 0395)

This manual gives a complete description of the system commands used to control system resources and work flow. This manual is written for systems operators and administrators.

A Series System Messages Support Reference Manual (8600 0429)

This manual presents operating system error messages and explains the most likely cause of each error and the most effective response to each message. This manual is written for operators and programmers responsible for the operation of A Series systems, and for the resolution of error conditions on those systems.

A Series System Software Support Reference Manual (8600 0478)

This manual describes a number of facilities used for system monitoring and debugging, including BARS, DUMPANALYZER, LOGANALYZER, and LOGGER. It also describes the format of the SUMLOG file. This manual is written for system support personnel and operators.

A Series SYSTEMSTATUS Programming Reference Manual (8600 0452)

This manual documents the SYSTEMSTATUS intrinsic of the master control program (MCP). The SYSTEMSTATUS intrinsic provides information that can be used to efficiently monitor the performance of a running system. This manual is written for systems programmers.

A Series Task Management Programming Guide (8600 0494).

This guide explains how to initiate, monitor, and control processes on an A Series system. It describes process structures and process family relationships, introduces the uses of many task attributes, and gives an overview of interprocess communication techniques. The *A Series Task Attributes Programming Reference Manual* is a companion manual.

A Series Work Flow Language (WFL) Programming Reference Manual (8600 1047)

This manual presents the complete syntax and semantics of WFL. WFL is used to construct jobs that compile or run programs written in other languages and that perform library maintenance such as copying files. This manual is written for individuals who have some experience with programming in a block-structured language such as ALGOL and who know how to create and edit files using CANDE or the Editor.

Contents

About This Guide	v
Section 1. Overview	
Using Scratch Pools	1-2
Using the Scratch Pool Concept in System Operations .	1-2
Using the Scratch Pool Concept in Programs	1-3
Using the Scratch Pool Concept in DMSII	1-5
Controlling Automatic Tape Unloading	1-5
Controlling Automatic Unloading through System Operations	1-6
Controlling Automatic Unloading through Programs. . .	1-6
Examining a Tape before It Is Assigned	1-7
Purging Tapes Synchronously	1-8
Storing Security Attribute Information in Tape Labels	1-8
Using the ANSI87 Tape Label Format	1-9
Controlling Tape Label Formats with System Commands	1-9
Controlling Tape Label Formats through Programs. . . .	1-10
Ignoring Tape Expiration Dates	1-10
Using New System Log Entries and Enhanced Existing Log Entries	1-11
Implementing Customized Tape Management Systems	1-12
Calling Standard Exported Procedures	1-13
Controlling Tape Usage	1-13
Installing a TAPEMANAGER Library	1-13
Section 2. System Commands	
HOLD (Hold Unit for Operator Use)	2-2
MODE (Unit Mode)	2-4
OL (Display Label and Paths)	2-8
OP (Options)	2-21
OU (Output Unit)	2-33
PER (Peripheral Status)	2-37
PG (Purge) and PGL (Purge and Lock)	2-43
SEND (Send Message)	2-47
SN (Serial Number) and SNL (Serial Number Lock) . .	2-56
SYSOPS (System Options)	2-59
Section 3. Programming Considerations	
File Attributes	3-1
AUTOUNLOAD	3-2
SCRATCHPOOL	3-5

Task Attributes	3-6
LABELFORMAT	3-7
HISTORYCAUSE	3-9
HISTORYREASON	3-11
SETSTATUS Changes	3-30
SEND TAPEMANAGER and SEND TAPESERVER Calls	3-31
HOLD Call	3-33
PG and PGL Calls	3-35
SN and SNL Calls	3-39
Security Enhancements	3-42
SECURITYLABELS Volume Attribute	3-42
Establishing the Volume Directory Entry	3-42
Writing Files to the Volume	3-42
Reading from the Volume	3-43
ANSI87 Tape Format	3-43
Tape Management Enhancements in WFL	3-43
Using File Attributes	3-44
ARCHIVE Statements	3-45
ARCHIVE Statement Options	3-47
COPY or ADD Statement	3-47
Specifying File and Tape Volume Attributes	3-50
VOLUME Statement	3-51
Logging Enhancements	3-53
LOGANALYZER Output Illustration for VOLUMES	
Option	3-54
Major Type 1–Minor Type 5: File Open Entry	
(LOGOPENTYPE)	3-56
Major Type 1–Minor Type 6: File Close Entry	
(LOGCLOSETYPE)	3-60
Major Type 2–Minor Type 24: CLU Hardware Report	
(CLUHWREPORT)	3-67
Major Type 6–Minor Type 3: SETSTATUS Call	
(LOGSETSTAT)	3-68
Major Type 15–Minor Types 1–5, 12, 13: Volume	
Status Entry (LOGMAJVOL)	3-70
Major Type 26: Tape Library Entry	
(LOGMAJ_MEDIALIB)	3-74

Section 4. DMSII Considerations

Tape Management Enhancements in the Technical Overview ...	4-1
Tape Management Enhancements in DMSII Utilities	4-2
Table of Contents	4-2
DMCONTROL Statement	4-2
DMUTILITY DUMP Statement	4-3
COPYAUDIT Commands	4-4
VISIBLE DBS Commands	4-6
AUDIT SCRATCHPOOL Command	4-6

Section 5. TAPEMANAGER Library Implementation

TAPEMANAGER Functions for TMS Design	5-3
Event and Status Information Retrieval for TMS Design	5-4
Monitoring Tape Usage Events	5-5
Monitoring Current Tape Status	5-6
Initiating Programmatic Tape Operations	5-6
CALL_TAPEMANAGER Procedure	5-7
General Considerations for TAPEMANAGER Libraries	5-7
TAPEMANAGER Error Handling	5-8
Fault Handling within TAPEMANAGER	5-9
Error Handling by the MCP	5-9
Security Considerations	5-10
Interaction with the MCP	5-11
TAPEMANAGER Library Start Up and Shut Down	5-11
Starting Up	5-12
Shutting Down	5-14
Opening Files	5-15
CHECK_TAPE_ASSIGNMENT	5-15
CHECK_TAPE_UNIT_ASSIGNMENT	5-18
CHECK_TAPE_REOPEN	5-20
Purging Tape Volumes	5-21
Handling Programmatic Calls to the TAPEMANAGER Library	5-23
Exported TAPEMANAGER Procedures	5-23
CHECK_TAPE_ASSIGNMENT	5-25
CHECK_TAPE_PURGE	5-30
CHECK_TAPE_REOPEN	5-32
CHECK_TAPE_UNIT_ASSIGNMENT	5-34
CONTROL_INTERFACE	5-37
PROGRAM_INTERFACE	5-39
Sample TAPEMANAGER Library	5-41
Exported MCP Procedures for Tape Management	5-52
CALL_TAPEMANAGER Procedure	5-53
Operations Interface to the TAPEMANAGER Library	5-55
Installing the TAPEMANAGER Library	5-55
Initiating the TAPEMANAGER Library	5-56
Inspecting the TAPEMANAGER Status	5-56
Terminating the TAPEMANAGER Library	5-57
Temporarily Disabling Use of the TAPEMANAGER Library	5-58
Responding to Processes Waiting on the TAPEMANAGER Library	5-59
Replacing the TAPEMANAGER Library with a Newer Version	5-60

Appendix A. System Messages

Appendix B. Standard Tape Formats

Unlabeled Tapes	B-1
ANSI X3.27-1969 (ANSI69) Tapes	B-2
ANSI X3.27-1987 (ANSI87) Tapes	B-8
B 3500 USASI Tapes	B-17
B 5500 Tapes	B-19
ALGOL Files	B-20
COBOL Files	B-20

Appendix C. Understanding Railroad Diagrams

What Are Railroad Diagrams?	C-1
Constants and Variables	C-2
Constraints	C-2
Following the Paths of a Railroad Diagram	C-5
Railroad Diagram Examples with Sample Input	C-6

Glossary	1
----------------	---

Bibliography	1
--------------------	---

Index	1
-------------	---

Figures

3-1.	VOLUMES Option Output	3-55
5-1.	The Tape Management System (TMS) Environment	5-2
5-2.	TAPEMANAGER Library Status Change Diagram	5-12
5-3.	Information Flow in CHECK_TAPE_ASSIGNMENT	5-17
5-4.	Information Flow in CHECK_TAPE_UNIT_ASSIGNMENT	5-19
5-5.	Information Flow in CHECK_TAPE_REOPEN	5-21
5-6.	Information Flow in CHECK_TAPE_PURGE	5-22
B-1.	Unlabeled Single-File-Volume and Unlabeled Multifile-Volume Formats	B-1
B-2.	ANSI69 Single-File, Single-Volume Format	B-2
B-3.	ANSI69 Multivolume-File and Multifile-Volume Formats	B-3
B-4.	ANSI69 Multifile, Multivolume Format	B-4
B-5.	ANSI69 Volume Header Formats	B-5
B-6.	ANSI69 File Header 1 Format	B-6
B-7.	ANSI69 File Header 2 Format	B-7
B-8.	ANSI69 User Header and Trailer Label Formats	B-8
B-9.	ANSI69 Scratch Tape Format	B-8
B-10.	ANSI87 Multivolume-File and Multifile-Volume Formats	B-9
B-11.	ANSI87 Multifile, Multivolume Format	B-10
B-12.	ANSI87 Volume Header 1 Format	B-11
B-13.	ANSI87 Volume Header 2 and Volume Header 3 Formats	B-12
B-14.	ANSI87 File Header 1 Format	B-14
B-15.	ANSI87 File Header 2 Format	B-15
B-16.	ANSI87 File Header 3 Format	B-16
B-17.	ANSI87 Scratch Tape Format	B-17
B-18.	B 3500 Volume Header Format	B-17
B-19.	B 3500 File Header 1 Format	B-18
C-1.	Railroad Constraints	C-5

Tables

2-1.	AUTOUNLOAD Option Effects	2-6
3-1.	Comparison of File Attribute Types to WFL Attribute Types	3-44
3-2.	Major Type 1–Minor Type 5	3-56
3-3.	Major Type 1–Minor Type 6	3-60
3-4.	Major Type 6–Minor Type 3	3-68
3-5.	Major Type 15–Minor Types 1 through 5, 12, and 13	3-70
4-1.	Tape Management Enhancements to DMSII	4-1
5-1.	Log Entries for Tape Management	5-5
5-2.	Exported TAPEMANAGER Procedures	5-24
A-1.	Metatokens Appearing in Message Listings	A-2
B-1.	BCL Characters for B 5500 Tape Labels	B-19
B-2.	B 5500 Tape Label for ALGOL Files	B-20
B-3.	B 5500 Tape Label for COBOL Files	B-20

Section 1

Overview

This guide presents a number of enhancements to the A Series software. These enhancements provide additional flexibility in the ways installations can manage their data that is stored on tape.

The enhanced tape management facilities enable you to perform the following operations:

- Logically group scratch tapes in scratch pools, each with a unique identifier. A *scratch pool* is a group of tapes that have been given a name. This capability means that when you create a new tape volume, you can provide either of the following types of instructions concerning the tape to be used:
 - Direct the system to use a particular tape by supplying a specific serial number.
 - Specify that the system can use any tape from a given scratch pool.
- Exercise greater control over how file-close operations affect tape drives. Specifically, you can designate which of the following actions the system is to take:
 - Rewind the tape.
 - Rewind the tape and then unload it.
- Examine a tape before the master control program (MCP) assigns it to a task.
- Maintain the settings of tape volume security attributes in the tape label rather than in the volume directory so that you can more easily transfer the tape among hosts in a multihost environment. This enhancement is available only on systems with the InfoGuard security enhancement software for A Series systems.
- Read and write tapes in ANSI X3.27 1987 (ANSI87) format as well as tapes in ANSI X3.27 1969 (ANSI69) format.
- Read tapes in other formats supported by the A Series software.
- Specify that expiration dates read from tape labels are to be ignored. (By default, the MCP treats a write-enabled tape as a scratch tape if its expiration date has passed.)
- Use modified versions of system log entries.
- Use enhanced tape management interfaces.

The remainder of this section describes these capabilities and the means by which you can invoke them. Subsequent sections provide detailed documentation.

Using Scratch Pools

Scratch pools enable you to group scratch tapes under a common identifier. Rather than supplying an individual serial number to identify a particular tape, you can specify a scratch pool name to enable the system to request or select any tape from that scratch pool.

You can create scratch pools according to any criteria you choose. For example, you could create a scratch pool for each of the following types of tapes:

- Tapes of a given media type
- Tapes whose labels specify a given account number
- Dump tapes
- Archive tapes

You can specify a scratch pool in system commands, in programs, and in Data Management System II (DMSII) utilities. The scratch pool identifier appears in the RSVP request issued to the operator.

A scratch pool name contains from 1 to 17 alphanumeric characters.

Using the Scratch Pool Concept in System Operations

System operators can use the scratch pool concept in a variety of situations including the following:

- To assign a tape to a scratch pool, an operator can provide a scratch pool name as part of the tape-purging process.
- To request a tape from a specific scratch pool, an operator can specify the scratch pool name in the WFL statements COPY or ADD and ARCHIVE.
- An operator can override an earlier scratch pool assignment with a file equation.

The following system commands are affected by the scratch pool concept:

- OU (Output Unit)
- PER (Peripheral Status)
- PG (Purge) and PGL (Purge and Lock)
- SN (Serial Number) and SNL (Serial Number Lock)

To purge a tape and assign it to a scratch pool, use a variation of the PG (Purge) system command. To purge a tape, assign or change its serial number, and assign it to a scratch pool, use the SN (Serial Number) system command.

See Section 2, "System Commands," for additional details on the means by which operators can create and point to scratch pools in system commands. See "Tape Management Enhancements in WFL" in Section 3 for information on how to specify scratch pools in WFL statements.

Examples

The following examples illustrate how operators can assign tapes to scratch pools.

- To purge a tape on unit 110 and assign the tape to a scratch pool named ACCOUNTING, enter the following PG (Purge) system command:

```
PG MT 110 SCRATCHPOOL = ACCOUNTING
```

- To purge a tape on unit 65 and assign the tape to a scratch pool named BACKUP, enter the following PG command:

```
PG MT 65 POOL = BACKUP
```

- To purge a tape on unit 34, assign the serial number PART1 to that tape, and assign the tape to a scratch pool named MANUFACTURING, use the following SN (Serial Number) system command:

```
SN MT 34 PART1 POOL = MANUFACTURING
```

- To purge a tape on unit 114, assign the serial number 12345 to that tape, and assign the tape to a scratch pool named DEVELOPMENT, use the following SN command:

```
SN MT 114 12345 SCRATCHPOOL = DEVELOPMENT
```

The following examples illustrate how an operator can request a tape from a scratch pool.

- To copy the file PAYABLES from disk PACK to any tape in the ACCOUNTING scratch pool, enter the following WFL COPY statement:

```
COPY PAYABLES AS PAYABLES FROM PACK TO ACCOUNTING  
(KIND = TAPE, SCRATCHPOOL = ACCOUNTING)
```

- To override an earlier specification that the tape output from a particular process be written to the RESEARCH scratch pool, enter the following OU (Output Unit) system command to direct output from that process to whatever tape is currently mounted on unit 116:

```
<mix number> OU MT 116
```

Using the Scratch Pool Concept in Programs

Programmers can use the scratch pool concept in a variety of situations:

- By specifying the SCRATCHPOOL file attribute in file-assignment statements to ensure that when a tape file is created, the physical tape used to store that file is from a specific scratch pool.
- By specifying the SCRATCHPOOL file attribute in the Work Flow Language (WFL) statements COPY or ADD and ARCHIVE. This file attribute changes the syntax of only the WFL programming language.

- By specifying the SCRATCHPOOL file attribute in the DUMPALL utility COPY command.
- By defining a scratch pool for the destination of a task request to the FILECOPY utility.

Examples

The following examples illustrate how programmers can use the scratch pool concept.

- The following ALGOL statement specifies that when a file is opened and a new tape is to be written, the output is to be written to a tape from the ACCOUNTING scratch pool:

```
REPLACE F.SCRATCHPOOL BY "ACCOUNTING.";
```

This command takes effect when the system is ready to write tape output. The results of the command depend on the circumstances existing at that time, as follows:

- If a tape from the ACCOUNTING scratch pool is currently mounted, the file is assigned to that tape.
- If no tape is currently mounted, or if none of the currently mounted tapes is from the ACCOUNTING scratch pool, the system displays an RSVP request (MSRFOUT30) on the operator display terminal (ODT). The following is an example of this request:

```
F REQUIRES MT #1 SCRATCHPOOL = ACCOUNTING
```

The operator can then perform either of the following actions:

- Mount a tape that has been assigned to the ACCOUNTING scratch pool.
- Purge a tape and assign it to the ACCOUNTING scratch pool as described and illustrated in "Using the Scratch Pool Concept in System Operations" later in this section.

See Appendix A, "System Messages," for information on MSRFOUT30 and other messages generated by the MCP that deal with the tape management facilities.

- The following WFL statement copies the file X from disk FRONT to tape BACK in the TEST scratch pool:

```
COPY X FROM FRONT (PACK) TO BACK (SCRATCHPOOL=TEST)
```

- The following specification in a DUMPALL request copies the file FILEONE to tape TAPE/FILEONE in the DEFAULTTAPE scratch pool:

```
COPY FILEONE TO TAPE/FILEONE  
(KIND = TAPE, SCRATCHPOOL = DEFAULTTAPE)
```

See Section 3, "Programming Considerations," for additional information on the means by which programs can manipulate scratch pools.

Using the Scratch Pool Concept in DMSII

Users of the Data Management System II (DMSII) can use the SCRATCHPOOL file attribute instead of the SERIALNO option when they back up a database to tape.

DMSII users can also use the AUDIT SCRATCHPOOL command in the VISIBLE DBS utility to change the SCRATCHPOOL attribute for audit tape options.

Examples

The following examples illustrate how users can designate scratch pools in DMSII.

- The following DMUTILITY command dumps the database to a tape in the TEST scratch pool:

```
OFFLINE DUMP = TO TAPEX (SCRATCHPOOL = TEST)
```

- The following COPYAUDIT command copies the audit file TESTDB/AUDIT1 from pack USER to a tape in the ACCOUNTING scratch pool:

```
COPY TESTDB/AUDIT1 ALL FROM PACK = USER TO TAPE SCRATCHPOOL = ACCOUNTING
```

See Section 4, "DMSII Considerations," for additional details on how DMSII users can manipulate scratch pools.

Controlling Automatic Tape Unloading

You can control the actions taken when the MCP either releases a tape drive from use or performs a reel switch from the unit. By default, the tape is rewound but not unloaded in both cases.

You can override this default procedure by invoking *automatic-unload mode* as described in the following paragraphs. When automatic-unload mode is in effect, the tape drive rewinds the tape. The system then automatically unloads the tape unless the unit is a quarter-inch cartridge. (Since a quarter-inch tape drive does not physically eject the tape, the tape drive unwinds and untenses the tape so that the operator can unload it.)

The following methods enable you to control when automatic-unload mode is or is not in effect:

- You can use the MODE (Unit Mode) AUTOUNLOAD system command to define the automatic-unload mode of a particular drive as either ON or OFF.
- At run time, you can use the AUTOUNLOAD file attribute to define the automatic-unload mode for a file as ON or OFF, either to override a previous MODE AUTOUNLOAD command or to unconditionally force automatic tape unloading. The AUTOUNLOAD file attribute is valid in the WFL statements COPY or ADD and ARCHIVE and in file-equation statements.

- In addition to ON or OFF, you can assign the value DONTCARE to the AUTOUNLOAD file attribute. This value causes the behavior of tape file close operations to be defined by the most recent MODE AUTOUNLOAD system command for the drive. DONTCARE is the default value of the attribute.

See Section 3, “Programming Considerations,” for additional information on the MODE AUTOUNLOAD system command and the AUTOUNLOAD file attribute.

Controlling Automatic Unloading through System Operations

An operator can control automatic unloading of tapes by using the MODE (Unit Mode) system command in the form *MODE MT AUTOUNLOAD*. This command sets the automatic-unload mode of a given tape drive to either ON or OFF. The default setting is OFF.

The automatic-unload mode setting of a drive is preserved across halt/loads, but can be overridden by the AUTOUNLOAD file attribute value of a program or library maintenance task. See “Controlling Automatic Unloading through Programs,” next in this discussion.

Examples

The following examples illustrate the MODE system command.

- To define the automatic-unload mode of drive 9 as OFF, enter the following command:

```
MODE MT 9 AUTOUNLOAD OFF
```

- To set the automatic-unload mode of drive 6 to ON, enter the following command:

```
MODE MT 6 AUTOUNLOAD ON
```

See Section 2, “System Commands,” for additional information on the means by which operators can manipulate the automatic unloading of tapes.

Controlling Automatic Unloading through Programs

Programmers can use the automatic unloading capability in the following ways:

- By specifying the AUTOUNLOAD file attribute in a program to explicitly invoke automatic unloading of a tape as follows:

AUTOUNLOAD File Attribute Value	Action
OFF	Tape rewound but not unloaded
ON	Tape rewound and unloaded
DONTCARE	Depends on whether automatic-unload mode has been set to ON by a MODE (Unit Mode) system command, described earlier in this discussion

If the AUTOUNLOAD file attribute conflicts with the AUTOUNLOAD mode value, the file attribute takes precedence.

- By specifying the AUTOUNLOAD file attribute in any of the WFL statements COPY, ADD, or ARCHIVE.
- By specifying AUTOUNLOAD in defining the destination of a task request to the FILECOPY utility.

Examples

The following examples illustrate how programmers can use the automatic unloading capability.

- The following WFL COPY statement copies the file X from disk PACK as the file XX on tape T. The AUTOUNLOAD attribute specifies that the tape is to be rewound and unloaded.

```
COPY X AS XX FROM PACK TO T(AUTOUNLOAD=ON)
```

- The following specification in a task request to the FILECOPY utility invokes automatic rewinding and unloading of the tape.

```
FILES (= FROM DISK) TO WEEK36 (SCRATCHPOOL=BACKUP,AUTOUNLOAD=ON)
```

See Section 3, "Programming Considerations," for additional details on the means by which programs can manipulate the automatic unloading of tapes.

Examining a Tape before It Is Assigned

HOLD (Hold Unit for Operator Use) is a new system command that enables you to mount and look at a physically unlabeled tape before the MCP can assign it to a task. This command enables you to decide what you want to do with the tape before a task can assign it. That is, you can examine the label, purge the tape, and so on, before a program uses the tape.

You can also use the HOLD command to remove the HOLD restriction.

Examples

The following examples illustrate the HOLD command.

- To hold a tape that is located on unit 15, enter the following command:

```
HOLD MT 15
```

- To remove the HOLD restriction, enter the following command:

```
HOLD - MT 15
```


Purging Tapes Synchronously

When you use a PG (Purge), PGL (Purge and Lock), SN (Serial Number), or SNL (Serial Number Locked) call to the SETSTATUS intrinsic, you can specify that the tape is to be purged synchronously with the calling task—that is, that the task is to wait for the operation to complete before proceeding.

The advantage to a synchronous tape purge is that the success or failure of the call is returned to the task, which can then proceed appropriately, based on that outcome. The disadvantage is that processing of the task is interrupted until the success or failure of the call is returned.

See “PG and PGL Calls” and “SN and SNL Calls” in Section 3 for information on the parameter bit that controls whether these calls purge the tape synchronously or asynchronously.

Storing Security Attribute Information in Tape Labels

A new volume attribute, SECURITYLABELS, enables users to maintain the settings of security attributes in the tape label rather than in the volume directory. When this information is on the tape itself, the tape is more easily transferred among hosts in a multihost environment.

The SECURITYLABELS attribute is controlled by the Work Flow Language (WFL) *VOLUME* statement. The following criteria must be met for this option to take effect:

- The InfoGuard security subsystem must be activated on the system.
- The CLASS system option must have a value other than S2.

The SECOPT (Security Options) system command controls the CLASS option. To use this command, a user must have security administrator status. For information on the SECOPT command and the CLASS option, see the *A Series Security Administration Guide*.

SECURITYLABELS is a Boolean attribute. When it has a value of TRUE, the values of the following attributes are maintained in the tape label: SECURITYTYPE, SECURITYUSE, SECURITYGUARD, and FAMILYOWNER.

You control the setting of the SECURITYLABELS attribute with the WFL *VOLUME* statement. See “Tape Management Enhancements in WFL” in Section 3.

For additional information on the SECURITYLABELS volume attribute, see “Security Enhancements,” also in Section 3.

Using the ANSI87 Tape Label Format

A Series systems can read and write tapes in ANSI X3.27 1987 (ANSI87) format as well as in ANSI X3.27 1969 (ANSI69) and other tape formats. Appendix B, “Standard Tape Formats,” illustrates the standard tape label formats supported by the A Series software.

You can define which label format is to be used when a tape is first opened for output. The format used for the first file on the tape defines the format for the entire volume set. After a file has been written to the tape, the system ignores any label-formatting specifications supplied by the user.

The following tools enable you to control how a tape label is formatted:

- LABELFORMAT system option
- LABELFORMAT task attribute

The default value of the LABELFORMAT system option is ANSI69. If no value is defined for the LABELFORMAT task attribute, the system uses the current LABELFORMAT system option value as the default task attribute value.

See “Task Attributes” and “Security Enhancements,” both in Section 3, for details on these tools.

Notes:

- *If the SECURITYLABELS volume attribute is TRUE for a tape, the system uses the ANSI87 standard when writing to that tape. See “Security Enhancements” in Section 3 for information on the SECURITYLABELS volume attribute.*
- *A Series software at the Mark 3.9.1 or earlier release level is incapable of reading tapes written in ANSI87 format.*

Controlling Tape Label Formats with System Commands

An operator can use the SYSOPS (System Options) system command to control the setting of the LABELFORMAT system option as follows:

```
SYSOPS LABELFORMAT = ANSI87DEFAULT
```

Including this form of the option in a SYSOPS command specifies that all subsequently written tape labels are to be written in ANSI87 format. If the value of the LABELFORMAT option is ANSI69 (the default value of the option), the system writes tape labels in ANSI69 format.

All options set with the SYSOPS command retain their values across halt/loads.

For additional information on the SYSOPS command, see Section 2, “System Commands.”

Controlling Tape Label Formats through Programs

A programmer can use the LABELFORMAT task attribute to define a format to be used when he or she creates new tape volumes as follows:

LABELFORMAT Task Attribute Value	Format Used
ANSI69	ANSI X3.27-1969
ANSI87	ANSI X3.27-1987
UNSPECIFIED	Format defined by the LABELFORMAT system option

You can use this attribute either to override an earlier specification – made with the LABELFORMAT system option or a previous LABELFORMAT task attribute – or to unconditionally request a particular label format.

For additional information on the LABELFORMAT task attribute, see Section 3, “Programming Considerations.”

Ignoring Tape Expiration Dates

Whenever it reads a tape in order to write data to it, the MCP reads the expiration date from the tape label. If the expiration date has passed, the MCP acts according to whether the system option TAPEEXPIRATION is currently TRUE or FALSE, as follows:

TAPEEXPIRATION Option Setting	Action Taken by MCP if Expiration Date Has Passed
TRUE (default)	Treats the tape as a scratch tape
FALSE	Ignores the expiration date of the tape

The MCP identifies a tape that is past its expiration date as a scratch tape only if it is write-enabled. The data on the tape is not actually purged; a tape that is past its expiration date can still be read if the tape is not write-enabled. On reel-to-reel tape drives, a tape is write-enabled if it has a write ring in place. A cartridge tape is write-enabled if its file protection mechanism is set to the unprotected position. You set a cartridge tape file protection mechanism in one of the following ways, depending on the type of cartridge you are using:

- On 5073 cartridges, use your thumb to roll the small knurled knob on the side of the cartridge until the rounded, unmarked surface appears.
- On Unisys HD600XTD or HD6150 or equivalent tape cartridges for quarter-inch cartridge tape drives, position the write-protect plug on the top so that the arrow points away from the embossed word *SAFE*.

Consult the documentation for your cartridge drives for additional information.

An operator can use the SYSOPS (System Options) system command to control the setting of the TAPEEXPIRATION option as follows:

- To request that the option be activated, include any of the following forms of the option in a SYSOPS command:
 - TAPEEXPIRATION +
 - TAPEEXPIRATION ON
 - TAPEEXPIRATION TRUE
- To request that the option be deactivated – that is, returned to its default status (not active) – include any of the following forms of the option in a SYSOPS command:
 - TAPEEXPIRATION –
 - TAPEEXPIRATION OFF
 - TAPEEXPIRATION FALSE

The default value for the option is TRUE.

See Section 2, “System Commands,” for additional information on the SYSOPS command.

Note: A requested change to the setting of this option does not take effect immediately; it goes into effect the next time the MCP reads the tape label.

Using New System Log Entries and Enhanced Existing Log Entries

The enhancements to the tape management software include new entry points to the system log (SUMLOG) and changes to existing log entries, as follows:

- The File Open log entry (LOGOPENTYPE) has been enhanced to record whether the TAPEMANAGER library changed any file attributes of a tape file before the file was opened. This entry now records additional error message numbers.
- The File Close log entry (LOGCLOSETYPE) now records additional error message numbers.
- The SETSTATUS call entry (LOGSETSTAT) has been enhanced to store an additional SETSTATUS value.
- The Volume Status entries (LOGMAJVOL) have been enhanced in the following areas:
 - The new Minor Type 12 (Tape Volume Used) entry is issued when an existing tape volume is assigned to a program.
 - The new Minor Type 13 (Tape Volume Hold) entry is issued when a HOLD (Hold Unit for Operator Use) system command changes the HOLD status of a tape unit.

- The Minor Type 5 (Tape Volume NEWFILE) entry is now issued for all cases in which an unlabeled tape file or a labeled tape file with NEWFILE = TRUE is assigned to a tape volume.
- The HOLD status of the volume is now included.
- Minor Type 3 (Tape Volume Purged) log entries now store the scratch pool name in EBCDIC characters.
- A new log entry minor type – CLU Hardware Report (CLUHWREPORT) – records Cartridge Tape Library (CTL) hardware activity (for RoboHost A users only).
- The new Major Type 26 – Tape Library Entry (LOGMAJ_MEDIALIB) – documents TAPESERVER library entries (for RoboHost A users only).

For more information on these entries, see “Logging Enhancements” in Section 3. The new system log (SUMLOG) entries for RoboHost A users are described in the *RoboHost A Operations, Installation, and Administration Guide*.

Implementing Customized Tape Management Systems

A *tape management system (TMS)* is any application that is designed to record or control the usage of tapes on a system.

The A Series system provides enhanced operating system support for user-created TMSs. Your site can make use of these enhancements by following these guidelines:

- You must implement as part of your TMS a support library with the function name TAPEMANAGER.
- You must provide certain exported procedures in the TAPEMANAGER library.

Whenever certain tape accesses occur, the operating system calls procedures from the TAPEMANAGER library. The library can then respond to each request in any of the following ways:

- Perform the requested action.
- Deny the request.
- Modify the request.
- Record the request.

The TAPEMANAGER library is entirely optional.

The following paragraphs provide introductory information on using these entry points for your installation. For details, see Section 5, “TAPEMANAGER Library Implementation.”

Calling Standard Exported Procedures

The standard exported procedures of a TAPEMANAGER library are as follows:

- **CHECK_TAPE_ASSIGNMENT**
This procedure determines what happens when a process attempts to open a tape file that has not yet been assigned to a tape volume, and for which the UNITNO file attribute has not yet been assigned. The procedure also supplies or modifies file attribute values to further specify the desired tape.
- **CHECK_TAPE_UNIT_ASSIGNMENT**
This procedure determines what happens when a process attempts to open a new tape file that has not yet been assigned to a tape volume, and for which the UNITNO file attribute has been assigned to a tape unit number.
- **CHECK_TAPE_PURGE**
This procedure determines whether to allow or deny a request to purge a tape.
- **CHECK_TAPE_REOPEN**
This procedure determines whether to allow or deny a request to reopen a tape file previously closed with retention.
- **CONTROL_INTERFACE**
This procedure determines whether to allow or deny a request to initialize, terminate, or communicate with the TAPEMANAGER library.
- **PROGRAM_INTERFACE**
The purpose of this procedure is defined by the library implementation.

Controlling Tape Usage

Following are examples of the types of control a TMS can exert over tape usage:

- Denying a process the right to open a given tape file
- Modifying the attributes of a tape file before the file open operation completes
- Denying a process, or an operator, the right to purge a given tape volume

Installing a TAPEMANAGER Library

See “Operations Interface to the TAPEMANAGER Library” in Section 5 for a discussion of the steps involved in installing, terminating, and inspecting the status of a TAPEMANAGER library.

Section 2

System Commands

Computer operators need to be aware of one new system command and changes to several existing system commands to effectively use the new tape management operation features.

The HOLD (Hold Unit for Operator Use) command is new.

The following commands have been modified to support new tape management features:

- MODE (Unit Mode)
- OL (Display Label and Paths)
- OP (Options)
- OU (Output Unit)
- PER (Peripheral Status)
- PG (Purge) and PGL (Purge and Lock)
- SEND (Send Message)
- SN (Serial Number) and SNL (Serial Number Lock)
- SYSOPS (System Options)

Note: *The information in this section is based on the A Series System Commands Operations Reference Manual.*

HOLD (Hold Unit for Operator Use)

HOLD (Hold Unit for Operator Use)

The HOLD command either prevents the system from assigning files to a unit or allows the system to assign files to a unit that was previously the object of a HOLD command.

The HOLD command is valid only for magnetic tape units.

Syntax

```
— HOLD — MT —<unit number list> —
```

Explanation

A unit that is held aside for operator use cannot be assigned to any file, but the unit is still accessible to the system. Unit-related operator commands are not affected by the HOLD status of a unit. HOLD status is lost if the unit is the subject of a FREE command. Because the unit cannot be assigned to any command. Because the unit cannot be assigned to any file, an attempt to assign it by using an IL, UL, or OU command is not effective.

Examples

```
HOLD MT 10
```

```
MT 10 FILE ASSIGNMENT HOLD IN EFFECT
```

```
PER MT
```

```
10 SCRATCH (HOLD)
```

```
HOLD - MT 10
```

```
MT 10 FILE ASSIGNMENT NOT IN EFFECT
```

Considerations for Use

If you want to acquire a tape unit and prevent file assignment so that you can examine a tape label, perform the following steps:

1. Acquire the unit with the SAVE option by entering *ACQUIRE MT <unit number>: SAVE*.
2. Enter the HOLD command to prevent the system from assigning files to the unit.
3. Enter the RY command so that the system can read the tape label. command so that the system can read the tape label.
4. Enter the PER MT command so that you can read the name, status and serial number of the tape.

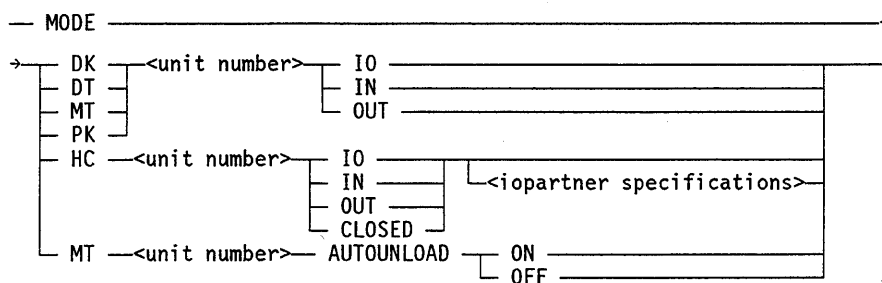
Because the command TDIR <unit number> initiates the FILEDATA utility, which tries to open the tape on the specified unit, you cannot use that command on a tape drive for which you entered a HOLD command.

MODE (Unit Mode)

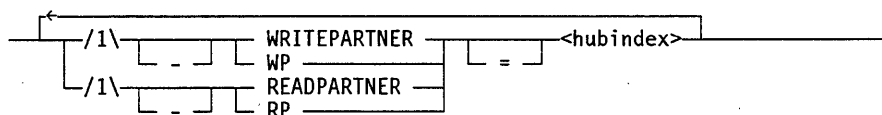
MODE (Unit Mode)

The MODE (Unit Mode) command tells the system what you want the write-enable status of a unit to be. This command also changes the AUTOUNLOAD status.

Syntax



<iopartner specifications>



The hubindex is an integer in the range 0 through 15.

Explanation

MODE DK <unit number> IO
 MODE DT <unit number> IO
 MODE MT <unit number> IO
 MODE PK <unit number> IO

Allow the device to revert to normal operation.

MODE DK <unit number> IN
 MODE DT <unit number> IN
 MODE MT <unit number> IN
 MODE PK <unit number> IN

Prevent new files from being created on the specified unit. This prohibition extends to new RESERVEDISK and BADDISK files, so that the RESERVE and XD functions are effectively blocked. These commands also prevent the allocation of new file areas to existing files, and file removal through the WFL command *REMOVE A/B, C/= FROM D*. However, user programs can still write to existing file areas, and can remove individual files through the CLOSE with PURGE command.

MODE DK <unit number> OUT
MODE DT <unit number> OUT
MODE MT <unit number> OUT
MODE PK <unit number> OUT

Allow the device to revert to normal operation.

MODE HC <unit number> IO

Allows the specified host control (HC) unit to be opened for input/output (I/O) use. It is valid only for HC-2 units (DLP-based I/O).

MODE HC <unit number> IN

Allows the specified HC unit to be opened for input (read operations) only.

MODE HC <unit number> OUT

Allows the specified HC unit to be opened for output (write operations) only.

MODE HC <unit number> CLOSED

Disallows HCFILE use of the specified HC unit. BNA Version 1 can use HC units that are closed, changing the mode of use as needed.

MODE MT <unit number> AUTOUNLOAD ON

Instructs the MCP to unload a tape on the device when the unit is closed and released or when the MCP performs a reel switch from the unit.

MODE MT <unit number> AUTOUNLOAD OFF

Instructs the MCP to unload a tape on the device only when specifically directed by a programmatic close or by the RW (Rewind) system command.

WRITEPARTNER = <hubindex>
WP = <hubindex>

Cause all HCFILE write operations on the specified HC unit to be directed to the specified hubindex.

READPARTNER = <hubindex>
RP = <hubindex>

Cause HCFILE read operations from the specified HC unit to expect input only from the hubindex.

- WRITEPARTNER
- WP
- READPARTNER
- RP

Remove WRITEPARTNER or READPARTNER specifications from the specified HC unit.

Examples

MODE HC 096 OUT

HC096 MODE IS OUT

MODE MT 48 AUTOUNLOAD OFF

MT 48 MODE IS AUTOUNLOAD OFF

Considerations for Use

If applied to HC units, the MODE command controls the direction of the I/O as well as specifying partner HC units to which I/O is permitted.

Direction can be specified only for a closed HC; this is the default state of an HC unit when it is not being used by BNA Version 1.

The IOPARTNER specifications can be used only for labeled HC units. The minus sign (-) syntax removes an I/O partner restriction from a labeled HC unit and causes any related hubindex to be ignored. A null READPARTNER or WRITEPARTNER specification is indicated by a value of -1.

The IO, IN, OUT, and CLOSED options apply to both unlabeled HC units (used by BNA Version 1) and labeled HC units (used by HCFILE operations).

The effect of the AUTOUNLOAD MODE option on programmatic file closes is summarized in Table 2-1.

Table 2-1. AUTOUNLOAD Option Effects

Option Value	Effect on File CLOSE	Result
ON	CLOSE(F)	Rewound/unloaded
	CLOSE(F,REWIND)	Just rewind
	CLOSE(F,REEL)	Rewound/unloaded
	CLOSE(F,PURGE)	Rewound/unloaded

continued

Table 2-1. AUTOUNLOAD Option Effects (cont.)

Option Value	Effect on File CLOSE	Result
OFF (default)	CLOSE(F,*)	Position retained
	CLOSE(F,LOCK)	Rewound/unloaded
	REWIND(F)	Just rewind
	F.OPEN:=FALSE	Just rewind
	Block exit close	Rewound/unloaded
	Implicit close on reel switch	Rewound/unloaded
	CLOSE(F)	Just rewind
	CLOSE(F,REWIND)	Just rewind
	CLOSE(F,REEL)	Just rewind
	CLOSE(F,PURGE)	Just rewind
	CLOSE(F,*)	Position retained
	CLOSE(F,LOCK)	Rewound/unloaded
	REWIND(F)	Just rewind
	F.OPEN:=FALSE	Just rewind
	Block exit close	Just rewind
	Implicit close on reel switch	Just rewind

The value of the AUTOUNLOAD option is preserved across a halt/load. A program can override the setting of the AUTOUNLOAD MODE by using the AUTOUNLOAD file attribute. You can interrogate the value of the AUTOUNLOAD mode by using the system command OL (Display Label and Paths).

OL (Display Label and Paths)

OL (Display Label and Paths)

The OL (Display Label and Paths) command displays the label and path information for peripheral devices.

Syntax

```
OL <device> <unit number list>
   <A 17 OL options>
```

<A 17 OL options>

```
CTL <device number list>
PORT <port number list>
```

Explanation

The OL <device> <unit number list> command displays the label and path information for the devices. Because of differences in the hardware, the path information displayed for an Entry and Medium System (EMS) is different from the information displayed for a host data unit (HDU) system or a resource management module (RMM) system.

Most OL displays are divided into two parts: an upper area that contains information specific to the type of unit, and a lower area that describes the paths to the unit.

On EMS systems, if the path to the unit is through an outboard host – for example, the network support processor (NSP) data link processor (DLP) – a HOSTDLP column follows the PATHID column. The entries in the HOSTDLP column are the physical unit numbers of the controlling DLPs.

If no firmware-level information exists, the FIRMWARE column is omitted from the display.

The open count of each pack is displayed provided the total time to compute the number of open files for all packs on a screen does not exceed 60 seconds. The remaining packs on the screen (including the pack being computed when the time limit was reached) do not have an open count displayed. This case is different from a pack without open files, which displays an open count of zero.

The path information displayed varies depending on whether the system is an EMS or HDU system.

OL Displays for A 17 Systems

The A 17 OL displays are explained in the following paragraphs.

OL CTL <device number>

Displays the inboard and outboard information for the control.

The OL control display is divided into three parts: an upper area composed of one or more lines describing information specific to the control, an area describing the inboard paths, and an area describing the outboard devices.

Inboard information usually consists of the ports that serve the control. The following information is displayed, describing ports inboard of a control:

Inboard Port	Description
PORT	Gives the number of the port serving this control.
HOSTNSP	Gives the number of the NSP serving this line support processor (LSP).
STATUS	Describes the status of the port that serves the control. The following list shows the possible port status conditions: <ul style="list-style-type: none"> ● RESERVED The port is reserved for maintenance. ● OUT OF SERVICE The port is not available for use. ● IN USE The port has been chosen for all I/Os to the control. ● ONLINE The port is online.

OL PORT <port number list>

Displays the inboard and outboard information for the port.

The OL port display is divided into two parts: an upper area composed of one or more lines describing information specific to the port, and a lower area describing the controls outboard of the port.

The following information is provided for controls outboard of a port:

Outboard Port	Description
CTL (Control)	Gives the device number of the control outboard of the port.
TYPE	Gives the type of the control.
STATUS	Describes the status of the control. The possible control status conditions are listed under the STATUS heading for resource management module (RMM) systems later in this section.

continued

continued

Outboard Port	Description
VIA PORT	<p>Lists all ports that can access the control. The names of ports not online are enclosed in parentheses.</p> <p>The path information displayed varies, according to whether the system is an entry and medium system (EMS), host data unit (HDU), or resource management module (RMM) system.</p>

OL Displays for EMS Systems

The OL displays for EMS systems are explained in the following paragraphs.

PATHID

A number representing a unique path from a host to the peripheral. The path includes the DLP and controller of the unit, if a controller exists.

HOST

The unique host processor number where the path originates. A hyphen (-) in this column indicates that the path originates at a host DLP (NSP).

IOP PORT

The number of the input/output processor (IOP) port to which the base containing the DLP is attached. A hyphen (-) in this column indicates that the path originates at a host DLP.

MIRRORS

For a mirrored pack only. This line of information displays the number of offline members, the number of online members, the number of pending members, and the unit numbers of online and offline members. Pending members are those in the process of being created or being brought online. This line is omitted for packs that are not mirrored.

LEM PORT

A number representing the relative line expansion module (LEM) port the path traverses. When a path originates at a HOST DLP (NSP), the LEM PORT number refers to the LEM attached to the host DLP.

REL DLP

The relative DLP within the base.

BASEID

Identifies the base in which the DLP of the unit resides.

FIRMWARE

A number representing the firmware level of a path. If no path firmware was loaded, NOTLOADED is displayed in the firmware column for that path. If path firmware level does not apply to the unit, the entire column is omitted. For group-coded recording (GCR) DLPs, the information describes master electronics control (MEC) controlware. For NSP and LSP DLPs, the firmware level and firmware value are displayed on the top line. For all other DLPs, the DLP firmware level is described.

CONTROLWARE

For Host Transfer (HT) and HT-sequential DLPs, the value of the controlware level is shown for each path. It is displayed in both EBCDIC and hexadecimal characters.

HOST DLP

The physical unit number of the outboard host DLP in control of the unit. A hyphen appears in the IOP PORT column because the port is not part of the path, implying that this path originates at a host processor. The column is omitted if it does not apply.

CTL ID

The controller identification number (0 through F). It is valid only for HT-sequential DLPs; otherwise, it is omitted.

EXCH

The active exchange numbers (0, 1, or both). If 0, exchange 0 is available. If 1, exchange 1 is available. If the active exchange is both (0,1), exchange 0 and exchange 1 are both available. Exchange numbers are valid only for HT-sequential DLPs; otherwise, the column is omitted.

HOST ADAP

The function level and revision level of the host adapter (function level, revision level). These values are valid only for SMD1 DLPs; otherwise, the column is omitted.

FORMAT

The function level and revision level of the formatter (function level, revision level). These values are valid only for SMD1 DLPs; otherwise, the column is omitted.

ATTR PROM

The revision level of the current unit attribute PROM. This value is valid only for SMD1 DLPs; otherwise, the column is omitted.

AUTOUNLOAD

For magnetic tapes only. AUTOUNLOAD indicates whether or not the MCP is to unload a tape on the device when the unit is released by a close operation or when the MCP performs a reel switch from the unit. This option can be overridden by the AUTOUNLOAD file attribute.

BLOCKSIZE

For magnetic tape only, a number representing the maximum blocksize (in bytes, not words) that can be handled by the unit.

IN USE

An asterisk (*) indicates the path is currently selected for I/O by the processor of the path. If a column entry is blank, the path is not in use.

STATUS

Describes the status of a unit through a path and the status of a path to a unit. The possible path status conditions are described in the following table. READY and NOT READY status values do not apply to data comm units.

Status	Meaning
READY	The processor can determine that the unit is ready through the path.
NOT READY	The processor can determine that the unit is not ready through the path.
ONLINE	Unit status can be determined through the path.
OFFLINE	Unit status cannot be determined through the path.
RESERVED	The path is reserved for maintenance.
MAINTENANCE	The path is assigned to a maintenance task.

OL Displays for HDU Systems

The OL displays for HDU systems are explained in the following paragraphs.

HOSTDLP

The device number of the outboard host DLP in control of the unit. This column is omitted if it does not apply.

FIRMWARE

Same as FIRMWARE for EMS systems.

CONTROLWARE

Same as CONTROLWARE for EMS systems.

AUTOUNLOAD

Same as AUTOUNLOAD for EMS systems.

BLOCKSIZE

Same as BLOCKSIZE for EMS systems.

STATUS

Same as STATUS for EMS systems, but another entry type, IN-USE, indicates the current path used by the LSP.

MIRRORS

Same as MIRRORS for EMS systems.

OL Display for Resource Management Module (RMM) Systems

The following path information is provided in the OL display of a unit:

Path Information	Meaning
CTL	Gives the device number of the control that services the unit.
HOSTCTL	Gives the device number of the inboard host control for the unit (for example, the device number of the NSP servicing the LSP). This column is omitted when it is not applicable.
FIRMWARE	Gives a number representing the firmware level of a path. If path firmware level does not apply to the unit, the column is omitted. For group-coded recording (GCR) controls, the information describes the task control unit (TCU) firmware. For NSP, LSP, and DCDLP controls, the firmware level and firmware value are displayed on the top line. For all other controls, the control firmware is described.
CONTROLWARE	Lists EBCDIC characters representing the controlware level of the controller of the device.

continued

OL (Display Label and Paths) (cont.)

continued

Path Information

Meaning

STATUS

Describes the status of the control serving the unit. The following list shows the possible control status conditions:

- **FREED**
The control is not attached to the executing group.
- **RES MAINT**
The control is reserved for offline maintenance.
- **RESERVED**
The control is reserved for maintenance.
- **HUNG**
The control has not completed an outstanding I/O in a reasonable amount of time for that device type and will not be used for further I/O operations.
- **OUT OF SERVICE**
The control is not available for use.
- **NO PATH**
There is no available path to this control.
- **IN USE**
The control has been selected for I/O operations to the unit. This state is only displayed for NSP paths to LSP units.
- **ONLINE**
The control is online.

Example 1

The following display shows output from the OL command on EMS systems:

OL MT28

```

MT 28*P [000304] (3834) 6250 #1 1:0 <08/09/91> A9SUPPORT/FILE1271
CREATIONDATE: 08/09/91 EXPIRATION DATE: 09/08/91
CREATION SITE 20363
LABELKIND: LIBRARY MAINTENANCE
AUTOUNLOAD ON
PATHID HOST IOP LEM REL BASEID FIRMWARE BLOCKSIZE IN STATUS
          PORT PORT DLP
    03   1   6   0   7 1/1/0   0110   393210   *  READY
  
```

OL MT 48

MT 48*P D [TOMHB2] 1600 S C R A T C H POOL: TABLE (HOLD)
 AUTOUNLOAD OFF

PATHID	HOST	IOP	LEM	REL	BASEID	FIRMWARE	BLOCKSIZE	IN	STATUS
		PORT	PORT	DLP				USE	
01	1	0	1	0	0/0/0	0101	0		READY

OL DC110

DC 110 NSP (BLOCKED, LEVEL: 9.71) (8611) [3]

PATHID	HOST	IOP	LEM	REL	BASEID	IN	STATUS
		PORT	PORT	DLP		USE	
11	5	2	0	1	3/3/0	*	ONLINE
03	3	3	0	1	3/3/0		ONLINE
01	1	-	0	1	3/3/0		OFFLINE

OL DC114

DC 114 LSP (PROM), DL=1:0 (8611) [3]

PATHID	HOST	IOP	LEM	REL	BASEID	HOSTDLP	IN	STATUS
		PORT	PORT	DLP			USE	
11	5	2	0	1	3/3/0	110	*	ONLINE
03	3	3	0	1	3/3/0	-		ONLINE
01	1	-	0	1	3/3/0	-		OFFLINE

OL DC112

DC 112 NSP (BLOCKED, NOT LOADED) (8615) [3]

PATHID	HOST	IOP	LEM	REL	BASEID	IN	STATUS
		PORT	PORT	DLP		USE	
11	3	2	0	1	3/3/0		ONLINE

OL DC115

DC 115 LSP (RAM56KB, LEVEL: 2:45, FIRMWARE: 56KBLSP), DL=1:0(8611) [3]

PATHID	HOST	IOP	LEM	REL	BASEID	HOSTDLP	IN	STATUS
		PORT	PORT	DLP			USE	
11	5	2	0	1	3/3/0	110	*	ONLINE
03	3	3	0	1	3/3/0	-		ONLINE
01	1	-	0	1	3/3/0	-		OFFLINE

OL DC118

DC 118 LSP (PROM) (8615) [3]

PATHID	HOST	IOP	LEM	REL	BASEID	HOSTDLP	IN	STATUS
		PORT	PORT	DLP			USE	
11	3	2	0	1	3/3/0	112		OFFLINE

OL (Display Label and Paths) (cont.)

OL DC113

DC 113 LSP (RAM, FIRMWARE: NOT-LOADED) (8615) [3]
 PATHID HOST IOP LEM REL BASEID HOSTDLP IN STATUS
 PORT PORT DLP USE
 11 3 2 0 1 2/2/0 112 OFFLINE

OL PK44

PK 44*B [207044] #1 MCPS (4) [1,2,3]
 CREATED ON: 01/18/90 AT 08:29:83
 B9387/207 (INTERLACED)
 CAPACITY: 1117545 SECTORS (201158100 BYTES)
 PATHID HOST IOP LEM REL BASEID CONTROLWARE IN STATUS
 PORT PORT DLP USE
 11 3 1 0 3 1/3/0 QE[D8C5] * READY
 07 2 0 0 5 0/2/0 QE[D8C5] NOT READY
 03 1 0 0 5 0/1/0 QE[D8C5] READY
 02 1 3 0 6 5/1/0 QE[D8C5] RESERVED

OL PK212 (for an HTS DLP)

PK 212*B [207044] #1 PKNM (4) [1,2,3]
 CREATED ON: 02/12/90 AT 10:39:83
 B9387S/207 (SEQUENTIAL)
 CAPACITY: 1117545 SECTORS (201158100 BYTES)
 PATHID HOST IOP LEM REL BASEID CONTROLWARE CNTRL EXCH IN STATUS
 PORT PORT DLP ID USE
 11 3 1 0 3 1/3/0 QE[D8C5] 8 0 * READY
 07 2 0 0 5 0/2/0 QE[D8C5] 7 0,1 READY
 03 1 0 0 5 0/1/0 QE[D8C5] A 0 NOT READY
 02 1 3 0 6 5/1/0 QE[D8C5] B 1 * READY

OL PK47 (for an SMD1 DLP)

PK 47*B [207044] #1 PACKNAME (7) [1,2,3]
 CREATED ON: 03/16/90 AT 10:45:65
 SMD/214
 CAPACITY: 339864 SECTORS (201158100 BYTES)
 PATHID HOST IOP LEM REL BASEID HOST FORMAT ATTR IN STATUS
 PORT PORT DLP ADAP PROM USE
 11 3 1 0 3 1/3/0 6.5 7.8 7 * READY
 07 2 0 0 5 0/2/0 1.2 4.5 2 * READY

Example 2

The following display shows output from the OL command on A 12 and A 15 systems:

OL MT57

```
MT 57*P K[037252] (8513) 1600 #1 1:0 <01/10/91> SYSTEM2/FILE028
CREATIONDATE: 01/10/91 EXPIRATION DATE: 02/09/91
CREATION SITE 5
LABELKIND: LIBRARY MAINTENANCE
AUTOUNLOAD ON
DLP      FIRMWARE  BLOCKSIZE  STATUS
1007     0110     393210    ONLINE
1107     0110     393210    ONLINE
```

OL DC108

```
DC 108 NSP (BLOCKED, LEVEL: 15.71) (8599)
DLP      FIRMWARE  STATUS
108     15.71    ONLINE
```

OL DC112

```
DC 112 LSP (PROM), DL=1:1 (8599)
DLP      HOSTDLP   STATUS
112     108      IN USE
112     110      ONLINE
112     111      ONLINE
```

OL DC122

```
DC 122 LSP (PROM), DL=3:0 (1428)
DLP      HOSTDLP   STATUS
122     108      ONLINE
122     110      BROKEN
122     111      IN USE
```

OL DC123

```
DC 123 LSP (RAM, FIRMWARE: SMACHINE, LEVEL: 1.1), DL=1:0 (8599)
DLP      HOSTDLP   FIRMWARE  STATUS
123     108      01000101  IN USE
123     110      01000101  BROKEN
123     111      01000101  ONLINE
```

OL PK46

```
PK 46*B [207046] #1 DOC (0)
CREATED ON: 12/04/90 AT 14:59:04
B9387/207 (INTERLACED)
CAPACITY: 1117545 SECTORS (201158100 BYTES)
DLP      CONTROLWARE STATUS
2005     FREED
2045     QG      ONLINE
```


OL (Display Label and Paths) (cont.)

OL PK102

```
PK 102*B [235102] #1 NI32 (0)
CREATED ON: 01/09/90 AT 18:48:31
BX385/235
CAPACITY: 969145 SECTORS (174446100 BYTES)
DLP          CONTROLWARE STATUS
2015                FREED
2186                KC      ONLINE
```

Example 3

The following display shows output from the OL command on an A 17 system:

OL MT 82

```
MT 82*P [JRH ] 1600 #1 1:0 <01/10/91> JRHTAPE/FILE000
CREATIONDATE: 01/10/91 EXPIRATION DATE: 02/09/91
CREATION SITE 5
LABELKIND: LIBRARY MAINTENANCE
AUTOUNLOAD ON
CTL          FIRMWARE STATUS
2057        0110  ONLINE
2157                RESERVED
```

OL PK 164

```
PK 164*C [207124:207051:149] #2 PMDB (24)
CREATED ON: 05/05/90 AT 00:06:29
B9387/207 (INTERLACED)
CAPACITY: 155500 SECTORS (279990000 BYTES)
CTL          CONTROLWARE STATUS
2005                FREED
2045                QG      ONLINE
```

OL DC 109

```
DC 109 NSP (MULTI-HOST-BLOCK, LEVEL: 10.71) (2292)
CTL          FIRMWARE STATUS
109          10.71  ONLINE
```

OL DC 115

DC 115 LSP (RAM, FIRMWARE: SMACHINE, LEVEL: 1.1) (2292)
 CTL HOSTNSP FIRMWARE STATUS
 115 108 1.1 ONLINE
 115 109 1.1 IN USE
 115 110 1.1 RESERVED

OL HC 33

HC 33 (8232) BNA [IO]
 HUBNUMBER = 5, HUBINDEX = 07
 HCFILES HUBINDEX RESTRICTIONS:
 READPARTNER = NONE, WRITEPARTNER = NONE
 CTL FIRMWARE STATUS
 33 8405 ONLINE

OL PORT 12

PORT 12 [ERL=09.0,SLL=09.2905,SLL=0,FEATURES=0, (ALL HEX), 10/30/91]
 CTL TYPE STATUS VIA PORT
 321 HTS1 ONLINE 12,(14)
 52 HC2 RESERVED 12,(14)
 109 NSP4 OUT OF SERVICE 12,(14)

OL PORT 3

PORT 3 [ERL=09.0,SLL=09.2800,SLL=0,FEATURES=0 (ALL HEX), 10/10/91]
 CTL TYPE STATUS VIA PORT
 322 HTS1 FREED 1,3
 108 NSP3 IN USE 1,3
 2031 TP2 ONLINE 1,3
 801 MT2 OUT OF SERVICE 1,3

OL CTL 322

CTL 322: HTS1 <ONLINE> [Ua]
 PORT STATUS
 1 IN USE
 3 ONLINE
 OUTBOARD: 32,33,34,35,(36),(37),(38),(39)

OL CTL 109

CTL 109: NSP4 [10.71]
 PORT STATUS
 12 IN USE
 14 RESERVED
 OUTBOARD: 109

OL (Display Label and Paths) (cont.)

OL CTL 115

CTL 115: LSP2 <ONLINE> [SMACHINE, 1.1] (2292)
HOSTNSP STATUS
108 ONLINE
109 IN USE
110 RESERVED

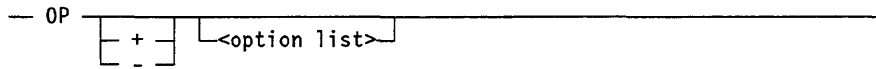
OL CTL 33

CTL 33: HC2 (8232) [I0]
HUBNUMBER = 5, HUBINDEX = 07
PORT STATUS
5 IN USE
OUTBOARD: 33

OP (Options)

The OP (Options) command displays, sets, or resets run-time operating system options that affect various system operations.

Syntax



<option list>



The available run-time operating system options are the following:

Option	Meaning	Option	Meaning
1	OPEN	24	OKTIMEANDDATE
2	TERMINATE	25	Not in use.
3	NOCHECK	26	LOGPOSITIONING
4	LPBDONLY	27	SERIALNUMBER
5	AUTORM	28	ARCHIVING
6	DIAGNOSTICS	29	Not in use.
7	CDONLY	30	LOCKTRACE
8	AUTORECOVERY	31	IORANGECHECK
9	DUPSUPERVISOR	32	Not in use.
10	DUPINTRINSICS	33	KEYEDIOII
11	TRANSWARNINGS	34	MIRRORING
12	AUTODC	35	DIAGNOSTICDUMP
13	NODUMP	36	AUDIT
14	CPBDONLY	37	FILESATURATION
15	AUTORUNNING	38	EOTSTATISTICS
16	CRUNCH	39	PATHBALANCING
17	BACKUPBYJOBNR	40	NETRECOVERY
18	PDTODISK	41	LOGIOERRORSNEW
19	NOFETCH	42	ISCDEBUG
20	RESOURCECHECK	43	IODIAGNOSTICS
21	Not in use.	44	PORTDEBUG
22	DIRDEBUG	45	USECATDEFAULT
23	CATALOGING	46	CATTEST
		47	MCPTTEST

Explanation

Following are descriptions of the OP command options:

OP

Displays all options and their respective states.

OP+

Displays all options that are set.

OP+ <option list>

Sets all options in the option list.

OP-

Displays all options that are reset.

OP- <option list>

Resets all options in the option list.

OP <option list>

Displays the options in the option list and their respective states.

<number>

Identifies the option desired. (Do not confuse the run-time operating system options with the option word assigned to each job.) The number is between 1 and 47; however, some values are not used. The number of each option is listed with the following descriptions.

OPEN (option 1)

If this option is set, a file-open message is displayed for each task whenever the task opens a file.

TERMINATE (option 2)

If this option is set, abnormal task terminations are processed normally. If the option is reset, abnormal terminations or other various error conditions result in a full memory dump.

NOCHECK (option 3)

If this option is set, memory dumps under FORGETCHECK conditions are inhibited. These dumps are automatic if the option is reset.

LPBDONLY (option 4)

If this option is set, all printer output files are assigned to a printer backup disk unless the PRINTDISPOSITION file attribute for the file has a value of DIRECT. (The printer backup disk is determined by the SB and DL BACKUP specifications.) These files can then be printed by the Print System, the SYSTEM/BACKUP utility, or the Work Flow Language (WFL) PRINT statement.

AUTORM (option 5)

If this option is set, the system automatically removes the old disk file when a duplicate-file condition occurs. If the option is reset, an RM (Remove) or OF (Optional File) command is required when such a condition occurs.

DIAGNOSTICS (option 6)

If this option is set, an RSVP request (for example, "RF DEGRADATION") is displayed at the console any time the reliability of a hardware unit is degraded by a set amount.

CDONLY (option 7)

With this option set, any job opening a card input file that is not internal to the job is discontinued. Also, no card reader can be labeled.

AUTORECOVERY (option 8)

If this option is set, a halt/load is attempted after all system-fatal memory dumps (except dumps caused by a suspended processor). After the halt/load, DCPs or NSPs that were running before the halt/load are reinitialized and the autobackup number is restored to the value it had before the halt.

If this option is reset, the AUTORECOVERY operations do not occur after a halt/load. Furthermore, the mix limit of all queues is set to 0 (zero) so that no jobs are automatically restarted.

DUPSUPERVISOR (option 9)

If a code file titled <file title> is designated as the supervisor program and this option is set, at halt/load time the system executes a code file titled <file name>/FMLYINXnnn. If this option is reset, the system executes the designated supervisor program. For more information about the supervisor program, refer to the description of the CS (Change Supervisor) command earlier in this section.

DUPINTRINSICS (option 10)

If a file titled <file title> is designated as the intrinsics file and this option is set, then at halt/load time the system uses the code file titled <file name>/FMLYINXnnn as the intrinsics file. If the option is reset, the code file <file title> is used as the intrinsics file at halt/load time. For more information on the intrinsics file, refer to the description of the SI (System Intrinsics) command later in this section.

TRANSWARNINGS (option 11)

If this option is set and a program is given translation warnings, then the translation warnings are displayed and the code file is marked as having been given translation warnings. (Translation warnings are issued when a file is opened and there is a risk of invalid data access and/or data corruption.)

If the option is reset, the translation warning mechanism is inactive. By default, the TRANSWARNINGS option is reset. For more information about translation, refer to the *A Series I/O Subsystem Programming Guide*.

AUTODC (option 12)

If this option is set, data comm is automatically initiated if it is not running when an MCS performs a DCWRITE operation.

The AUTODC option works as documented provided a station is defined in the SYSTEM/DATACOMINFO file as belonging to an MCS.

NODUMP (option 13)

If this option is set, the system does not try to take memory dumps. Potential nonfatal dumps are denoted by a display at the supervisory console and logged. The source of a fatal dump is listed in a system display at halt/load time. If the option is reset, dumps are taken in the normal fashion.

CPBDONLY (option 14)

If this option is set, all card punch files are assigned to a punch backup disk. (The punch backup disk is determined by the SB and DL BACKUP specifications.) These files can be punched by the Print System, the SYSTEM/BACKUP utility, or the Work Flow Language (WFL) PRINT statement.

AUTORUNNING (option 15)

When this option is set, certain default actions are taken in response to some of the RSVP requests that can arise during initialization. For some of these default actions, instead of waiting for you to reply, the RSVP request and the following system message (MSRMISC117) are displayed:

```
'OP AUTORUNNING' IS SET, REPLY <reply> ASSUMED
```

The reply indicates the default action selected by the system.

AUTORUNNING also controls how operator intervention is handled during memory dumps. When the AUTORUNNING option is set, DUMPDISKMASTER always creates a dump file on disk. If the DL DPFILES specification is null, the default disk location is the halt/load family.

The following default actions are taken by system initialization when the AUTORUNNING option is set:

- If the halt/load unit is not in the current configuration, the unit is acquired without an RSVP.
- If the OKTIMEANDDATE option is set, the RSVP is omitted.
- If the CATALOG family cannot be found and the site is not cataloging (the OP CATALOGING option is not set), the DL (Disk Location) specification defaults to the halt/load family.
- If the CATALOG family cannot be found and the site is cataloging (if the OP CATALOGING option is set), no default action is taken.
- All family complements proceed without RSVPs. If FLATREADER discovers bad records in a directory, it rebuilds the directory. If a disk family with duplicated directories is being complemented and one or more of the disks with duplicate directories is missing, the family is readied for use without the missing directories. If a site uses cataloging and a family is found to be missing from the volume library, the RSVP is skipped with an assumed OF (Optional File) response.
- If the JOBDESC family cannot be found, the DL specification defaults to the halt/load family.

The following default actions are taken by system memory dumps when the AUTORUNNING option is set:

1. If no disk dump files are specified, the dump is not taken. For more information, refer to the descriptions of the CM (Change MCP) and DN (Dump Name) commands in this section.
2. If a dump disk file exists and is not full, the memory dump is captured. When the DUMPDISKMASTER command is initiated after the dump completes, a check is made to determine the MARKLEVEL of the dump. If the MARKLEVEL is not equal to the currently running MCP, a message is displayed and the dump file within the dump disk file is skipped, leaving the file intact.

Note: *When the AUTORUNNING option is set, operator intervention is required when the disk dump file is full.*

CRUNCH (option 16)

If this option is set, code files and backup disk files are automatically crunched when they are closed. All other files are crunched when the CRUNCH option is used with the CLOSE statement.

BACKUPBYJOBNR (option 17)

If this option is set, print requests are selected for printing by job number. If this option is reset, print requests are selected for printing according to the criteria specified in the PS SELECTION command.

PDTODISK (option 18)

If this option is set, all program dumps are automatically sent out to disk. If this option is reset, all program dumps are automatically sent out to a printer.

NOFETCH (option 19)

If this option is set, you need not enter an OK (Reactivate) command to reactivate a WFL job containing a FETCH statement.

RESOURCECHECK (option 20)

If this option is set, resource limits established by queues are enforced (refer to the tape specification parameter of the MQ command, described earlier in this section). Before jobs are admitted into queues, the RESOURCE task attributes are checked to ensure they do not exceed the established limits. If the option is reset, the limits are ignored.

NOSUMMARY (option 21)

This option has been deimplemented. Use the PS DEFAULT JOBSUMMARY command.

DIRDEBUG (option 22)

Note: This option is intended for use by Unisys facilities only.

CATALOGING (option 23)

CATALOGING is tested during disk initialization after a halt/load. If this option is set, the CATALOGLEVEL attribute is initialized to CATALOGLEVELSET; if it is reset, CATALOGLEVEL is set to 0. A cataloging operating system is any operating system whose CATALOGLEVEL is greater than 0. The effect of CATALOGLEVEL is described in the *A Series Disk Subsystem Administration and Operations Guide*.

Note: This option can be enabled or disabled during the OKTIMEANDDATE verification. If you enable or disable the option after system initialization is completed, the change does not take place until a halt/load occurs.

OKTIMEANDDATE (option 24)

This option requires you to verify the system TIME and DATE values at halt/load time.

If the option is set, the current TIME and DATE settings are displayed on the ODT after a halt/load, and you must enter TIMEOK in response to the message "PLEASE VERIFY TIME AND DATE" to resume normal processing. To correct TIME or DATE before entering TIMEOK, use the DR (Date Reset) or TR (Time Reset) commands.

Caution:

If you use disk mirroring at your site, a halt/load performed during the delay while the system waits for the response to the RSVP message causes the mirrors to lose their contents.

LOGPOSITIONING (option 26)

If this option is set, when a parity I/O error on a tape occurs, the system tries to recover from the error by repositioning the tape and retrying the I/O process. If an I/O error occurs during the repositioning of a tape, you can use the log records that are written to analyze why the recovery was not successful.

SERIALNUMBER (option 27)

If this option is set, files are not assigned to scratch tapes unless either the SERIALNO file attribute matches the serial number of the tape, the SCRATCHPOOL file attribute matches the scratch pool designation of the tape, or an OU (Output Unit) system command is used to direct the assignment to a particular scratch tape. For more information, refer to the OU command description later in this section.

ARCHIVING (option 28)

This option enables the archiving function of the catalog subsystem. If this option is set and the catalog level of the system is greater than 0 (zero), an archive log is created. The name of the archive log is *ARCHIVELOG/<date>/<time>*, where <date> and <time> are the creation date and creation time of the file, respectively. If the archive log cannot be set up, the system automatically resets the option.

LOCKTRACE (option 30)

Note: This option is intended for use by Unisys facilities only.

IORANGECHECK (option 31)

This option verifies that the disk address requested for an I/O is within the range of one of the areas of the file. For IORANGECHECK to have any effect, the compile-time option DIAGNOSTICS must be set when the system is being compiled.

KEYEDIOII (option 33)

Setting this option ensures that all new files created with FILEORGANIZATION = VALUE (INDEXED), VALUE (INDEXEDNOTRESTRICTED), or VALUE (KEYEDIOII) are linked to the SYSTEM/KEYEDIOII library and are created as KEYEDIOII files. Existing files are correctly linked to either SYSTEM/KEYEDIO or SYSTEM/KEYEDIOII depending on how they were created. Therefore, you can read and update both KEYEDIO and KEYEDIOII files, but all new files are KEYEDIOII.

Resetting this option ensures that all new files created with FILEORGANIZATION = VALUE (INDEXED) or VALUE (INDEXEDNOTRESTRICTED) are linked to the SYSTEM/KEYEDIO library and are created as KEYEDIO files. Existing files are correctly linked to either SYSTEM/KEYEDIO or SYSTEM/KEYEDIOII depending on how they were created. KEYEDIOII files are created only when the FILEORGANIZATION is explicitly set to VALUE (KEYEDIOII). Otherwise, a KEYEDIO file is created.

MIRRORING (option 34)

This option allows disk mirroring to occur after the next halt/load. At that time, a mirror information table (MIT) and other structures needed for disk mirroring are created.

Mirrored sets are allowed to come online only if the MIRRORING option was set before the last halt/load, and it has remained set.

The option can be changed, subject to the following conditions:

- An *OP+ MIRRORING* command is ignored if mirroring is already in effect.
- An *OP- MIRRORING* command is rejected if any mirrored sets are currently present in the system. All mirrored sets must first either be closed or be reduced to single packs with the *MIRROR RELEASE* command.

DIAGNOSTICDUMP (option 35)

If set, this option enables certain other memory dump calls to be made for diagnostic purposes. Also, the system provides maximum diagnostic information when an exception condition occurs.

AUDIT (option 36)

Note: This option is intended for use by Unisys facilities only.

FILESATURATION (option 37)

If set, this option causes the following warning message to be issued whenever the system allocates the last row of a disk file:

WARNING, ALLOCATING LAST ROW OF FILE <file name>

The preceding message is subject to the following restrictions:

- It applies to “programmer” files, including ordinary data files, printer/punch backup files, and compiler code files.
- It does not apply to internal system files such as SYSTEM/SUMLOG or library maintenance files.
- It does not necessarily mean that the file is almost full. Some areas might have been skipped if the file was written randomly.

For files whose FLEXIBLE attribute is set to TRUE, the message is issued only when row 1000 is allocated. Most files have the TRUE value for the FLEXIBLE attribute (it is the default value), and few files need 1000 areas. Therefore, the warning message seldom appears.

EOTSTATISTICS (option 38)

If this option is set, end-of-task (EOT) statistics for each task are displayed on the ODT in the following format:

```
<mix number> EOT TIMES: ET=hhh:mm:ss.s PT=hhh:mm:ss.s IO=hhh:mm:ss.s
```

In this display, *hhh* denotes the hours, *mm* denotes the minutes, and *ss.s* denotes the seconds (rounded to the nearest tenth).

PATHBALANCING (option 39)

This option, which enables dynamic path balancing, applies to EMS systems – with the exception of A 10 systems that use MCP. (Path balancing is performed automatically on HDU systems and on A 10 systems that use MCP.)

Dynamic path balancing distributes pack I/Os more evenly across busy multiple-path disk pack subsystems. For tape units, the order in which data is transferred is crucial; therefore, the static path allocation technique is always used for tapes.

Path business calculations and dynamic path balancing do not occur on single-path subsystems.

NETRECOVERY (option 40)

If this option is set, during halt/load processing the system determines the last state of the currently selected BNA. If the state was *not running*, BNA is not initiated. If BNA was *running* before the halt/load, BNA libraries and stacks are initiated.

LOGIOERRORSNEW (Option 41)

If this option is set, all I/O errors are logged in the Maintenance MLIOEXCEPTION SUMLOG record. If this option is reset, MLI I/O errors continue to be logged in the Maintenance MLIOERRORHDP SUMLOG record.

Note: In a future release this option will be removed and all I/O errors will automatically be logged in the Maintenance **MLIOEXCEPTION SUMLOG** record.

ISCDEBUG (Option 42)

Note: This option is intended for use by Unisys facilities only.

IODIAGNOSTICS (option 43)

Note: This option is intended for use by Unisys facilities only.

PORTDEBUG (option 44)

If this option is set on an operating system that was compiled with the compile-time option **INTERNAL** set (a *diagnostic* system), port debug trace information is written to the **SYSTEM/SUMLOG** file in BNA debug log records. You can examine this information by running **LOGANALYZER**.

USECATDEFAULT (option 45)

The value of this option has meaning only on an operating system that is running **CATALOGING**. If this option is set, the default value of the file attribute **USECATALOG** is **TRUE**.

CATTEST (option 46)

Note: This option is intended for use by Unisys facilities only.

MCPTTEST (option 47)

Note: This option is intended for use by Unisys facilities only.

Example 1

The following example shows the response to a simple OP command. Options that have an asterisk (*) next to them are set; all others are reset.

OP

```

----- OPTIONS -----
 1 OPEN          2*TERMINATE      3 NOCHECK        4*LPBDONLY
 5*AUTORM        6 DIAGNOSTICS      7 CDONLY         8*AUTORECOVERY
 9 DUPSUPERVISOR 10 DUPINTRINSICS  11 TRANSWARNINGS 12*AUTODC
13 NODUMP        14*CPBDONLY        15 AUTORUNNING   16*CRUNCH
17 BACKUPBYJOBNR 18 PDTODISK        19*NOFETCH       20*RESOURCECHECK
26*LOGPOSITIONING 22 DIRDEBUG        23*CATALOGING    24*OKTIMEANDDATE
27*SERIALNUMBER  28 ARCHIVING       30 LOCKTRACE     31 IORANGECHECK
33 KEYEDIOII     34 MIRRORING       35 DIAGNOSTICDUMP 36 AUDIT
37 FILESATURATION 38 EOTSTATISTICS  39 PATHBALANCING 40 NETRECOVERY
41 LOGIOERRORSNEW 42 ISCDEBUG        43 IODIAGNOSTICS 44 PORTDEBUG
45 USECATDEFAULT 46*CATTEST         47 MCPTEST
    
```

Example 2

This example shows the display for a status query of option 1:

OP 1

1 OPEN RESET

Example 3

This example displays all the options that are reset:

OP-

```

----- RESET OPTIONS -----
 1 OPEN          3 NOCHECK
 6 DIAGNOSTICS  7 CDONLY
 9 DUPSUPERVISOR 10 DUPINTRINSICS
13 NODUMP        17 BACKUPBYJOBNR
22 DIRDEBUG      24 OKTIMEANDDATE
31 IORANGECHECK  43 IODIAGNOSTICS
46 CATTEST       47 MCPTEST
    
```

Example 4

These commands set or reset particular options:

OP+ 3

3 NOCHECK SET

OP- 3

3 NOCHECK RESET

OP+ OPEN

1 OPEN SET

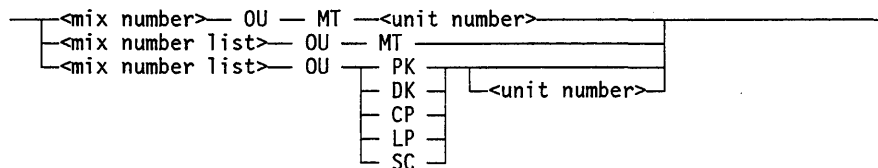
OP- OPEN

1 OPEN RESET

OU (Output Unit)

The OU (Output Unit) command directs output from the specified task or tasks to a specified output device or to any available device of the specified type.

Syntax



Explanation

<mix number> OU MT <unit number>

Directs output from the task specified by the mix number to the specified tape unit, provided it is write-enabled and identified as SCRATCH. (This command overrides locked units.) This form of the command can override a request for a specific type of tape unit and can substitute any other type of unit, if no units of the requested type are available. The system ignores any substitute backup setting set by the SB (Substitute Backup) command. The OU MT command overrides any tape serial number, density specification, and SCRATCHPOOL specification that the program makes. To use this form of the command, the unit must be ready, write-enabled, and not marked with HOLD status. For more information on HOLD status, refer to the HOLD (Hold Unit for Operator Use) command earlier in this guide.

<mix number list> OU MT

Has no effect on nonbackup files. Causes the backup file opened by the specified task to be written to any available tape drive. If a substitute backup medium was established for tape by the SB (Substitute Backup) command, the output goes to the substitute medium instead.

<mix number list> OU PK

Has no effect on nonbackup files. Controls the destination for backup output files as follows:

- If the file has a family name specified, the file is sent to the pack with that family name.
- If a file has no family name specified, it is sent to PACK by default. If a substitute backup medium was established for PACK by the SB (Substitute Backup) command, output goes to the substitute medium instead.

<mix number list> OU PK <unit number>

Places the files on the specified disk pack unit. Only a native-mode, write-enabled base pack with the specified name can be designated. Any substitute backup setting set by the SB (Substitute Backup) command is ignored.

<mix number list> OU DK

Has no effect on nonbackup files. Controls the destination for backup output files as follows:

- If the file has a family name specified, the file is sent to the pack with that family name. Only a native-mode, write-enabled base pack with the specified name can be designated.
- If a file has no family name specified, it is sent to DISK by default. If a substitute backup medium was established for DISK by the SB (Substitute Backup) command, output goes to the substitute medium instead.

<mix number list> OU DK <unit number>

Places the files on the memory disk unit identified by <unit number>. Any substitute backup setting set by the SB (Substitute Backup) command is ignored.

<mix number list> OU CP

<mix number list> OU CP <unit number>

Direct the output files to a specific card punch unit if <unit number> is specified, or to any available card punch if <unit number> is omitted.

<mix number list> OU LP

<mix number list> OU LP <unit number>

Direct the output files to a specific line printer if <unit number> is specified, or to any available line printer if <unit number> is omitted.

<mix number list> OU SC

<mix number list> OU SC <unit number>

Direct the output files to a specific ODT if <unit number> is specified, or to any available ODT if <unit number> is omitted.

Example

Several RSVP requests accept a response of OU. One such request is system message MSRFOUT30: "<file name> REQUIRES <unit identifier>". This message is generated when a task cannot locate a unit that matches the selection attributes of an output file.

If the program requests a backup medium, backup information is appended to the message. If the program requests a particular tape serial number or density, the specified serial number or density is appended to the message.

The operator can enter an OU response; for example:

```
2121 OU MT 1
```

If the OU response specifies an inappropriate device, an error message is displayed. Two possible messages are the following:

```
DIRECT FILE CANNOT BACKUP
```

```
NEED AN OUTPUT TAPE FOR OU MT
```

The first message is generated because a direct file asking for a line printer or card punch cannot go to backup under any circumstances. *Direct* implies that the program can look at result descriptors and can set error maskout. Direct files must therefore deal with an actual target peripheral.

In the second message, an OU MT response was unsuccessful because no suitable tape is available.

Considerations for Use

Other messages that accept an OU response are as follows:

```
PK<unit number> <number> SECTORS REQUIRED
```

```
CATALOG FAMILY MISSING (IL OR HL): <family name>
```

```
<unit type> <unit number> TYPE DISAGREES WITH KIND,  
REQUIRES MT # <unit number>
```

```
<device ID> UNABLE TO LOAD TRAIN, SELECT ANOTHER PRINTER
```

```
<unit identifier> REQUIRES FORMID <form name>
```

```
<unit identifier> MUST BE UNFORMED
```

For installations that use the tape security subsystem, be aware of the following effects of the OU command:

OU (Output Unit) (cont.)

- You can use the OU MT command to assign to the program an output tape volume that is not listed in the tape volume directory.
- You cannot use OU MT command to override the familyowner specification of the tape security subsystem.

For information about the tape security subsystem, refer to the TAPECHECK = AUTOMATIC option in the *A Series Security Administration Guide*.

You can use the OU command to override a mismatch between the FORMID attribute of the file, if specified, and the form of a printer, if any. However, these commands cannot override the KIND attribute of the file.

Any attempt to override the KIND attribute of the file generates the following message that indicates the devices expected by the OU command:

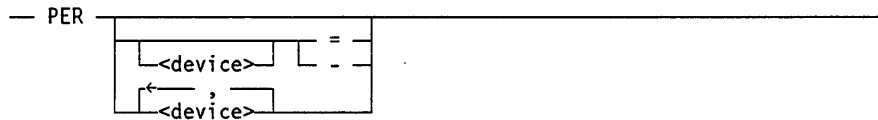
```
OU REJECTED, REQUIRES <device list>
```

The RSVP that requested the OU command is reissued.

PER (Peripheral Status)

The PER (Peripheral Status) command displays the status of peripheral units.

Syntax



Explanation

PER

Displays the status of all peripheral units.

PER <device>

Displays the status of all peripheral units of the specified device type.

PER=

PER-

Display the status of all peripheral units, including unlabeled and unready devices.

PER <device> =

PER <device> -

Display the status of all peripheral units of the specified device type, including unlabeled and unready devices.

If a pack on a given unit has a status of U (unavailable for normal use), any of the following messages might be displayed:

Message	Meaning
UNLABELED	Indicates that no label information exists in memory for the pack. Either the pack has not been initialized, or a <i>CLOSE PKnn</i> command has been entered for the unit, or the directory complementor failed while reading the directories on the pack.
UNINITIALIZED	Indicates that the pack has not been reconfigured with an RC (Reconfigure Disk) command.
BLASTED	Indicates that the pack was cleared and must be closed and readied before being used.
Any of the following messages: LABEL ERROR, SCRATCH, NOT READY, SAVED, or DUP SERIAL.	Indicates that the status of a pack is U (unavailable for normal use).

Example 1

PER MT

```

----- MT STATUS-----
48*\ [TOMHB2] 6250 S C R A T C H POOL: HALL (HOLD)
50*\ [111222] 1600 S C R A T C H
51*\ [911118] (887) 6250 #1 1:0 <11/18/91> DTAPE/FILE001
52*\ [000001] 6250 #1 1:0 EXAM (HOLD)
53*\ [IADDIE] 6250 S C R A T C H POOL: CUE
115*\ [000116] 38000 #1 1:0 <3/07/90> MASTER/TEST
116 H SAVED
131*\ [000065] (2321) 1600 #1 1:3 <6/12/90> CAN/FILE000
132*\ R[999999] 1600 #1 1:1 <11/13/90> DANGEROUS/FILE000
133*\ D [000002] 1600 #1 1:1 <12/04/90> A/B
134*\ [000033] 1600 #1 1:1 <10/17/91> A/B
164 P [012345] 1600 #1 1:1 BLOCKSIZE 66000 TOO LARGE, CAN'T READ
    
```

The preceding display shows the serial number in brackets ([]), the tape density, the reel number, the cycle number and version number separated by a colon (:), the creation date in angle brackets(<>), and the title of the tape. If the tape is in use, the mix number is also displayed in parentheses following the serial number. If the tape is in a scratch pool, the pool name follows the colon after the words S C R A T C H POOL. If the tape is being held, the designation HOLD appears in parentheses.

The following table describes the displayed fields and their meanings:

Field	Description
1	Shows the unit number.
2	Contains an asterisk (*) if the tape is write-enabled.
3	The number 8 indicates an 8 millimeter cartridge tape. The number 9 indicates a 9-track nonreturn-to-zero (NRZ) tape. The letter P indicates a 9-track phase-encoded (PE) or group-coded recording (GCR) tape. The letter H indicates a half-inch cartridge tape. The letter Q indicates a quarter-inch cartridge tape.
4	A backslash (\) or an x designates a tape entered in a volume library.
5	The letter D indicates that the mounted volume is listed in the Volume Directory.
6	The letter R indicates that the unit has had security restrictions imposed on it. For more information, see the description of the RESTRICT (Restrict Unit or Volume) command.

If the label of the tape mounted on a tape drive indicates that the block size of the file exceeds the maximum block size that the unit can handle, the following message is displayed in place of the title. In the message that follows, <I> is the block size in the label:

```
BLOCKSIZE <I> TOO LARGE, CAN'T READ
```

Note *If SYSOPS TAPEEXPIRATION = ON and an expired tape is mounted with a write ring, the display shows the tape as a SCRATCH tape. If SYSOPS TAPEEXPIRATION = ON and an expired tape is mounted without a write ring, the display shows the tape as a labeled tape.*

Example 2

For mirrored packs, the *PER PK* command displays only one member of each mirrored set, but the *PER PK=* command displays all members.

PER PK

```

----- PK STATUS -----
96*B      [000097] (MCP) #1 DAMISDB (1)
97*Bx     [000096] #1 SSDMASTER (11)
98*C      [000098:000097:096] #2 DAMISDB (0)
99 U      C L O S E D
100*U     U N I N I T I A L I Z E D
103*B     [000099] #1 TRIAL (1)
112 B R   [002112] #1 HAZARD (0)
    
```

Each pack entry in the PER PK display includes the unit number, type of pack, and label information (family name and serial number). The display also indicates whether the pack is currently being used by an MCP procedure (for example, READPACKLBL). The following fields are displayed:

Field	Description
1	The unit number.
2	An asterisk (*) if the write-enable switch is on and the MODE is IO or OUT; a blank if the write-enable switch is off or the MODE is IN.
3	The pack type. The following codes are displayed: <ul style="list-style-type: none"> ● B: Native-mode base pack ● I: Interchange-mode pack ● C: Native-mode continuation pack ● U: A pack that is not available for normal use
4	On the ODT in this example, the character displayed to denote volume library entries is an <i>x</i> .
5	The letter R if security restrictions were imposed on the pack. For more information, see the description of the RESTRICT command.
6	The letter M if the pack is mirrored. If a mirrored set is in the process of coming online but is waiting for another previously online member, it is considered a partial mirrored set. These packs are marked with a lowercase <i>m</i> instead of an uppercase M.

continued

continued

Field	Description
7	The disk pack family description for native-mode packs or the disk pack serial number in brackets ([]). For native mode packs, the family description consists of the following three items, with colons (:) separating them: <ul style="list-style-type: none"> • The serial number of the disk pack on this unit. • The serial number of the pack used to identify the family; that is, the first pack created in the family. Note that this will not necessarily be the base pack if the family has duplicated directories. • The unit number of the current base pack for this family. If item 3 refers to the unit itself, then it is omitted. If item 3 refers to the unit itself and item 2 is the serial number of the pack on this unit, then items 2 and 3 are omitted.
8	The family index number, from 1 through 255.
9	The family name.
10	The count (in parentheses) of files open or in use on the disks.

Example 3

This example shows a display for a memory disk on an A 3 system:

```
PER DK-
----- DK STATUS-----
255*MD (READY:0) [000255] #1 MEMDISK (0)
```

Example 4

This example shows the response to a PER DC command. DC designates a data-comm-type device: an NSP, LSP, or DCDLP.

```
PER DC
----- DC STATUS-----

107 NSP (BLOCKED, NOT LOADED)
108 NSP (MULTI-HOST-BLOCK, LEVEL: 10.70) (1430)
109 NSP (DC-DLP, LEVEL 25.8) (8915)
110 NSP (MULTI-HOST-BLOCK, LEVEL: 10.70) (1429)
111 NSP (BLOCKED, LEVEL: 15.70) (1428)
122 LSP (PROM), DL=4:0 (1428)
123 LSP (RAM, FIRMWARE: SMACHINE, LEVEL: 1.1), DL=1:0 (1430)
124 LSP (RAM, FIRMWARE: BDLC, LEVEL: 1.54), DL=4:2 (1428)
125 LSP (RAM56KB, FIRMWARE: 56KBLSP, LEVEL: 2.18), DL=3:0(1429)
126 LSP (RAM, FIRMWARE: NATIVE, LEVEL: 1.35), DL=4:1(1428)
127 LSP (RAM, FIRMWARE: NOT LOADED)
128 LSP (RAM, FIRMWARE: POLLSELECT, LEVEL: 1.80), DL=1:1(1430)
```

For an NSP or a DCDLP, the first entry describes the NSP type, and the second entry describes its firmware level (the first number describes the patch level, and the second number describes the PROM level). If no firmware has been loaded, "NOT LOADED" is displayed as the firmware level. If the firmware level is zero, it is not displayed.

For an LSP, the first entry describes its type, the second entry indicates its firmware type, and the third entry indicates its firmware level (the patch level and PROM level). If no firmware has been loaded, the message "FIRMWARE: NOT LOADED" appears in the firmware type entry, and the firmware level is 0 (zero). The level is not displayed for LSPs that are of the type PROM.

Example 5

The open count of each pack is displayed in parentheses if the total time to compute the number of open files for all packs on a page does not exceed 5 seconds. If the computation time exceeds 5 seconds, the field is not displayed. (A pack without open files, on the other hand, displays an open count of zero.)

This example shows the display in which the open count for some units is not computed because the computation time exceeded 5 seconds:

```
150*B [207150] #1 DISK (27)
151*B [206151] #1 TESTPACK (0)
*** OPEN COUNT NOT COMPUTED FOR REMAINING UNITS (USE OL FOR COUNT) ***
153*B [207153] #1 PACK
162*B [207162] #1 DBUNIT
```

Example 6

This example shows the display for three ODTs. The first two are not restricted; the third is restricted:

PER SC

```
----- SC STATUS-----
1  S C R A T C H
2  S C R A T C H
3  (MCP) COMS/ODT/DRIVER RESTRICTED
```


Example 7

This example shows displays for line printers attached to the system:

PER LP

```

----- LP STATUS-----
4          S C R A T C H  EBCDIC96
5  (007)   SERVER/LP5/"R#8476"/"J#3284" ASCII86
7  (0030)  *SYSTEM/PRINT COPY  (UNKNOWN TRAINID)
8  (6969)  FORMED : MEDFORM  SERVER/LP8/"R#8476"/"J#3299"
9          FORMED : "MEDICAL FORM # 2"
10 (MCP)   SAVED              (DEFAULT DESTINATION)
11        N O T R E A D Y  EBCDIC96 (DEFAULT DESTINATION)
12        R E S E R V E D
    
```

Each entry in the PER LP display includes the external unit number and the status of the printer. The following fields are displayed:

Field	Description
1	Indicates the external unit number of the printer.
2	Optional field that shows special I/O status enclosed in angle brackets (<>).
3	If the printer is in use, this optional field indicates the system or the mix number of the stack using the printer. If the printer is not in use, this optional field indicates that the printer is configured for special forms with the designation FORMED, followed by the form identifier.
4	Indicates the status of the printer or who is controlling the printer. Terms used in this field include TO BE SAVED, SAVED, LOCKED, S C R A T C H, NOT READY, the file title, and the name of the print server if a file is printing.
5	If the printer is not in use, this optional field contains either the name of the TRAINID character or the statement (<i>UNKNOWN TRAINID</i>). TRAINID is a file attribute that refers to the character set that is available to the printer.
6	If the printer is not in use, this field is blank. If a printer that is in use is ABED, this field contains the name of the TRAINID or the words (<i>UNKNOWN TRAINID</i>). If a printer that is in use is FORMED, then the field contains the name of the form.

PG (Purge) and PGL (Purge and Lock)

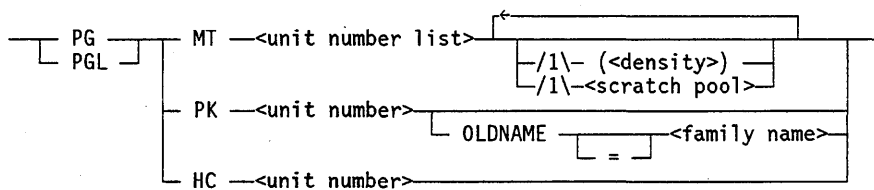
The PG (Purge) command purges tape, disk pack, or host control units. The pack and tape units must be ready and not in use, and the units must be write enabled. Host control units can be purged when they are not ready.

The PG command is not valid for use with online mirrored packs. Before using the PG command, you must release the pack with the MIRROR RELEASE command. The PG command is valid for closed mirrored packs provided you respond OK to the RSVP request that appears.

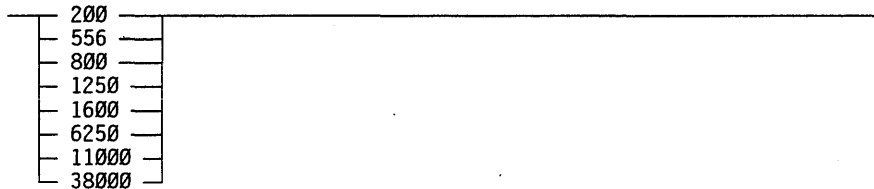
The PG command optionally specifies the recording density for tapes and assigns tapes to scratch pools.

The PGL (Purge and Lock) command purges and locks tape, disk pack, or host control units. You cannot use them until you ready the unit with the RY (Ready) command.

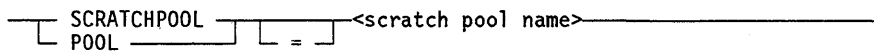
Syntax



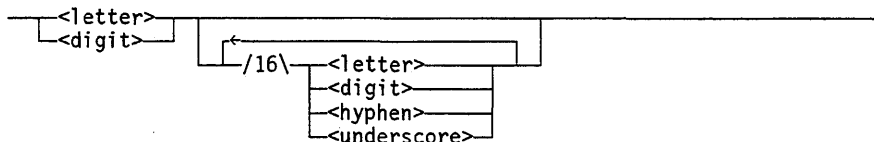
<density>



<scratch pool>



<scratch pool name>



Explanation

PG MT <unit number list>

Purges the volumes mounted on the tape units in the unit number list. The PG command requires that the tape is labeled and has a serial number; however, this serial number is not affected by the command. To assign serial numbers to unlabeled tapes, use the SN (Serial Number) command.

(<density>)

Specifies the recording density of the tape. If the density is specified, it applies to all tapes mounted on the units in the unit number list. If the density you specify is not valid for the unit, the system displays the following message:

```
MT <unit number> INVALID DENSITY SPECIFIED
```

In such a case, the tape density is not changed, but the system still purges the tape.

PGL MT <unit number list>
PGL MT <unit number list> (<density>)

Has the same effect as the PG command except that it locks the tape units after purging them so that no job can automatically acquire them as scratch tapes.

PG MT <unit number list> SCRATCHPOOL = <scratch pool name>
PG MT <unit number list> POOL = <scratch pool name>

Purges a tape and assigns a scratch pool name to it.

PG MT <unit number list> (<density>) SCRATCHPOOL = <scratch pool name>
PG MT <unit number list> (<density>) POOL = <scratch pool name>

Purges a tape and assigns a density and a scratch pool name to it.

PGL MT <unit number list> SCRATCHPOOL = <scratch pool name>
PGL MT <unit number list> POOL = <scratch pool name>

Purges a tape and assigns a scratch pool name to it. This command also locks the tape after it is purged so that no job can acquire the tape.

As an alternative to PGL, you might want to use the HOLD command before you purge the tape with the PG command. You can then see the tape label information after the purge is completed.

PGL MT <unit number list> (<density>) SCRATCHPOOL = <scratch pool name>
PGL MT <unit number list> (<density>) POOL = <scratch pool name>

Purges a tape and assigns a density and a scratch pool name to it. This command also locks the tape after it is purged so that no job can acquire the tape.

As an alternative to PGL, you might want to use the HOLD command before you purge the tape with the PG command. You can then see the tape label information after the purge is completed.

PG PK <unit number>

Relabels the disk pack identified by the unit number with the family name SCRATCH and makes it available to maintenance programs. This form of the PG command is intended primarily for maintenance and should not be used for normal operation.

PG PK <unit number> OLDNAME = <family name>

For labeled disk packs, the PG command checks the family name of the pack. If the OLDNAME option is not specified, you must verify the old name through an ACCEPT response to an RSVP request.

If you enter a W (Waiting Mix Entries) command, or have previously requested automatic display of current system status information by entering an ADM (Automatic Display Mode) command, the system displays the waiting entries. For example:

```
<mix number> PK <pack number> is [<serial number>] <pack name>  
<mix number> ACCEPT: OLDNAME = <pack name>
```

You must reply as follows:

```
<mix number> AX OLDNAME = <pack name>
```

The <pack name> entered is compared with the pack name in the pack label. If there is a mismatch, the task is terminated with the following error message:

```
PKnn INCORRECT OLDNAME ENTERED -- RC/PG/LB NOT DONE
```

The OWNER identification is verified before the OLDNAME.

PG HC <unit number>

Marks the Host Control (HC) unit identified by <unit number> as *unlabeled*. When the unit is in this state, it can be assigned to a BNA Version 1 station group. If the unit is the last unit of a given hub to be purged and the hubname was not specified in the group configuration, then the temporary hubmap is purged.

PG (Purge) and PGL (Purge and Lock) (cont.)

Examples

PG MT 114 (1600)

MT 114 WILL BE PURGED

PG MT 66

MT 66 WILL BE PURGED

PG HC 66

HC 66 WILL BE PURGED

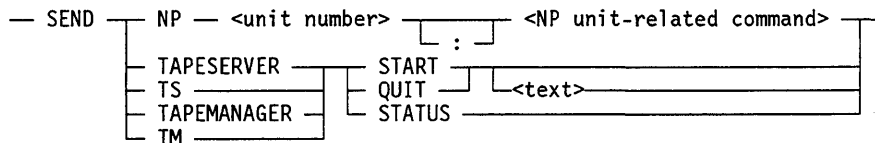
PG MT 110 (1600) SCRATCHPOOL = ACCOUNTING

MT 110 WILL BE PURGED

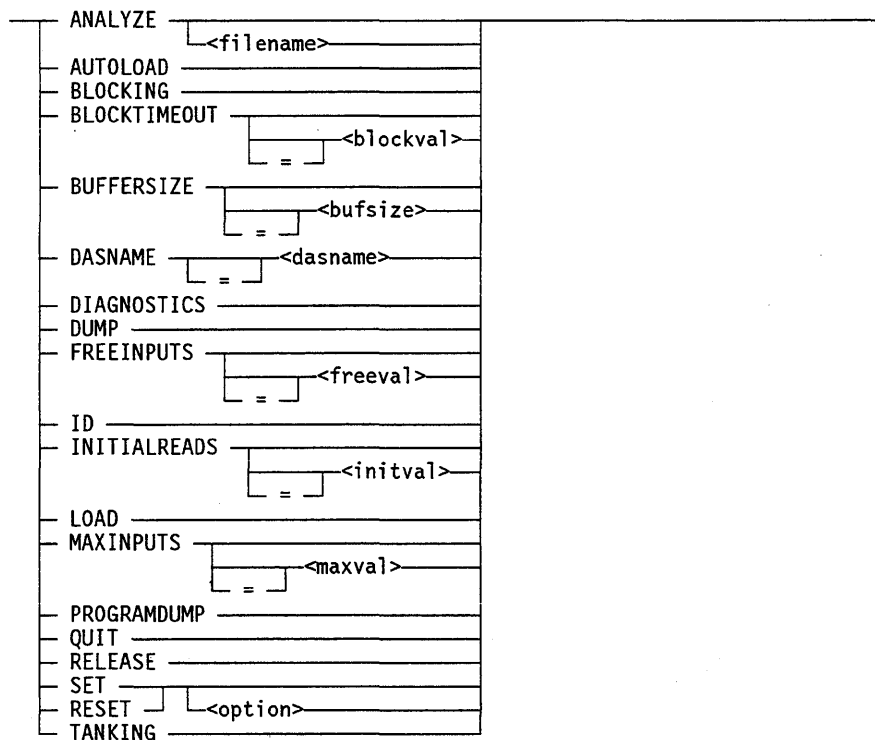
SEND (Send Message)

The SEND (Send Message) command communicates with the network processor (NP) support library. It also initiates, terminates, and displays the status of the TAPESERVER library and the TAPEMANAGER library.

Syntax



<NP unit-related command>



The <bufsize> construct specifies an integer in the range of 480 through 64000.

The <blockval> construct specifies an integer in the range of 10 through 2000.

The <dasname> value can be one of the following names:

- INTRASYSTEM/BNAV2ICPINITIALIZATION for BNA Version 2 network processors (NPs)
- INTRASYSTEM/NTS_TWIXAX_INITIALIZATION for Workstation data link processors (DLPs)
- INTRASYSTEM/HLCNICPINITIALIZATION for Host LAN Connection NPs

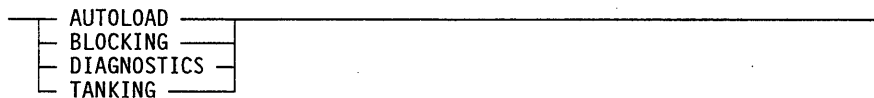
SEND (Send Message) (cont.)

The <maxval> construct specifies an integer in the range of 10 through 127.

The <freeval> construct specifies an integer in the range of 1 through 63.

The <initval> construct specifies an integer in the range of 1 through 63.

<option>



Network Processor System Commands

The following are system commands applicable to network processors:

SEND NP <unit number>: <NP unit-related command>

Specifies that the NP unit-related command is to be sent to the network processor identified by unit number. The semantics of the NP unit-related command, as well as the responses to these messages, are defined by the NP support library; the text of the message is not inspected by the CONTROLLER or the system. The NP support library must be initialized before the SEND NP command can be used.

If the unit is saved, it must be readied with the RY command before the NP support library can be used. The NP support library must have function names mapped to specific library code files with the SL command. Otherwise, if a ready NP exists, the system returns an RSVP request saying that the NP support library is not usable and the system waits for an SL command or a DS command.

There are two types of NP unit-related command options: diagnostic and nondiagnostic. Diagnostic options can be entered only when a diagnostic NP support library is being used. In other words, the SEND NP diagnostic options cannot be used unless the NP support library is compiled with the DIAGNOSTICS option. Otherwise, an INVALID OPERATOR INPUT message is returned. Nondiagnostic options can be entered any time the NP support library is active. The following are the diagnostic options:

- PROGRAMDUMP
- ANALYZE
- RELEASE
- TANKING
- DIAGNOSTICS

The following are the nondiagnostic options:

- DUMP
- ID
- LOAD

- AUTOLOAD
- QUIT
- BLOCKTIMEOUT
- BLOCKING
- DASNAME
- MAXINPUTS
- FREEINPUTS
- INITIALREADS
- BUFFERSIZE

The values of these options are stored in the file *NP/SUPPORT/RECOVERY on the family specified by a DL OVERLAY specification. If the file is removed or the DL OVERLAY command specifies a family that does not contain an *NP/SUPPORT/RECOVERY file, a new file is created with the default options the next time the NP support library is initialized.

ANALYZE

ANALYZE <file name>

Analyze the *tank* file and print the analyzed file. If a file name is not specified, the active tank file is closed and renamed *NP/TANKED/INPUT/xxx/<date>/<time>*, it is analyzed and printed, and a new tank file is created. If a file name is specified, that file is analyzed.

AUTOLOAD

Causes the driver stack to try to load firmware as soon as the driver stack is initialized. To turn the option on or off, precede the keyword AUTOLOAD with SET or RESET. The default is SET.

BLOCKING

Causes the network processor to block all incoming messages to the host. If the value of the BLOCKING option is SET, incoming messages are sent to the buffer of the network processor. When this buffer is full, the contents of the buffer are sent to the host as one set of messages. If the value of the BLOCKING option is RESET, then every incoming message is immediately sent to the host.

The BLOCKING option can be set or reset at any time. To turn the option on or off, precede the keyword, BLOCKING, with SET or RESET. (The default value is SET.) The new setting does not take effect until a new *NP/CONTROLLER/nnn* stack is created for the unit (see "Considerations for Use") or the NPSUPPORT library is reinitialized.

SEND (Send Message) (cont.)

Reset blocking is not supported for ICP DLPs. If an attempt is made to reset blocking for an ICP DLP, then a message indicating an NPSUPPORT protocol is displayed. If this message appears, do the following to regain communication with the ICP DLP:

1. Set the blocking option for the DLP.
2. Reinitialize the ICP DLP.

BLOCKTIMEOUT
BLOCKTIMEOUT <blockval>

Display or set the amount of time (in milliseconds) the ICP should wait before sending a set of blocked messages to the host. The value of <blockval> can be changed at any time and takes effect immediately. The default value is 10 milliseconds.

BUFFERSIZE
BUFFERSIZE <bufsize>

Display or set the size of the input buffers for messages from the network processor to the host. The minimum <bufsize> is 480 bytes, the maximum size is 64000 bytes, and the default size is 4000 bytes. The new buffer size takes effect immediately.

DASNAME <dasname>
DASNAME = <dasname>

DASNAME is valid only for network processors (NPs). Use DASNAME to specify the name of the device application supervisor. The MCP uses DASNAME to determine the correct supervisor for the device.

Possible DASNAMEs are shown in the following table:

Processor Type	DASNAME
BNV2 NPs	INTRASYSTEM/BNV2ICPINITIALIZATION
Workstation DLPs	INTRASYSTEM/NTS_TWIXAX_INITIALIZATION
HLCN NPs	INTRASYSTEM/HLCNICPINITIALIZATION

If the supervisor for the device is not initialized, the operating system displays the following message on the ODT:

```
MISSING DIRECTORY CONNECTION <dasname>
```

DIAGNOSTICS

Displays diagnostic messages about the network processor system. To turn this option on or off, precede the keyword, DIAGNOSTICS, with SET or RESET. The default value is RESET.

DUMP

Causes a dump of the network processor. The internal memory state of the network processor is dumped into a file named DUMP/NP/MMDDYYYY, where MMDDYYYY is the date the DUMP request was made. A DUMP request is fatal to the network processor, because it clears the network processor.

FREEINPUTS

FREEINPUTS <freeval>

Display or set the minimum number of available input buffers. If the number of available input buffers falls below this value, the support library starts a task to overlay the buffers being used by the host (not the buffers in use by the network processor). Setting this number too high causes frequent overlays and reduces throughput. Setting it too low could force the network processor to wait for an available buffer, thus reducing throughput. The minimum value of <freeval> is 1, the maximum is 63, and the default value is 5.

The value of FREEINPUTS can be changed at any time but does not take effect until a new NP/CONTROLLER/nnn stack is created for the unit (see "Considerations for Use") or the NP support library is reinitialized.

ID

Displays compile and run-time information about the network processor. See the example for a sample display resulting from the ID message.

INITIALREADS

INITIALREADS <initval>

Display or set the number of initial read operations that the support library will issue to the network processor. With more than one read operation outstanding, the network processor can complete one I/O and begin transferring data with the next I/O while the system processes the first. This option maximizes the throughput of the network processor. The value of INITIALREADS can be changed at any time but does not take effect until a new NP/CONTROLLER/nnn stack is created for the unit (see "Considerations for Use") or the NP support library is reinitialized.

LOAD

Requests that the driver stack load firmware into the network processor. This message is used when AUTOLOAD has the value RESET.

MAXINPUTS

MAXINPUTS <maxval>

Display or set the number of input buffers to be used by the NP support library for a particular network processor. The minimum value is 10 buffers, the maximum is 127 buffers, and the default value is 35 buffers. The value of MAXINPUTS can be changed at any time but does not take effect until a new NP/CONTROLLER/nnn stack

is created for the unit (see "Considerations for Use") or the NPSUPPORT library is reinitialized.

PROGRAMDUMP

Causes a dump of the driver stack to a printer backup file (a backup disk, or BD, file). This file is labeled with the mix number of the frozen NP support library, for identification. PROGRAMDUMP is not fatal. The unit continues its scheduled activities after the dump.

QUIT

Requests that the driver stack terminate. Once QUIT is specified, no more connections to the driver stack are allowed. The driver stack waits for the active count to go to 0 (zero) before terminating.

RELEASE

Does the following:

- Closes the current tank file.
- Changes the file name to *NP/TANKED/INPUT/xxx/<date>/<time>*.
- Opens a new tank file.

This option is valid only when the system is in diagnostics mode.

SET <option>

RESET <option>

Sets or resets any of the following options: AUTOLOAD, BLOCKING, DIAGNOSTICS, and TANKING. For more information, see the description of the individual options.

TANKING

Causes all incoming messages to be stored in a tank file. The tank file is named *NP/TANKED/INPUT/<ICP unit number>*. The tank file is created on the family specified in the DL OVERLAY command. To turn this option on and off, precede the keyword, TANKING, with SET or RESET. The default value is RESET.

Example

This example shows the terminal display resulting from a SEND NP ID message:

SEND NP157: ID

```

NP SUPPORT LIBRARY:*SYSTEM/NPSUPPORT
VERSION: 36.112.218;COMPILED: 8/27/90 @ 13:44:32
FIRMWARE LEVEL: 000000000101
DASNAME: INTRASYSTEM/BNAV2ICPINITIALIZATION
BUFFERSIZE: 4000 BYTES
BLOCKTIMEOUT: 10 MILLISECONDS
MAXINPUTS: 30
FREEINPUTS: 5
INITIALREADS: 10
ACTIVE UNITS: 5
ACTIVE USERS: 23
WAITING USERS: 0
OUTPUT FILES: ON PACK
RUN TIME OPTIONS:
  AUTOLOAD: SET
  DIAGNOSTICS: RESET
  TANKING: RESET
  BLOCKING: SET
COMPILE TIME OPTIONS:
  DIAGNOSTICS

```

Considerations for Use

Certain changes specified in the SEND NP command (BLOCKING, FREEINPUTS, INITIALREADS, and MAXINPUTS) do not take effect until a new NP/CONTROLLER/*nnn* stack is created for the unit or the NP support library is reinitialized. To create a new stack, perform the following steps:

1. Issue a *SV NP <unit number>* command to make the network processor inaccessible to the system.
2. Issue a *CL NP <unit number>* command to clear the network processor.
3. Issue an *RY NP <unit number>* command to make the network processor ready for use.

Media Library System Commands

The following are valid media library system commands.

```

SEND TAPESERVER START
SEND TS START

```

Initiate the TAPESERVER library. An example of a TAPESERVER library is RoboHost A.

SEND (Send Message) (cont.)

SEND TAPESERVER START <text>
SEND TS START <text>

Initiate the TAPESERVER library and pass text to its initiation routine.

SEND TAPESERVER QUIT
SEND TS QUIT

Terminate the TAPESERVER library.

SEND TAPESERVER QUIT <text>
SEND TS QUIT <text>

Terminate the TAPESERVER library and send text to its termination routine.

SEND TAPESERVER STATUS
SEND TS STATUS

Display the status of the TAPESERVER library. This command can return the following states:

- INACTIVE
- LINKING
- INITIALIZING
- ACTIVE
- TERMINATING
- WAITING TO DELINK

SEND TAPEMANAGER START
SEND TM START

Initiate the TAPEMANAGER library.

SEND TAPEMANAGER START <text>
SEND TM START <text>

Initiate the TAPEMANAGER library and pass text to its initiation routine.

SEND TAPEMANAGER QUIT
SEND TM QUIT

Terminate the TAPEMANAGER library.

SENDMANAGER QUIT <text>
SEND TM QUIT <text>

Terminate the TAPEMANAGER library and pass text to its termination routine.

SEND TAPEMANAGER STATUS
SEND TM STATUS

Display the status of the TAPEMANAGER library. This command can return the following states:

- INACTIVE
- LINKING
- INITIALIZING
- ACTIVE
- TERMINATING
- WAITING TO DELINK

Example 1

This example initiates the TAPESERVER library:

SEND TS START

TAPESERVER WILL START

Example 2

This example stops the TAPEMANAGER library:

SEND TM QUIT

TAPEMANAGER WILL QUIT

Example 3

This example asks for the status of the TAPESERVER library:

SEND TS STATUS

TAPESERVER OPTION SET, ACTIVE, MIXNUMBER 892

As an alternate reply to example 2, you might also see the following:

TAPESERVER OPTION RESET, INACTIVE

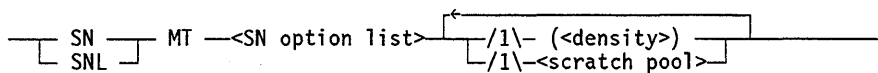
SN (Serial Number) and SNL (Serial Number Lock)

The SN (Serial Number) command purges and assigns a new or different serial number to a tape volume. It optionally specifies the recording density and assigns the tape to a scratch pool.

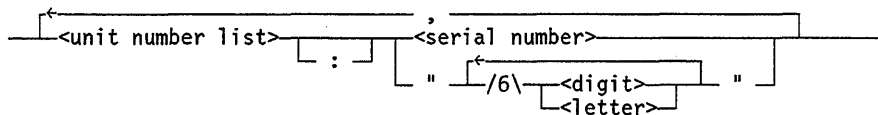
The tape drive unit where the tape volume is mounted must be ready, not in use, not marked as locked, and write-enabled.

If you want to retain the old serial number, use the PG (Purge) command.

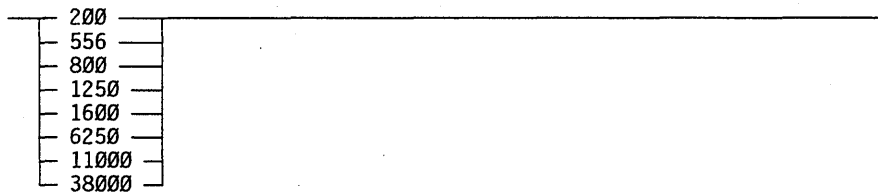
Syntax



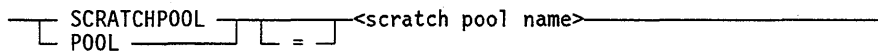
<SN option list>



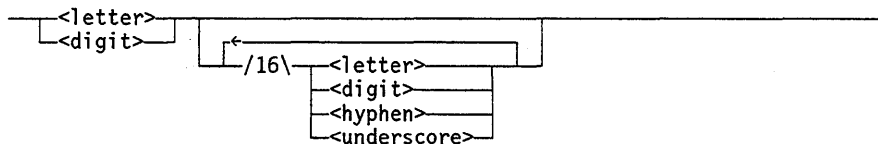
<density>



<scratch pool>



<scratch pool name>



Explanation

SN MT <SN option list>

Purges and assigns a serial number to the tapes mounted on the designated tape units. The serial number consists of up to six alphanumeric characters. If a number is used, it

is right-justified with leading zeros added. Any serial number containing alphabetic characters or any quoted alphanumeric string is left-justified with trailing blanks.

SN MT <SN option list> (<density>)

Purges and assigns the specified serial number and density (in bytes per inch) to all tapes mounted on the units in the unit number list.

If the density you specify is not valid for the unit, the system displays the following message:

MT <unit number> INVALID DENSITY SPECIFIED

In such a case, the tape density is not changed, but the system still purges the tape and assigns the specified serial number.

**SNL MT <SN option list>
SNL MT <SN option list> (<density>)**

Lock the specified tape units after purging, so that no job can automatically acquire the tapes as scratch tapes. The units must be readied through the RY (Ready) system command.

**SN MT <sn option list> SCRATCHPOOL = <scratch pool name>
SN MT <sn option list> POOL = <scratch pool name>**

Purge, assign a serial number, and assign a specified scratch pool name to the tape mounted on the designated unit.

**SNL MT <sn option list> SCRATCHPOOL = <scratch pool name>
SNL MT <sn option list> POOL = <scratch pool name>**

Purge, assign a serial number, and assign a specified scratch pool name to the tape mounted on the designated unit. This command also locks the tape after it is purged so that no task can access the tape.

As an alternative to SNL, you might want to use the HOLD command before you purge the tape with an SN command. You can then see the tape label information after the purge is completed while you prevent tasks from accessing the tape.

**SN MT <sn option list> (<density>) SCRATCHPOOL = <scratch pool name>
SN MT <sn option list> (<density>) POOL = <scratch pool name>**

Purge and assign the specified serial number, density, and scratch pool name to the tape.

SN (Serial Number) and SNL (Serial Number Lock) (cont.)

SNL MT <sn option list> (<density>) SCRATCHPOOL = <scratch pool name>

SNL MT <sn option list> (<density>) POOL = <scratch pool name>

Purge and assign the specified serial number, density, and scratch pool name to the tape. This command also locks the tape after it is purged so that no job can acquire the tape.

As an alternative to SNL, you might want to use the HOLD command before you purge the tape with an SN command. You can then see the tape label information after the purge is completed while you prevent tasks from accessing the tape.

Example 1

This example purges a tape mounted on unit 115, assigns a serial number called PART1 and assigns a density of 1600. The PER MT command verifies that all the requested actions were performed.

```
SN MT 115 PART1 (1600)
```

```
    MT 115 WILL BE SN-ED
```

```
PER MT
```

```
    115*P      [PART1 ] 1600      S C R A T C H
```

Example 2

This example purges a tape mounted on unit 115, assigns the tape the serial number 12345, and assigns the tape to a scratch pool named ACCOUNTING:

```
SN MT 115 12345 POOL = ACCOUNTING
```

```
    MT 115 WILL BE SN-ED
```

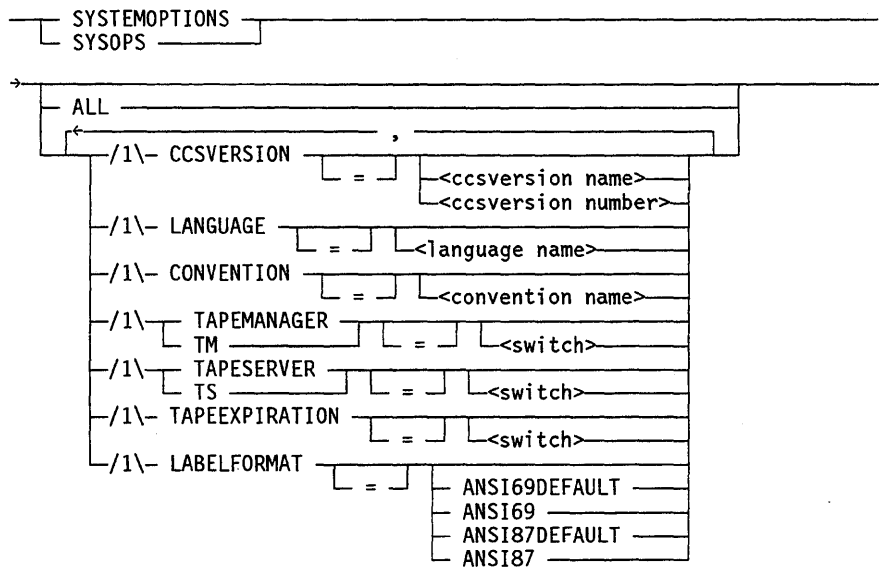
```
PER MT
```

```
    115*P      [012345] 1600      S C R A T C H POOL: ACCOUNTING
```

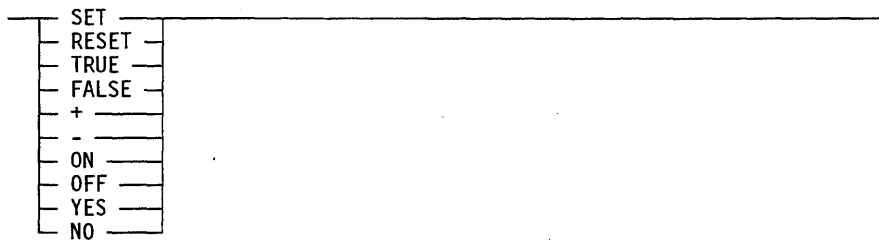
SYSOPS (System Options)

The SYSOPS (System Options) command is used to display or set the CCSVERSION, CONVENTION, LANGUAGE, TAPEMANAGER, TAPESERVER, TAPEEXPIRATION, and LABELFORMAT system options.

Syntax



<switch>



The ccsversion name, language name, or convention name can be within the range of 1 through 17 alphanumeric characters. The ccsversion number can be within the range of 0 through 65535.

Explanation

SYSTEMOPTIONS
SYSOPS

Display the current values for all the system options.

SYSOPS ALL

Displays the current values for all the system options and the possible values for these options.

SYSOPS CCSVERSION

SYSOPS CCSVERSION <ccsversion name>

SYSOPS CCSVERSION <ccsversion number>

Display or set the value for the CCSVERSION system option. If a new character set is specified, it is implemented after the next halt/load.

SYSOPS LANGUAGE

SYSOPS LANGUAGE <language name>

Display or set the value for the LANGUAGE system option. If a new system language is specified, it is implemented after the next halt/load.

SYSOPS CONVENTION

SYSOPS CONVENTION <convention name>

Display or set the value for the CONVENTION system option.

SYSOPS TAPEMANAGER

SYSOPS TAPEMANAGER <switch>

Display or modify the value for the TAPEMANAGER system option. The default value is RESET.

SYSOPS TAPESERVER

SYSOPS TAPESERVER <switch>

Display or modify the value for the TAPESERVER system option. The default value is RESET.

SYSOPS TAPEEXPIRATION

SYSOPS TAPEEXPIRATION <switch>

Display or set the value of the TAPEEXPIRATION system option. The default is TRUE. When the MCP reads the labels on a tape and this option is FALSE, the MCP ignores the expiration date of the tape. When the MCP reads the labels on a tape and this option is TRUE, the MCP treats the tape as a scratch tape if the current date is past the expiration date and the tape is write-enabled.

SYSOPS LABELFORMAT

Displays the ANSI standard being used to format tape labels when you create tapes. The default is ANSI69DEFAULT.

SYSOPS LABELFORMAT = ANSI69DEFAULT

If you choose this option, the ANSI69 standard is used to format tape labels when you create tape volumes. The option can be overridden by the setting of the LABELFORMAT task attribute. For information about the ANSI69 tape labels, refer to Appendix B.

SYSOPS LABELFORMAT = ANSI87DEFAULT

If you choose this option, the ANSI87 standard is used to format tape labels when you create tape volumes. The option can be overridden by the setting of the LABELFORMAT task attribute. For information about the ANSI87 tape labels, refer to Appendix B.

Note: A Series software at the Mark 39.1 or earlier release level is incapable of reading tapes written in ANSI87 format.

Example 1

Use the following example to display the current values for the system options:

SYSTEMOPTIONS

-----SYSTEM OPTIONS-----

The setting(s) for system options are as follows:

The current CCSVERSION is ASERIESNATIVE (Ø).
 The current LANGUAGE is ENGLISH.
 The current CONVENTION is ASERIESNATIVE.
 TAPESERVER is currently RESET.
 TAPEMANAGER is currently RESET.
 TAPEEXPIRATION is currently TRUE.
 LABELFORMAT is currently ANSI69DEFAULT.

Example 2

Use the following example to display all the current system options and all the available option values:

SYSOPS ALL

-----SYSTEM OPTIONS-----

The setting(s) for system options are as follows:

The current CCSVERSION is ASERIESNATIVE (Ø).
The current LANGUAGE is ENGLISH.
The current CONVENTION is ASERIESNATIVE.
TAPESERVER is currently RESET.
TAPEMANAGER is currently RESET.
TAPEEXPIRATION is currently TRUE.
LABELFORMAT is currently ANSI69DEFAULT.

The following object(s) are available for use in each system option category:

There are 12 CCSVERSIONs available. They are as follows:

ASERIESNATIVE	SWISS	SWEDISH1	SPANISH
CANADAEBDIC	CANADAGP	FRANCE	NORWAY
LATINGREEK	TURKISH	BRAZILIAN	CZECHOSLOVAKIA

The MCP BOUND LANGUAGE IS: ENGLISH

There is 1 CONVENTION available. The CONVENTION is ASERIESNATIVE.

The following system options may also be modified:

The valid values for TAPESERVER are SET and RESET.
The valid values for TAPEMANAGER are SET and RESET.
The valid values for TAPEEXPIRATION are TRUE and FALSE.
The valid values for LABELFORMAT are ANSI69DEFAULT and ANSI87DEFAULT.

Example 3

Use the following example to set the value NORWAY for the CCSVERSION system option:

SYSOPS CCSVERSION = NORWAY

-----SYSTEM OPTIONS-----

SYSTEM OPTION(S) are changed as follows:

The current CCSVERSION is ASERIESNATIVE (Ø).

The CCSVERSION following the next HALTLOAD will be NORWAY (71).

Example 4

Use the following example when you want the system to call a TAPEMANAGER library for tape file requests:

SYSOPS TAPEMANAGER = SET

-----SYSTEM OPTIONS-----

SYSTEM OPTION(S) are changed as follows:

TAPEMANAGER is being SET

Example 5

Use the following example to indicate that the MCP is to ignore the expiration dates on tapes when it reads tape labels:

SYSOPS TAPEEXPIRATION = FALSE

-----SYSTEM OPTIONS-----

SYSTEM OPTION(S) are changed as follows:

TAPEEXPIRATION is FALSE

Example 6

Use the following example when you want to indicate that all subsequent tapes are to be created with the ANSI87 standard format. Note that a LABELFORMAT task attribute specification can override the SYSOPS LABELFORMAT specification.

```
SYSOPS LABELFORMAT = ANSI87DEFAULT
```

```
-----SYSTEM OPTIONS-----
```

```
SYSTEM OPTION(S) are changed as follows:
```

```
    LABELFORMAT is ANSI87DEFAULT
```

Section 3

Programming Considerations

This section documents tape management enhancements that relate to programming. Topics include the following:

- New file attributes
 - AUTOUNLOAD
 - SCRATCHPOOL
- New and revised task attributes
 - LABELFORMAT (new)
 - HISTORYCAUSE (revised)
 - HISTORYREASON (revised)
- New and revised SETSTATUS calls
 - SEND (new)
 - HOLD (new)
 - PG and PGL (revised)
 - SN and SNL (revised)
- InfoGuard Security enhancements
- WFL considerations
- Logging enhancements

File Attributes

Two new file attributes, AUTOUNLOAD and SCRATCHPOOL, have been added to support tape management. Complete information follows.

Note: *The information in this discussion is based on the A Series File Attributes Programming Reference Manual.*

AUTOUNLOAD

AUTOUNLOAD

Kinds	Tape
Interrogate	Anytime
Modify	Anytime
Type	Mnemonic
Range	DONTCARE, ON, OFF
Default	DONTCARE
Stored Permanently	No
Parameters	None
FTAM File Access	Not Supported
Host Services Logical I/O	Not Supported
Port Services	Not Applicable

Use the AUTOUNLOAD attribute to control the behavior of a file close action on tape drives or to determine the behavior of the tape drive when a CLOSE operation is performed on the file. When the AUTOUNLOAD value of a tape file is ON, the associated tape volumes automatically unload when a program has either released the unit or performed a reel switch. A program can use the AUTOUNLOAD attribute to override the effects of the AUTOUNLOAD option of the system command MODE (Unit Mode).

The following are the possible mnemonic values, and their meanings:

Mnemonic	Action
ON	Causes a volume to unload automatically regardless of the mode of the unit according to the following forms of the file close action:
	CLOSE(F) Rewound/unloaded
	CLOSE(F,REWIND) Just rewind
	CLOSE(F,REEL) Rewound/unloaded
	CLOSE(F,PURGE) Rewound/unloaded
	CLOSE(F,*) Position retained
	CLOSE(F,LOCK) Rewound/unloaded
	REWIND(F) Just rewind
	F.OPEN:=FALSE Just rewind
	Block exit close Rewound/unloaded
	Implicit close on reel switch Rewound/unloaded

continued

continued

Mnemonic	Action
OFF	Prevents a volume from being automatically unloaded regardless of the MODE of the unit. This value causes the traditional semantics of the file close action to apply as follows:
	CLOSE(F) Just rewind
	CLOSE(F,REWIND) Just rewind
	CLOSE(F,REEL) Just rewind
	CLOSE(F,PURGE) Just rewind
	CLOSE(F,*) Position retained
	CLOSE(F,LOCK) Rewound/unloaded
	REWIND(F) Just rewind
	F.OPEN:=FALSE Just rewind
	Block exit close Just rewind
	Implicit close on reel switch Just rewind
DONTCARE	The default. This value indicates that the program does not care if the volume is automatically unloaded. The behavior taken at file close is based on the value of the AUTOUNLOAD option of the MODE (Unit Mode) system command.

If you interrogate this attribute, the current value of the attribute is returned. This value does not reflect the setting of the AUTOUNLOAD mode of a unit that might be associated with the logical file.

If you have a program that requires that a tape volume remain mounted after it closes a file, use CLOSE(F,REWIND) if file assignment to the unit is to be retained. If this version of CLOSE is not appropriate—that is, if the tape is to be accessed by a different logical file—you can set the AUTOUNLOAD file attribute to OFF before you execute the CLOSE(F) action. After the CLOSE(F) code, you can again set AUTOUNLOAD to DONTCARE, as shown in the following ALGOL example:

```

F.AUTOUNLOAD := VALUE(OFF);
CLOSE(F);
F.AUTOUNLOAD := VALUE(DONTCARE);
F.OPEN := TRUE
.
.
.
CLOSE(F);
    
```

It is possible to traverse a multivolume set without operator intervention by sequentially loading the tape cartridges in an automatic cartridge loader and by adding the following ALGOL code:

```
F.AUTOUNLOAD := VALUE(ON);
```

You can use the AUTOUNLOAD file attribute with the MODE AUTOUNLOAD system command to manage tape drive resources by unloading tapes when they are no longer needed. The drives then become available for use by other tasks. This factor is especially important in an automated tape library environment.

This file attribute does not change the syntax of the programming languages except for WFL. For information on WFL syntax changes, see "Tape Management Enhancements in WFL" later in this section.

SCRATCHPOOL

Kinds	Tape
Interrogate	Anytime
Modify	Anytime
Type	Character string
Range	See following text
Default	Null string
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Host Services Logical I/O	Not Supported
Port Services	Not applicable

Use the SCRATCHPOOL attribute to specify that the tape volume or volumes to be used when you create a new file are to come from a specific pool of scratch tapes. You can place a scratch tape in a specific pool by naming that pool when you use the PG (Purge) or SN (System Number) system command. The value of the SCRATCHPOOL attribute is an identifier of 1 to 17 EBCDIC characters, left-justified in a field of blanks (a hyphen or an underscore is not permitted as the first character). If you set the SCRATCHPOOL value equal to a null string, the tape volume selected is not a member of any named scratch pool.

If the SERIALNO file attribute has been specified for a particular member of a single volume or a multivolume set, the SCRATCHPOOL attribute is not taken into consideration when that volume is assigned.

If a tape volume is closed with the PURGE option, and the SCRATCHPOOL value is not a null string, the SCRATCHPOOL value is used as the scratch pool name for the volume in the resulting purge operation.

When you interrogate the SCRATCHPOOL attribute, the system always returns the specified value for that attribute. This action occurs because a tape volume is no longer considered a scratch tape when it is assigned.

Task Attributes

The following task attribute enhancements have been made to support tape management:

- A new task attribute, LABELFORMAT, has been added. This task attribute specifies the default label format for tape volumes created by a process.
- Value 14 has been defined for the HISTORYCAUSE task attribute. This value signifies a logical I/O error, as does value 8 (SOFTIOERRCAUSEV). The new value (SOFTIOERR2CAUSEV) enables the system to report additional logical I/O errors beyond the maximum of 256 that are already defined for SOFTIOERRCAUSEV.
- The following HISTORYREASON task attribute values have been defined for cases where the HISTORYTYPE task attribute value is 4 (DSEDV) and the HISTORYCAUSE value is 14 (SOFTIOERR2CAUSEV):
 - 12 = Reopen stopped by TAPEMANAGER library
 - 13 = Creation stopped by TAPESERVER library
 - 14 = Cannot open this file parameter
 - 15 = Assignment stopped by TAPEMANAGER library

The following pages present complete descriptions of each of these task attributes.

Note: *The information in this discussion is based on the A Series Task Attributes Programming Reference Manual.*

LABELFORMAT

Type:	Mnemonic
Range:	UNSPECIFIED, ANSI69, ANSI87
Default:	UNSPECIFIED
Read:	Anytime
Write:	Anytime
Inheritance:	From parent

Use the LABELFORMAT attribute to define the way tape volume labels written to tapes created by the task are to be formatted.

The attribute has three possible values, as follows:

LABELFORMAT Value	Format of Tape Output Produced by Task
ANSI87	Complies with ANSI X3.27 1987 standard
ANSI69	ANSI87 if SECURITYLABELS volume attribute is set to TRUE; otherwise complies with ANSI X3.27 1969 standard
UNSPECIFIED	ANSI87 if SECURITYLABELS volume attribute is set to TRUE; otherwise format defined by LABELFORMAT system option

Appendix B, “Standard Tape Formats,” illustrates the standard tape label formats supported by the A Series software.

You can define which label format is to be used when a tape volume is first opened for output. The format used for the first file on the tape defines the format for the entire volume set. Once a file has been written to the tape, any user-supplied label-formatting specifications for the tape are ignored.

You can use the LABELFORMAT attribute either to override an earlier specification – made with the LABELFORMAT system option or a previous LABELFORMAT task attribute specification – or to unconditionally request a particular label format for tape output produced by this task.

You control the LABELFORMAT system option with the SYSOPS (System Options) system command described in Section 2, “System Commands.”

The MCP determines which tape label format to use as follows:

- If the tape is assigned – that is, if one or more files have been written to it – all subsequent files written to that tape are written in the same format as the first file on the tape. The MCP ignores any user-supplied formatting specifications for the tape.
- The MCP uses ANSI87 format in writing the label when writing the first file (and subsequent files) in the volume set to that tape regardless of the settings of the LABELFORMAT system option and the LABELFORMAT task attribute if *all* the following situations are true:
 - The InfoGuard security subsystem is activated.
 - The TAPECHECK option is set to AUTOMATIC with the SECOPT (Security Options) system command.
 - The SECURITYLABELS volume attribute for the tape has a value of TRUE.
- If the SECURITYLABELS volume attribute is FALSE, the MCP uses ANSI87 format to write labels on tapes created by the task if *either* of the following situations is true:
 - The LABELFORMAT task attribute is set to ANSI87.
 - The LABELFORMAT task attribute is set to UNSPECIFIED and the LABELFORMAT system option is set to ANSI87DEFAULT.
- If the SECURITYLABELS volume attribute is FALSE, and you set the LABELFORMAT task attribute to ANSI69 but the LABELFORMAT system option is set to ANSI87DEFAULT, the MCP uses ANSI69 format in writing labels to the tape; that is, the task attribute overrides the system option.
- The LABELFORMAT system option has effect only when *both* of the following situations are true:
 - The SECURITYLABELS volume attribute is FALSE (the default value).
 - The LABELFORMAT task attribute is UNSPECIFIED (the default value).

The default value of the LABELFORMAT system option is ANSI69DEFAULT.

HISTORYCAUSE

Type	Mnemonic
Units	Not applicable
Range	See "Explanation" below
Default	0
Read Time	Anytime
Write Time	Never
Inheritance	None
Overwrite Rules	None (read-only)
Host Services	Supported
Attribute Number	67
Synonym	None
Restrictions	None

Explanation

The HISTORYCAUSE task attribute specifies the general type of condition that caused the process to terminate abnormally or to be suspended. The HISTORYCAUSE value is the same as field [15:08] of the HISTORY task attribute.

If the process did not terminate abnormally and is not suspended, the HISTORYCAUSE value is 0. No mnemonic is associated with this value.

If the process terminated abnormally, then the HISTORYTYPE value is 4 (DSEDV), and the HISTORYCAUSE value is one of the following:

Mnemonic Value	Integer Value	Meaning
(none)	0	The process has not been initiated, is still in use, or terminated normally.
OPERATORCAUSEV	1	The process was discontinued by an operator command.
PROGRAMCAUSEV	2	The process was deliberately terminated by a program action, or attempted an action that is not allowed by the operating system.
RESOURCECAUSEV	3	The process was terminated for exceeding a resource limit, such as MAXPROCTIME or MAXIOTIME.
FAULTCAUSEV	4	The process was terminated because it requested a machine operation that could not be executed, such as dividing by zero or reading past the end of an array.

continued

HISTORYCAUSE (cont.)

continued

Mnemonic Value	Integer Value	Meaning
SYSTEMCAUSEV	5	The process was terminated because it violated system parameters such as overlay row size and the amount of memory allowed.
DCERRCAUSEV	6	The process was terminated because of a data comm error.
IOERRCAUSEV	7	The process was terminated because of a physical I/O error.
SOFTIOERRCAUSEV	8	The process was terminated because of a logical I/O error.
NEWIOERRCAUSEV	9	The process was terminated because of an error in opening a file.
UNIMPLEMENTEDCAUSEV	10	The process was terminated because it attempted to use a feature that has not been implemented.
UNSPECIFIEDCAUSEV	11	The process was terminated because of an error of an unknown type.
EBDMSERRCAUSEV	12	The process was terminated because of a Data Management System II (DMSII) error.
NETWORKCAUSEV	13	The process was terminated because of a BNA error.
SOFTIOERR2CAUSEV or SOFTIOERR2V	14	The process was terminated because of a logical I/O error. This value enables additional I/O errors to be defined beyond the maximum number associated with value 8 (SOFTIOERRCAUSEV).

If the process is suspended, then the HISTORYTYPE value is 3 (STEDV), and the HISTORYCAUSE value is one of the following:

Mnemonic Value	Integer Value	Meaning
OPERATORCAUSEV	1	The process was suspended by an operator command.
PROGRAMCAUSEV	2	The process was suspended because of a missing resource.
SYSTEMCAUSEV	5	The process was suspended because of a shortage of available memory.
NETWORKCAUSEV	13	The process was suspended because of a BNA condition.

For a list of process termination messages and their relationships to HISTORYCAUSE values, refer to the discussion of process history in the *A Series Task Management Programming Guide*.

HISTORYREASON

Type	Mnemonic
Units	Not applicable
Range	See "Explanation" below
Default	0
Read Time	Anytime
Write Time	Never
Inheritance	None
Overwrite Rules	None (read-only)
Host Services	Supported
Attribute Number	68
Synonym	None
Restrictions	Not available in WFL

Explanation

The HISTORYREASON task attribute indicates the specific reason why a process terminated abnormally or was suspended. The HISTORYREASON value corresponds to field [23:08] of the HISTORY task attribute.

Most HISTORYREASON integer values have mnemonics associated with them. Each mnemonic briefly describes one reason this HISTORYREASON integer value could have occurred. You can determine which mnemonic applies in a particular case by comparing the HISTORYREASON integer value with the HISTORYTYPE and HISTORYCAUSE values.

If the process did not terminate abnormally and is not suspended, the HISTORYREASON value is 0. No mnemonic is associated with this value.

The standard method of reading mnemonic-valued task attributes might yield confusing results if applied to HISTORYREASON. The following is an ALGOL example of this method:

```
IF T.HISTORYREASON = VALUE(DIVIDEBYZEROV) THEN ...
```

The mnemonic DIVIDEBYZERO is associated with a HISTORYREASON value of 1. The expression shown in the example evaluates to TRUE whenever HISTORYREASON has a value of 1. However, a HISTORYREASON value of 1 indicates a DIVIDEBYZERO error only if HISTORYTYPE = DSEDV and HISTORYCAUSE = FAULTCAUSEV.

The following example shows a better method of reading HISTORYREASON. This example evaluates to TRUE only if a DIVIDEBYZERO error occurred:

```
IF T.HISTORYTYPE = VALUE(DSEDV)
   AND T.HISTORYCAUSE = VALUE(FAULTCAUSEV)
   AND T.HISTORYREASON = VALUE(DIVIDEBYZEROV) THEN ...
```

HISTORYREASON (cont.)

The following pages list the possible HISTORYREASON values for each combination of HISTORYTYPE and HISTORYCAUSE values. For HISTORYREASON values that have mnemonics, the mnemonics are listed under the column heading "Mnemonic Value." For HISTORYREASON values that do not have mnemonics, a short explanatory phrase is listed under the column heading "History Reason (No Mnemonic)."

HISTORYTYPE = 4 (DSEDV), HISTORYCAUSE = 6 (DCERRCAUSEV)

Integer Value	History Reason (No Mnemonic)
10	Message size error
12	Unknown file or station
13	File subtraction error

HISTORYTYPE = 4 (DSEDV), HISTORYCAUSE = 4 (FAULTCAUSEV)

Integer Value	Mnemonic Value
1	DIVIDEBYZEROV
2	EXPOVERFLOWV
3	EXPUNDERFLOWV
4	INVALIDINDEXV
5	INTEGEROVERFLOWV
6	INACTIVEQV
7	MEMORYPROTECTV
8	INVALIDOPV
9	LOOPV
10	MEMORYPARITYV
11	SCANPARITYV
12	INVALIDADDRESSV
13	STACKOVERFLOWV
14	STRINGPROTECTV
15	PROGRAMMEDOPV
16	FALSEASSERTV
17	SEQUENCEERRORV
18	INVALIDPCWV
19	STACKUNDERFLOWV
20	ZAPPEDV
21	LIBLINKERRORV
22	INVALIDINTV
23	MEMFAIL1V
24	PRIVILEGEDINSTRV
25	MEMPARANDFAIL1V

continued

continued

HISTORYTYPE = 4 (DSEDV), HISTORYCAUSE = 4 (FAULTCAUSEV)

Integer Value	Mnemonic Value
26	MEMORYFAIL2V
27	IOMERRV
28	PINTANDFAIL1V
29	PINTANDINVADDRV
30	PROCINTERNALV
35	PROCDIEDV
36	PROCFAILEDV
37	BCLPOINTERV
40	DISKPARITYV
41	EMODEVIOLATIONV
42	NOACTIVELINKV
43	PROCLINKPARITYV
44	BADTABLEEDITV
45	BOTTOMOFSTACKV
46	RUNLIGHTOUTV
47	STACKSTRUCTUREV

HISTORYTYPE = 4 (DSEDV), HISTORYCAUSE = 7 (IOERRCAUSEV)

Integer Value	History Reason (No Mnemonic)
0	Either a train printer I/O error occurred and could not be resolved, or the MCP procedure PATHRES did not successfully complete. PATHRES performs functions such as loading disk controller firmware.
6	Direct I/O attribute error

HISTORYTYPE = 4 (DSEDV), HISTORYCAUSE = 13 (NETWORKCAUSEV)

Integer Value	Mnemonic Value
1	NOHOSTV
3	HOSTHALTLOADEDV
5	HOSTNOTREACHABLEV
10	USERPORTERRORV
11	USERSIGNALERRORV
12	TASKPROTOCOLERRORV

HISTORYREASON (cont.)

HISTORYTYPE = 4 (DSEDV), HISTORYCAUSE = 9 (NEWIOERRV)

Integer Value	History Reason (No Mnemonic)
20	Data error—no label
37	Remote backup disk error
38	Unknown station
39	Invalid set of attributes

HISTORYTYPE = 4 (DSEDV), HISTORYCAUSE = 1 (OPERATORCAUSEV)

Integer Value	Mnemonic Value
0	RSVPV
1	CLEARUNITV
2	JUSTDSEDV

HISTORYTYPE = 4 (DSEDV), HISTORYCAUSE = 2 (PROGRAMCAUSEV)

Integer Value	Mnemonic Value
0	MISSINGCODEFILENAMEV
1	MISSINGCODEFILEV
3	INITACTIVETASKV
4	NOEXTERNALRUNV
5	VISITNONACTIVEV
6	ILLEGALVISITV
7	DYNCODEEOFV
8	BADD1STRECHV
9	ATTREADONLYV
11	NOTSESSIONNUMBERV
12	NONANCESTRALTASKFILEV
13	NOTIMPLEMENTEDV
14	INVALIDACCESSCODEV
15	INCOMPATIBLEBOXESV
18	DEATHINFAMILYV
19	CRITICALBLOCKV
20	BADGOTOV
23	INVALIDPARAMETERV
25	INCOMPATIBLECODEV
26	NOTEXECUTABLEV
27	UNMATCHEDPARAMSV
28	INVCOMPILERV

continued

continued

HISTORYTYPE = 4 (DSEDV), HISTORYCAUSE = 2 (PROGRAMCAUSEV)

Integer Value	Mnemonic Value
29	SECURITYERRORV
30	LIBMAINTV
31	ILLEGALTASKXFERV
32	BADRESIZEDEALLOCV
33	READONLYONACTIVEV
37	MISSINGINTRINSICV
38	INCOMPATIBLELEVELV
39	INFANTICIDEV
40	NOTBOUNDV
41	ILLEGALOWNARRAYV
42	DIMSIZERRORV
43	UPLEVELATTACHV
44	ILLEGALSWAPV
46	BADTASKATTRIBUTEV
47	MISSINGCARDDECKV
48	BADRESTARTV
49	BADEVENTUSAGEV
50	BADGIVELOCKV
51	BADGETLOCKV
52	ONLYMCSMAYSETV
56	NONOWNERACCESSV
57	COMPILERONLYV
58	TASKLIMITEXCEEDEDV
59	AXBADARRAYV
60	RUNTIMEWFLV
61	COMPILERERRORV
62	XSPARAMSV
63	SORTKILLV
64	MISSINGSYSTEMLIBV
65	ALREADYSELECTEDV
66	LIBMISSINGNAMEV
67	LIBTYPEMISMATCHV
68	LIBNOTINITIATEDV
69	CYCLICPROVISIONV

continued

HISTORYREASON (cont.)

continued

HISTORYTYPE = 4 (DSEDV), HISTORYCAUSE = 2 (PROGRAMCAUSEV)

Integer Value	Mnemonic Value
70	PREVIOUSLYFROZENLIBV
71	LIBIMPLEMENTATIONERRORV
72	FASTTASKFREEZEV
73	NONUNIQLIBV
74	SAVELIBTASKNEVERCALLEDV
75	LIBNEVERFROZEV
77	BADLIBTASKV
78	LIBFEATURENOTIMPLEMENTEDV
79	BADCOMPILERINDEXV
80	LIBNOTPROCESSEDORRUNV
81	LIBMUSTBESEPARATESTACKV
82	INVALIDPARAMV
83	FORTRANERRV
84	PLIRUNTIMEERRV
85	INTRINSICSERRV
86	MATHERRV
87	FORMATERRV
88	LIBDEIMPLEMENTATIONERRORV
89	LIBLEVELINCOMPATIBLEV
90	BADLIBTITLEV
91	CANTLINKTOASYSTEMLIBV
92	NOTASYSTEMLIBV
93	NOTLIBRARYCAPABLEV
94	LISTERRORV
95	LIBPARENTNOTALIBV
96	BADLIBFUNCTIONV
97	INVALIDATTVALUEV
98	UNAUTHORIZEDLIBUSEV
99	FOREIGNTASKINITFAILV
100	PORTERRORV
101	LIBCANCELERRV
102	INVALIDSAVECORELIMITV
103	NONVISTASKFILEV
104	BADINSCRIBEV

continued

continued

HISTORYTYPE = 4 (DSEDV), HISTORYCAUSE = 2 (PROGRAMCAUSEV)

Integer Value	Mnemonic Value
105	BADERASEV
106	CLIENTDIEDINACRV
107	BADPOBOXUSAGEV
108	INVALIDSTKNOV
109	BADTCPREQV
110	BYRESTRICTIONV
111	LIBWRONGMARKLEVELV
112	NOINITIATORV
113	UPELVETASKASSIGNV
114	FRAMEEXCEEDEDV
115	CODEFILEINCOMPATIBLEWITHMCPV
116	CODEFILENOTACTIVEV
117	BADPPBV
118	NOTAFASTTASKV
119	FASTTASKINITTASKV
120	FASTTASKNOPARAMSV
121	FASTTASKNOTADSV
122	FASTTASKBADEXCEPTIONTASKV
123	STACKHASFAMILYV
124	FASTTASKFAULTEDV
125	DATABASEDIEDV
126	LIBRARYDIEDV
127	STACKHASUNITATTACHEDV
128	RESTRICTEDACCESSV
129	ATTWRITEONLYV
130	NONANCESTRALEXCEPTEVENTV
131	INCORRECTSYNTAXV
132	ATTACCESSFAULTV
133	INVALIDLSNV
134	DATACOMMNOTACTIVEV
135	VALUETOOLARGEV
136	PRIVILEGEREQUIREDV
137	NONLOCALACCEPTEVENTV
138	INVSCHEDACTV

continued

HISTORYREASON (cont.)

continued

HISTORYTYPE = 4 (DSEDV), HISTORYCAUSE = 2 (PROGRAMCAUSEV)

Integer Value	Mnemonic Value
139	INVTIMESTATV
140	INVREACTIVATEV
141	INVSOURCEV
142	INVDUMPPARAMV
143	INVCPMACTIONV
144	INVPREFACTIONV
145	INVDISCONNECTV
146	INVDESTINATIONV
147	BLOCKHASNOSCWV
148	PRPROVIDERGONEV
149	LIBWRONGCODEFILEV

HISTORYTYPE = 4 (DSEDV), HISTORYCAUSE = 3 (RESOURCECAUSEV)

Integer Value	Mnemonic Value
0	PROCESSEXCEEDEDV
1	IOEXCEEDEDV
2	STACKEXCEEDEDV
3	PRINTEXCEEDEDV
4	PUNCHEXCEEDEDV
5	CARDREADEXCEEDEDV
6	MEMORYEXCEEDEDV
8	TAPEEXCEEDEDV
9	WAITEXCEEDEDV
10	ELAPSEDEXCEEDEDV
11	DISKLIMITEXCEEDEDV
12	STRINGPOOLEXCEEDEDV
14	SAVECORELIMITEXCEEDEDV
15	CAUEXCEEDEDV
16	SEGLIMITEXCEEDEDV

HISTORYTYPE = 4 (DSEDV), HISTORYCAUSE = 8 (SOFTIOERRCAUSEV)

Integer Value	History Reason (No Mnemonic)
0	No error
1	Label parity error

continued

continued

HISTORYTYPE = 4 (DSEDV), HISTORYCAUSE = 8 (SOFTIOERRCAUSEV)

Integer Value	History Reason (No Mnemonic)
2	Parity error on position
3	Invalid translation
4	Incompatible blocking
5	Invalid output reverse
6	Invalid input reverse
7	Short tape blocking
8	Invalid output file
9	No buffer space
10	No space in header
11	Duplicated file
12	Invalid direct I/O
14	Exceeded resources
15	No unit
16	Invalid optional file
17	Invalid final reel
18	Too many names
19	Failed entry
20	Invalid MYUSE value
21	Invalid NEWFILE value
22	DCOPEN failed
23	No write ring
24	Failed volume entry
25	Invalid unlabeled volume
26	Invalid BLOCKSTRUCTURE or FILETYPE
27	Invalid reel number
28	Find routines failed
29	Invalid backward seek
30	Invalid read reverse
31	Invalid seek
32	Parity error on seek
33	Read on output file
34	Read on unopened file

continued

HISTORYREASON (cont.)

continued

HISTORYTYPE = 4 (DSEDV), HISTORYCAUSE = 8 (SOFTIOERRCAUSEV)

Integer Value	History Reason (No Mnemonic)
35	Read reverse on unopened file
36	Seek on unopened file
37	Space forward on output file
38	Write on code file
39	Write on input file
40	Write on unopened file
41	Buffer in use
42	Up-level event
43	Security error
44	No room for buffer
45	Unknown error
46	Logic error
47	Already closed
48	No read before rewrite
49	No read before delete
50	Delete on non-I/O file
51	Invalid update file
52	Incompatible file organization
53	Close not called
54	File information block (FIB) stack transition error
55	Locking error
56	Kind list not allowed
57	Dialogue communication failure with other host
59	File not removed on disk
60	File not cataloged
61	Checkpoint file title not changed
62	Write user label error
63	RELEASEHEADER error
64	Tried to write beyond end of file (EOF)
65	Rewrite on non-I/O file
66	Logical/physical file mismatch

continued

continued

HISTORYTYPE = 4 (DSEDV), HISTORYCAUSE = 8 (SOFTIOERRCAUSEV)

Integer Value	History Reason (No Mnemonic)
67	Seek on output file
68	Tape position error
69	Distributed systems service (DSS) cannot handle this file.
70	Access restricted to A Programming Language (APL)
71	Open after close with lock
72	Invalid write random
73	Invalid read random
74	Not closed
75	Unexpected I/O error
78	Data error
79	Deleted/duplicate record
80	Parity error
81	I/O not done
82	Invalid subfile
83	Broadcast read error
84	Subfile closed
85	No available buffer
86	No available message
87	Port not connected
88	End of file (EOF)
89	Invalid short block read
90	Break on output
91	Unit in rewind
92	Time limit exceeded
93	File not available
94	No file
95	Mismatched genealogy
96	Mismatched serial number
97	File not resident
98	Pack not present
99	Invalid access code

continued

HISTORYREASON (cont.)

continued

HISTORYTYPE = 4 (DSEDV), HISTORYCAUSE = 8 (SOFTIOERRCAUSEV)

Integer Value	History Reason (No Mnemonic)
100	Foreign file open error
101	Port offer error
102	Invalid hostname for foreign file
103	Data might have been lost.
104	Record count error
105	Block count error
106	Host not reachable
107	Write lockout
108	FRAMESIZE and INTMODE values incompatible
109	Binary I/O not allowed
110	End of page
111	BCL not allowed on this machine
112	No continuation pack for audited file
113	Cannot be audit file
114	I/O error occurred during flushing of buffers.
115	Too many backup files
116	Maximum audit length exceeded
117	Unable to position file at end
118	Unsupported function
119	Bad use of use routines
120	Must have usercode to use DSS
121	Invalid port name
122	Requires direct I/O
123	SB must contain a disk/pack unit.
124	EIO logic error
125	Invalid array index
126	Incompatible I/O length
127	SIZEVISIBLE/FRAMESIZE/INTMODE value conflict
128	I/O error occurred during closing of file.
129	I/O support library error

continued

continued

HISTORYTYPE = 4 (DSEDV), HISTORYCAUSE = 8 (SOFTIOERRCAUSEV)

Integer Value	History Reason (No Mnemonic)
130	I/O error
131	INQ_LIST allocation failed
132	End file not allowed
133	I/O error changing host control (HC) unit access mask register (AMR)
134	Unsupported protocol type
135	Protocol error
136	No resource to open port
137	YOURHOST is not in YOURHOSTGROUP.
138	User is not an authorized user of the application group.
139	Support library unavailable
140	Error in one or more port-subfiles open operations
141	Error in one or more port-subfiles close operations
142	Incompatible attribute value or values
143	Function not available
144	Unacceptable character set
145	Networking not supported
146	TRANSLATE=FORCESOFT not allowed with binary I/O
147	I/O error clearing adapter or unit
148	Access restrictions not met
149	Cannot create restricted file
150	Security error on output tape open
151	Cannot write on guard file
152	Logical I/O not supported for this type of unit
153	Attribute already set in physical file
154	FAMILYOWNER conflicts with task usercode.
155	Invalid I/O to coactive disk
156	Coactive unit not in output mode

continued

HISTORYREASON (cont.)

continued

HISTORYTYPE = 4 (DSEDV), HISTORYCAUSE = 8 (SOFTIOERRCAUSEV)

Integer Value	History Reason (No Mnemonic)
157	Incompatible with this MCP version
158	DSS dialogue number too large for logical I/O
159	I/O error occurred during closing of file.
160	Tape-drive-mode-change operation failed.
161	BYTES is not supported by this unit.
162	Random add not allowed unless delete-capable
163	Not delete-capable
164	Record has not been read
165	Beyond extend area
166	Record not locked
167	Record position occupied
168	Sequential write not permitted to EFS direct file
170	Attempt to exceed family limit
171	Family integral limit exceeded
172	Attempt to exceed temporary file limit
174	Invalid write option specified
175	Invalid specification of ANYSIZEIO
176	Area length exceeds maximum allowed.
177	Logical file INTMODE incompatible with permanent file FRAMESIZE
178	Incompatible FILESTRUCTURE
179	Permanent file FILESTRUCTURE must be STREAM.
180	Logical file MAXRECSIZE inconsistent with permanent file MAXRECSIZE
181	Logical file BLOCKSIZE inconsistent with permanent file BLOCKSIZE
182	MAXRECSIZE exceeds AREALENGTH.
183	Logical file FRAMESIZE incompatible with permanent file area length
184	Unsupported parameter for this service

continued

continued

HISTORYTYPE = 4 (DSEDV), HISTORYCAUSE = 8 (SOFTIOERRCAUSEV)

Integer Value	History Reason (No Mnemonic)
185	Local interprocess communication (IPC) not supported for this service
186	Unsupported translation for this service
187	DIOFILESTRUCTURE value requires FILESTRUCTURE to be set.
188	Not in proper state for direct I/O to unit EIO
189	Cannot access a file of this FILEKIND
190	Open rejected by correspondent
191	Close rejected by correspondent
192	Endpoint not registered
193	Invalid respond option
194	Service invalid for provider
195	Provider restricted
196	Connect time limit exceeded
197	Correspondent does not support APPLICATIONCONTEXT value.
198	Correspondent rejected DEFAULTPCONTEXT value.
199	Invalid value or values for DEFINEDPCONTEXTSET
200	Warning—DEFINEDPCONTEXTSET values have changed.
201	Warning—port attribute ignored
202	Invalid associated data
203	Associated data too long
204	Invalid attribute values for respond
205	Invalid attribute values for AWAITOPEN
206	DEPENDENTSPECS must be TRUE when DIOFILESTRUCTURE=SECTORSTREAM.
207	Warning—initiator close collision
208	Warning—responder close collision
209	Endpoint incompatible with service
210	Unsupported primitive
211	Open failure in KEYEDIO library

continued

HISTORYREASON (cont.)

continued

HISTORYTYPE = 4 (DSEDV), HISTORYCAUSE = 8 (SOFTIOERRCAUSEV)

Integer Value	History Reason (No Mnemonic)
212	Read reverse is not supported by this unit.
213	Specified MAXRECSIZE is not supported by this unit.
214	MAXRECSIZE must equal BLOCKSIZE for this unit.
215	MCP does not support DMSII use of this FILESTRUCTURE.
216	Insufficient disk space
217	Operator entered OF (Optional File) system command.
218	KEYEDIOII write error occurred.
219	Unmatched DIOFILESTRUCTURE value
220	Invalid connect TIMELIMIT value
221	Error encoding data
222	No data available to be read
223	Error on broadcast write
224	No buffer available for write
225	Message truncated warning
226	Open data was received.
227	Open response data was received.
228	Close request data was received.
229	Close abort data was received.
230	Close response data was received.
231	More data to come
232	Fault in use routine
233	Logical file MINRECSIZE inconsistent with permanent file MINRECSIZE
233	Logical file MINRECSIZE inconsistent with permanent file MINRECSIZE
234	BASICSERVICE violation
235	Action not valid in this FILESTATE
236	Transparent LOCALSYNTAX cannot be supported for this subfile.
237	Open aborted by correspondent.
238	Open rejected—transient

continued

continued

HISTORYTYPE = 4 (DSEDV), HISTORYCAUSE = 8 (SOFTIOERRCAUSEV)

Integer Value	History Reason (No Mnemonic)
239	Warning: APPLICATIONCONTEXT and/or presentation context or contexts ignored
240	Warning: abort request ASSOCIATEDDATA ignored
241	Requested PROVIDERGROUP not defined
242	Warning: PROVIDERGROUP must be defined before request can complete.
243	Warning: initiating provider
244	Write on read-only file
245	During a file open, either the CENTRALSUPPORT library could not be accessed, or CENTRALSUPPORT reported an error related to CCSVERSION validation, INTMODE/EXTMODE validation, or translation tables availability.
248	Operation requires ownership of all available tokens.
249	Tape was changed while it was assigned.
250	Word oriented access not supported for KIND=CD
251	NETBIOS name in use
252	Warning: switching between read and write might result in program being discontinued.

HISTORYTYPE = 4 (DSEDV), HISTORYCAUSE = 14 (SOFTIOERR2CAUSEV)

Integer Value	Mnemonic Value
12	Reopen stopped by TAPEMANAGER library
13	Creation stopped by TAPESERVER library
14	Cannot open this file parameter
15	Assignment stopped by TAPEMANAGER library

HISTORYREASON (cont.)

HISTORYTYPE = 4 (DSEDV), HISTORYCAUSE = 5 (SYSTEMCAUSEV)

Integer Value	Mnemonic Value
1	NOMEMV
2	PARITYONPBIV
3	ARRAYTOOLARGEV
4	INCOMPATIBLEWFLJOBFILEV
6	MFTABORTV
7	SYSTEMEIV

HISTORYTYPE = 4 (DSEDV), HISTORYCAUSE = 10 (UNIMPLEMENTEDCAUSEV)

Integer Value	Mnemonic Value
1	DYNAMICOWNARRAYV
2	DCHOLDV

HISTORYTYPE = 3 (STEDV), HISTORYCAUSE = 13 (NETWORKCAUSEV)

Integer Value	Mnemonic Value
1	DISCONNECTEDV
3	SUSPENDEDV

HISTORYTYPE = 3 (STEDV), HISTORYCAUSE = 1 (OPERATORCAUSEV)

Integer Value	Mnemonic Value
0	(No mnemonic. This value means the command was entered from an ODT.)
1	REMOTELYCAUSEDV

HISTORYTYPE = 3 (STEDV), HISTORYCAUSE = 2 (PROGRAMCAUSEV)

Integer Value	Mnemonic Value
1	RESPONSEREQUIRED

HISTORYTYPE = 3 (STEDV), HISTORYCAUSE = 5 (SYSTEMCAUSEV)

Integer Value	Mnemonic Value
1	WSSIZEEXCEEDV

Examples

Suppose that the following task attributes have the values shown:

```
HISTORYTYPE = 4 (DSEDV)
HISTORYCAUSE = 4 (FAULTCAUSEV)
HISTORYREASON = 1
```

In this context, a HISTORYREASON of 1 means DIVIDEBYZEROV. In other words, the process was discontinued because it attempted to divide by zero.

Now suppose that these task attributes have the following values:

```
HISTORYTYPE = 4 (DSEDV)
HISTORYCAUSE = 3 (RESOURCECAUSEV)
HISTORYREASON = 1
```

In this context, a HISTORYREASON of 1 means IOEXCEEDEDV. In other words, the process was discontinued because it used more I/O time than was allowed by its MAXIOTIME task attribute value.

SETSTATUS Changes

The following SETSTATUS changes have been made to support tape management:

- The SEND TAPEMANAGER and SEND TAPESERVER calls have been added to SETSTATUS miscellaneous requests.
- The following changes have been made to SETSTATUS unit type requests:
 - HOLD has been added as a new call.
 - Changes have been made to the PG/PGL and SN/SNL calls.

Complete information for each of these calls follows.

Note: *The information in this discussion is based on the A Series GETSTATUS/SETSTATUS Programming Reference Manual.*

SEND TAPEMANAGER and SEND TAPESERVER Calls

Use these calls to initiate or to terminate either the TAPEMANAGER or the TAPESERVER library.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	2
SUBTYPE	88
VAL	1 = SEND TAPESERVER 2 = SEND TAPEMANAGER
ARRAYROW	A[0] = 2 A[1] = 1 if START A[1] = 2 if QUIT A[1] = 3 if STATUS A[2] = Length in bytes of optional text A[3] and following = optional text

The following information describes words [2] through [4] that SETSTATUS returns in the array:

Word	Status
A[2]	The library state, enumerated as follows: 0 = INACTIVE 1 = LINKING 2 = INITIALIZING 3 = ACTIVE 4 = TERMINATING 5 = WAITING TO DELINK
A[3]	0 = NOT IN THE MIX, otherwise, the mix number
A[4]	The library option, enumerated as follows: 1 = SET 2 = RESET

Call

```
RSLT := SETSTATUS (2, 88, VAL, A);
```

Results

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
4	PREVIOUSLY DONE
58	NOT ALLOWED
60	INVALID PARAM INDEX
63	PARAM LENGTH ERROR
69	INVALID VALUE
95	OPTION IS NOT SET

In addition, SETSTATUS can return Hard Error 57 for a SEND TM STATUS request if the array is not large enough to contain the response.

Example

For the system command SEND TAPESEVER START, use the following input:

```
A[0] :=2  
A[1] :=1  
A[2] :=0  
RSLT := SETSTATUS (2, 88, 1, A);
```

HOLD Call

Use this call either to prevent the system from assigning files to tape units or to allow the system to assign files to a unit that was previously the subject of a HOLD call.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	5
SUBTYPE	35
VAL	0 = HOLD – 1 = HOLD
ARRAYROW	A[0] = Length of request A[1] to A[A[0]-1] = Unit list

Each entry in the unit number list is as follows:

Array A	Message
A[n].[31:08]	Unit type; 13 = MT
A[n].[38:06]	Number of additional words; must be 1
A[n+1]	Unit number

Call

```
RSLT := SETSTATUS (5, 35, VAL, A);
```

Results

Bit 0 of the result is turned on if either a hard error or a soft error occurs. Bits [11:08] indicate the hard error. If bits [11:08] are equal to 0, a soft error occurred.

SETSTATUS can return the following soft errors for this call:

Soft Error Number	Message
24	UNIT IN USE
41	INVALID UNIT TYPE
167	PREVIOUSLY DONE

SETSTATUS Request Type 5 (Unit Requests) (cont.)

Example

For the system command *HOLD MT 56*, use the following input:

```
A[0] := 3;  
A[1] := 0 & 1 [38:06] & 13 [31:08];  
A[2] := 56;  
RSLT := SETSTATUS (5, 35, 1, A);
```

PG and PGL Calls

Use the PG call to purge tape, disk pack, or host control units. Use the PGL call to lock the tape units after they have been purged (the unit type must be MT). PG is not valid for use with online mirrored packs. You must release the mirrored pack before the PG call can be issued.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	5
SUBTYPE	2
VAL	[0:1] = 1 if lock (PGL) [1:1] = 0 [7:6] = 0 [15:8] = Density code: <ul style="list-style-type: none"> ● 0 = Not specified (Density will not be changed.) ● 1 = 800 bytes per inch ● 2 = 556 bytes per inch ● 3 = 200 bytes per inch ● 4 = 1600 bytes per inch ● 5 = 6250 bytes per inch ● 6 = 38000 bytes per inch ● 7 = 1250 bytes per inch ● 8 = 11000 bytes per inch
	[45:01] = 1 if a SCRATCHPOOL is to be assigned
	[46:01] = 1 if PG is to be performed synchronously
ARRAYROW	A[0] = 2 for MT or HC and 3 for PK A[1].[31:08] = Unit type: <ul style="list-style-type: none"> ● 13 : MT ● 17 : PK ● 20 : HC
	A[1].[38:06] = Number of additional words; must be 1 for MT or HC
	A[2] = Unit number

For a PG MT request, if bit [45:01] = 1, words 3 through 5 contain the pool name in substandard form.

SETSTATUS Request Type 5 (Unit Requests) (cont.)

For a PG PK request, the next few words of the entry in the unit number list are used to designate the OLDNAME. The first three bytes in word 3 must be specified for PG PK requests even if there is no OLDNAME (see the example).

Array A	Message
A[3].[47:08]	Byte length from this byte to the end of entry
A[3].[39:08]	0
A[3].[31:08]	Number of specifications (0 or 1)
A[3].[23:08]	Byte length from the next byte to the end of the entry
A[3].[15:08]	11 (OLDNAME code)
A[3].[07:08]	Beginning of the family name
A[4] through end of entry	Continuation of the family name

Call

```
RSLT := SETSTATUS (5, 2, VAL, A);
```

Results

SETSTATUS can return the following soft errors for PG and PGL:

Soft Error Number	Message
11	ADDITIONAL INFO NEEDED
14	UNIT LOCKED
41	INVALID UNIT TYPE
45	NAME REQUIRED
56	INVALID SYNTAX
59	NAME TOO LONG
77	INCOMPLETE ENTRY
92	REQUIRED LOCK IN USE
144	SYNCHRONOUS UNIT REQUEST FAILED
167	INVALID FOR EXTERNAL IO UNITS
193	CANNOT PG A MIRRORED UNIT

Soft errors are reported in word 1 of the ARRAYROW in bits [46:08]. For soft error 144, a result is returned in bits [15:16] of the word. The error values are as follows:

- 1 = The tape was not purged for a reason related to the volume directory.
- For values 88 through 96, 104, 115, and 177 through 187, refer to the table that follows:

Value	Related Message
88	WRITE LOCK OUT
89	SN REQUIRED
90	WRITE PARITY ERROR
91	WRITE ERROR
92	UNIT NOT READY
93	REQUIRES VOLUME DELETE
94	UNIT REWINDING
95	UNABLE TO OBTAIN UNIT STATUS
96	INVALID DENSITY SPECIFIED
104	UNABLE TO SET DENSITY
115	WARNING – DATA SECURITY ERASE FAILED TO COMPLETE
177	PURGE REQUEST REJECTED BY TAPEMANAGER
178	PURGE REQUEST REJECTED BY TAPESERVER
179	INVALID SCRATCHPOOL SYNTAX
180	PURGED, BUT NOT CONFIRMED BY TAPESERVER
181	SERIAL NUMBER REJECTED BY TAPEMANAGER
182	SERIAL NUMBER REJECTED BY TAPESERVER
183	SCRATCHPOOL REJECTED BY TAPEMANAGER
184	SCRATCHPOOL REJECTED BY TAPESERVER
185	PURGE ATTEMPT DSED IN TAPEMANAGER LIBRARY
186	PURGE ATTEMPT DSED AWAITING TAPESERVER LIBRARY
187	PURGE ATTEMPT DSED IN TAPESERVER LIBRARY

Examples

For the system command *PG MT 30*, use the following input:

```

A[0] := 3;
A[1] := 0 & 13 [31:08]      %% MT unit type
        & 1 [38:06];      %% an additional word
A[2] := 30;                %% unit number
RSLT := SETSTATUS(5,2,0,A);
    
```

SETSTATUS Request Type 5 (Unit Requests) (cont.)

For the system command *PGL MT 160 (1600)*, use the following input:

```
A[0] := 3;
A[1] := 0 & 13 [31:08]      %% MT unit type
      & 1 [38:06];        %% an additional word
A[2] := 160;                %% unit number
RSLT := SETSTATUS (5, 2,
                  1 &      %% PGL
                  VALUE (BPI1600) + 1 [15:8], %% 1600 BPI
                  A);
```

For the system command *PG PK 55*, use the following input:

```
A[0] := 4;                %% total words
A[1] := 0 & 17 [31:08]    %% PK unit type
      & 2 [38:06];        %% 2 additional words
A[2] := 55;                %% unit number
REPLACE POINTER (A[3], 8) BY
  48"03"                    %% total length
  48"00"                    %% 0
  48"00";                   %% 0 (no specs)
RSLT := SETSTATUS (5, 2, 0, A);
```

For the system command *PG PK 45 OLDNAME = USERPACK*, use the following input:

```
A[0] := 6;                %% total words
A[1] := 0 & 17 [31:08]    %% PK unit type
      & 4 [38:06];        %% 4 additional words
A[2] := 45;                %% unit number
REPLACE POINTER (A[3], 8) BY
  48"0D"                    %% total length
  48"00"                    %% 0
  48"01"                    %% one spec
  48"09"                    %% length of spec
  48"0B"                    %% OLDNAME
  8"USERPACK";              %% value
RSLT := SETSTATUS (5, 2, 0, A);
```

SN and SNL Calls

Use these calls to purge and to assign serial numbers to tape volumes and to designate their recording density.

Input

Use the following input for this call:

Parameter	Value or Description
TYPE	5
SUBTYPE	2
VAL	[0:1] = 1 If lock (SNL) [1:1] = 1 [7:6] = 0 [15:8] = Density code: <ul style="list-style-type: none"> ● 0 = Not specified (Density will not be changed.) ● 1 = 800 bytes per inch ● 2 = 556 bytes per inch ● 3 = 200 bytes per inch ● 4 = 1600 bytes per inch ● 5 = 6250 bytes per inch ● 6 = 38000 bytes per inch ● 7 = 1250 bytes per inch ● 8 = 11000 bytes per inch [45:01] = 1 if a SCRATCHPOOL is to be assigned [46:01] = 1 if SN is to be performed synchronously
ARRAYROW	A[0] = 4 A[1] = 0 & 2 [38:06] & unit type [31:08] (Refer to Appendix C for a list of unit type codes.) A[2] = Unit number A[3] = Serial number as 6 EBCDIC characters A[4] to A[6] = if bit [45:01] = 1, pool name in substandard form

Call

```
RSLT := SETSTATUS (5, 2, VAL, A);
```

SETSTATUS Request Type 5 (Unit Requests) (cont.)

Results

SETSTATUS can return the following soft errors that are applicable to SN and SNL:

Soft Error Number	Message
10	INVALID NUMBER
11	ADDITIONAL INFO NEEDED
14	UNIT LOCKED
41	INVALID UNIT TYPE
45	NAME REQUIRED
56	INVALID SYNTAX
57	CANNOT BE SNED
59	NAME TOO LONG
77	INCOMPLETE ENTRY
92	REQUIRED LOCK IN USE
144	SYNCHRONOUS UNIT REQUEST FAILED
167	INVALID FOR EXTERNAL IO UNITS
193	CANNOT PG A MIRRORED UNIT

Soft errors are reported in word 1 of the ARRAYROW in bits [46:08].

For soft error 144 (SYNCHRONOUS UNIT REQUEST FAILED), a result is returned in bits [15:16] of the word. The error values are as follows:

- 1 = The tape was not purged for a reason related to the volume directory.
- For values 88 through 96, 104, 115, and 177 through 187, refer to the table that follows:

Value	Related Message
88	WRITE LOCK OUT
89	SN REQUIRED
90	WRITE PARITY ERROR
91	WRITE ERROR
92	UNIT NOT READY
93	REQUIRES VOLUME DELETE
94	UNIT REWINDING
95	UNABLE TO OBTAIN UNIT STATUS
96	INVALID DENSITY SPECIFIED
104	UNABLE TO SET DENSITY
115	WARNING – DATA SECURITY ERASE FAILED TO COMPLETE
177	PURGE REQUEST REJECTED BY TAPEMANAGER
178	PURGE REQUEST REJECTED BY TAPESERVER

continued

continued

Value	Related Message
179	INVALID SCRATCHPOOL SYNTAX
180	PURGED, BUT NOT CONFIRMED BY TAPESERVER
181	SERIAL NUMBER REJECTED BY TAPEMANAGER
182	SERIAL NUMBER REJECTED BY TAPESERVER
183	SCRATCHPOOL REJECTED BY TAPEMANAGER
184	SCRATCHPOOL REJECTED BY TAPESERVER
185	PURGE ATTEMPT DSED IN TAPEMANAGER LIBRARY
186	PURGE ATTEMPT DSED AWAITING TAPESERVER LIBRARY
187	PURGE ATTEMPT DSED IN TAPESERVER LIBRARY

Example

For the system command *SN MT 49 UNISYS (1600)*, use the following input:

```
A[0] := 4;  
A[1] := 0 & 2 [38:06] & 13 [31:08] & 4 [15:08];  
A[2] := 49;  
A[3] := REPLACE POINTER (A[5]) BY 8"UNISYS";  
RSLT := SETSTATUS (5, 2, 2 & (VALUE (BPI1600)+1), A);
```


Security Enhancements

A new volume attribute, SECURITYLABELS, enables you to maintain the settings of security attributes in the tape label rather than in the volume directory on disk. When this information is on the tape itself, the tape is more easily transferred among hosts in a multihost environment.

The values of these attributes override the corresponding values in the volume directory on disk if the following conditions are true:

- The tape is mounted.
- The SECURITYLABELS option was set to TRUE when the entry for the volume in the tape volume directory was established. See “Establishing the Volume Directory Entry,” later in this section.

For information on how security information is maintained in the volume directory, see the *A Series Security Administration Guide*.

SECURITYLABELS Volume Attribute

SECURITYLABELS is a Boolean attribute that you can control with the WFL *VOLUME* statement, discussed later in this section. When SECURITYLABELS is TRUE, the values of the following volume security attributes are maintained in the tape labels:

- SECURITYTYPE
- SECURITYUSE
- SECURITYGUARD
- FAMILYOWNER

When these attribute values are maintained in the tape label, they override the values in the volume directory on disk when the system determines whether a given input or output operation is permitted on a given tape.

The following paragraphs describe the effects of setting the SECURITYLABELS attribute of a tape volume to TRUE.

Establishing the Volume Directory Entry

You can set the SECURITYLABELS attribute to TRUE in a VOLUME ADD statement that establishes an entry for the volume in the tape volume directory.

Writing Files to the Volume

When you create the tape volume – that is, when you first write a file to it – the system determines the values of the SECURITYTYPE, SECURITYUSE, SECURITYGUARD, and FAMILYOWNER file attributes of the first file to be written to the volume. You have the normal options for setting these values:

- In declaration statements in the program that requests the operation
- With file equation
- Dynamically at compilation, run time, or both

Any attribute not explicitly set receives the default value. See the *A Series File Attributes Programming Reference Manual* for details. The system then writes those values to the tape label and to the volume directory.

When writing a continuation volume of a multivolume file, the system requires a volume that has a SECURITYLABELS value that matches the SECURITYLABELS value on the first volume of the set. When it finds an appropriate volume, the system automatically copies the security attribute values from the first volume to each continuation volume.

Reading from the Volume

When you subsequently bring the tape online, the system reads the security attribute values from the tape label and updates the volume directory on disk accordingly.

ANSI87 Tape Format

Use of the SECURITYLABELS volume attribute requires that the tape be formatted in compliance with the ANSI X3.27 1987 (ANSI87) standard. See “ANSI X3.27-1987 (ANSI87) Tape Labels” in Appendix B for the ANSI87 formats.

You can create ANSI87 tapes with either of the following methods:

- Use the SYSOPS (System Options) system command to set the LABELFORMAT system option value to ANSI87DEFAULT as described in Section 2, “System Commands.”
- Follow the guidelines under “Using the ANSI87 Tape Label Format” in Section 1, “Overview,” to set the LABELFORMAT task attribute to define ANSI87 as the tape label format to be used in creating a new tape volume.

Note: *If the SECURITYLABELS volume attribute is set for a tape volume, the system uses the ANSI87 standard when writing to that tape, regardless of any user-supplied specifications or the system option setting.*

Tape Management Enhancements in WFL

The following information describes how the tape management enhancements affect the Work Flow Language (WFL). Unless otherwise stated, the section references in the following text refer to the *WFL Reference Manual*.

The enhancements change the syntax of the file attribute assignment, the ARCHIVE statements, and the VOLUME statement through the addition of the AUTOUNLOAD and SCRATCHPOOL file attributes, and the SECURITYLABELS volume attribute.

Using File Attributes

The SCRATCHPOOL file attribute is added to Table 5-4, "File Attribute Types" in Section 5, "Task Initiation."

Table 3-1 shows how the attribute types listed in the *File Attributes Reference Manual* correspond to the types used in WFL. Attribute types that cannot be accessed through WFL are listed as not available.

Table 3-1. Comparison of File Attribute Types to WFL Attribute Types

I/O Subsystem Type	Corresponding WFL Type	Valid File Attributes
Boolean	<Boolean file attribute>	All Boolean attributes
Event	Not available	None of the attributes
Integer	<integer file attribute>	All integer attributes
Mnemonic	<mnemonic file attribute>	All mnemonic attributes
Pointer	<file name file attribute>	FAMILYOWNER
		FILENAME
		PRINTCHARGE
		STATIONLIST
Pointer	<name file attribute>	APPLICATIONGROUP
		FAMILYNAME
		HOSTNAME
		INTNAME
		MYHOST
		MYHOSTGROUP
		SCRATCHPOOL
		YOURHOST
		YOURHOSTGROUP
		YOURUSERCODE
Pointer	<string file attribute>	AFTER
		DESTINATION
		FORMID
		LICENSEKEY
		NOTE
		RELEASEID
		TRANSFORM
		WARNINGS

continued

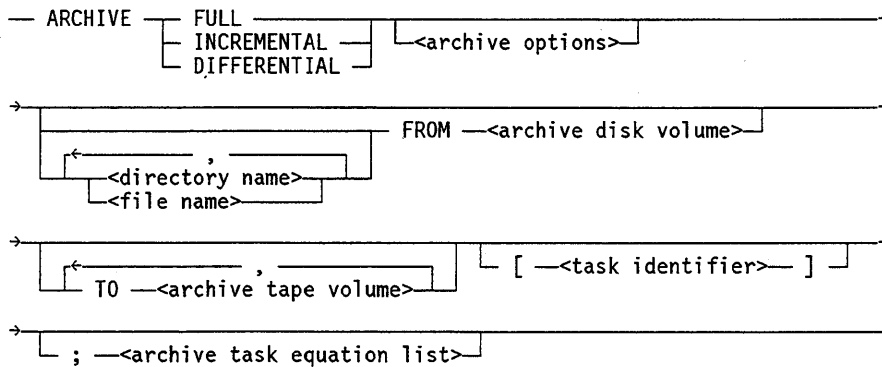
Table 3-1. Comparison of File Attribute Types to WFL Attribute Types (cont.)

I/O Subsystem Type	Corresponding WFL Type	Valid File Attributes
Pointer	<title file attribute>	ALIGNFILE
		COPYNAME
		MYNAME
		PRINTERCONTROL
		SECURITYGUARD
		STATIONNAME
		TITLE
		YOURNAME
		Real
Translatable	Not available	None of the attributes
Word	<real file attribute>	STATE
		TIMESTAMP
Word	<serial number list>	SERIALNO

ARCHIVE Statements

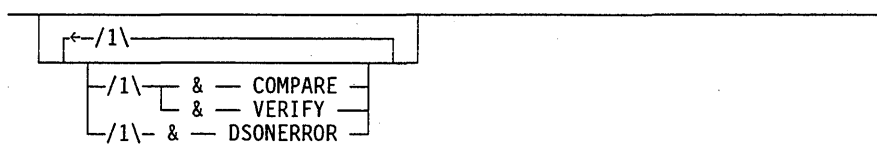
The SCRATCHPOOL and AUTOUNLOAD file attributes are added to the archive tape volume attribute list of the ARCHIVE statements in Section 6, "Statements."

<archive backup statement>

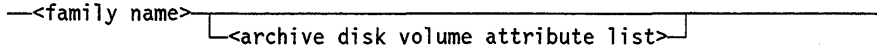


Programming Considerations (cont.)

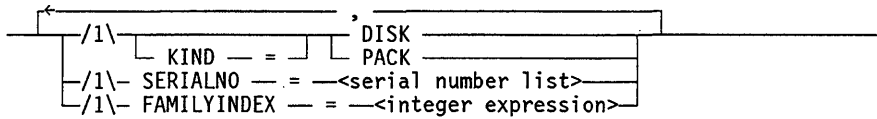
<archive options>



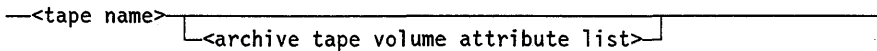
<archive disk volume>



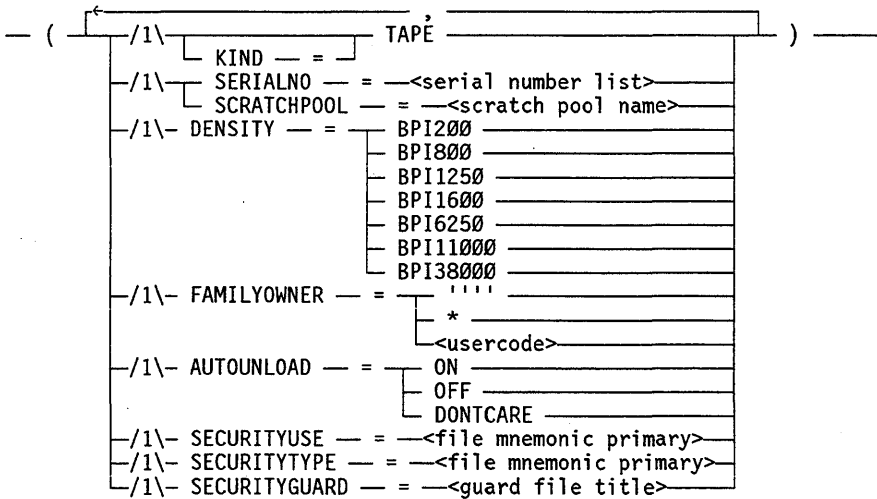
<archive disk volume attribute list>



<archive tape volume>



<archive tape volume attribute list>



<scratch pool name>



ARCHIVE Statement Options

The following table describes the syntax constructs added because of tape management enhancements:

Syntax Construct	Explanation						
<archive tape volume attribute list>	The following table describes the additional attributes of the archive tape volume attribute list:						
	<table border="0"> <thead> <tr> <th style="text-align: left;">Attribute Name</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td style="vertical-align: top;">SCRATCHPOOL</td> <td style="vertical-align: top;">Identifies the scratch pool from which the tape used for archiving is retrieved. The scratch pool name is a 17-character identifier without a default value.</td> </tr> <tr> <td style="vertical-align: top;">AUTOUNLOAD</td> <td style="vertical-align: top;">Determines whether or not a tape is unloaded when it is released by the system during a reel switch or a file close operation. If the value is ON, the tape is rewound and unloaded. If the value is OFF, the tape is not unloaded. If the value is DONTCARE, or if this attribute is not specified, tape behavior is controlled by the setting of the AUTOUNLOAD option of the MODE (Unit Mode) system command. See "MODE (Unit Mode)" in Section 2, "System Commands" of this <i>Tape Management Guide</i>.</td> </tr> </tbody> </table>	Attribute Name	Description	SCRATCHPOOL	Identifies the scratch pool from which the tape used for archiving is retrieved. The scratch pool name is a 17-character identifier without a default value.	AUTOUNLOAD	Determines whether or not a tape is unloaded when it is released by the system during a reel switch or a file close operation. If the value is ON, the tape is rewound and unloaded. If the value is OFF, the tape is not unloaded. If the value is DONTCARE, or if this attribute is not specified, tape behavior is controlled by the setting of the AUTOUNLOAD option of the MODE (Unit Mode) system command. See "MODE (Unit Mode)" in Section 2, "System Commands" of this <i>Tape Management Guide</i> .
Attribute Name	Description						
SCRATCHPOOL	Identifies the scratch pool from which the tape used for archiving is retrieved. The scratch pool name is a 17-character identifier without a default value.						
AUTOUNLOAD	Determines whether or not a tape is unloaded when it is released by the system during a reel switch or a file close operation. If the value is ON, the tape is rewound and unloaded. If the value is OFF, the tape is not unloaded. If the value is DONTCARE, or if this attribute is not specified, tape behavior is controlled by the setting of the AUTOUNLOAD option of the MODE (Unit Mode) system command. See "MODE (Unit Mode)" in Section 2, "System Commands" of this <i>Tape Management Guide</i> .						

Example

The following example shows how to back up all files under the usercode MYCODE to a tape volume called NEWTAPE from the scratch pool POOL1:

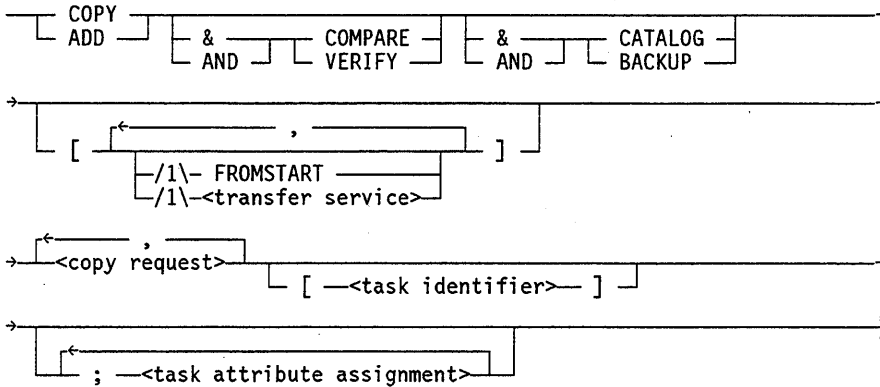
```
ARCHIVE DIFFERENTIAL (MYCODE) = FROM TESTFAMILY
      TO NEWTAPE (KIND = TAPE, SCRATCHPOOL = POOL1);
```

COPY or ADD Statement

The SCRATCHPOOL and AUTOUNLOAD file attributes are added to the syntax diagram and text for the source volume attribute list and destination volume attribute list of the COPY or ADD statement in Section 6, "Statements." These file attributes are also valid for the CANDE COPY and ADD commands.

Programming Considerations (cont.)

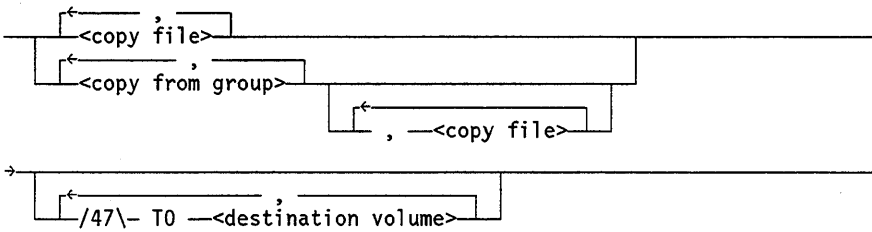
<copy or add statement>



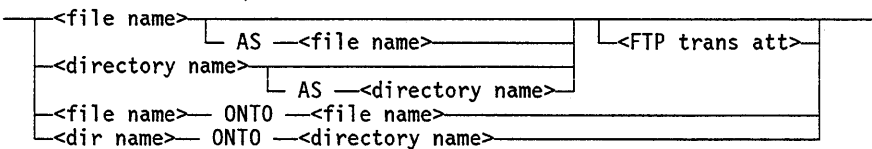
<transfer service>



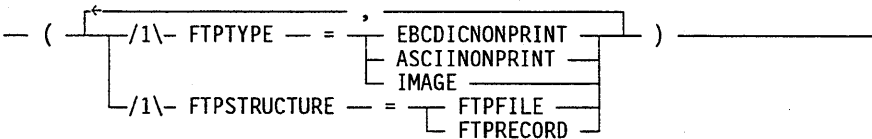
<copy request>



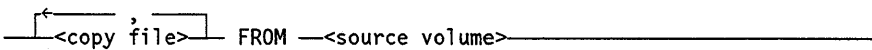
<copy file>



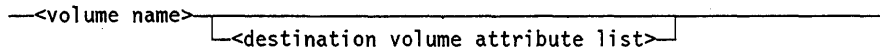
<FTP transform att list>



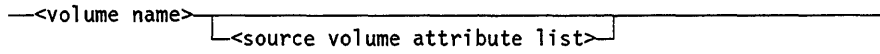
<copy from group>



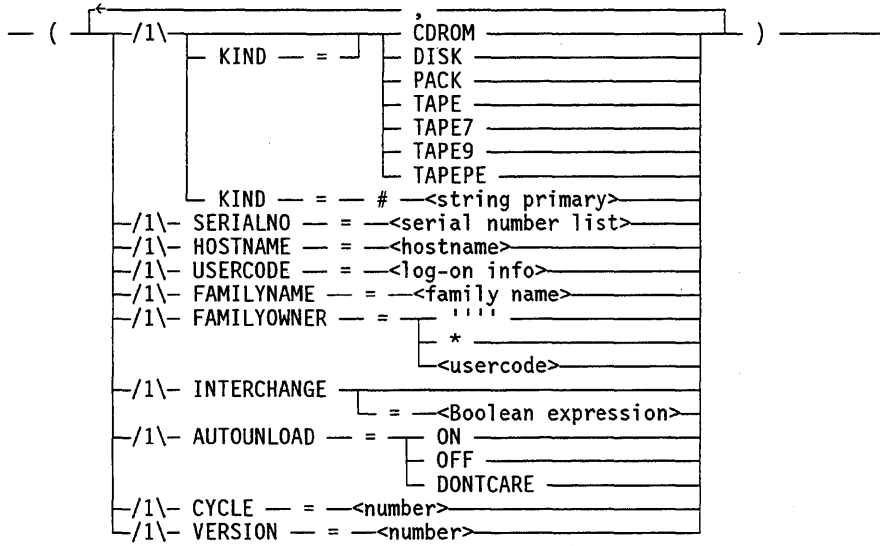
<destination volume>

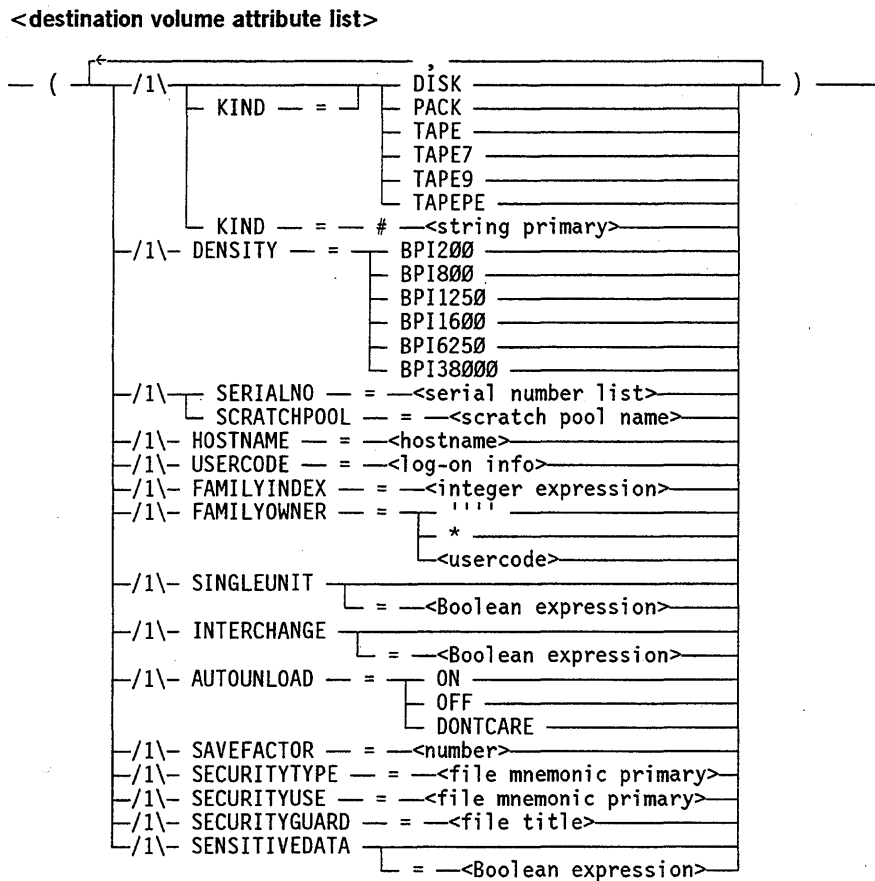


<source volume>



<source volume attribute list>





<scratch pool name>

— <name> —

Specifying File and Tape Volume Attributes

The DENSITY option can be used only with the <destination volume attribute list> and not the <source volume attribute list>. For more information on this option, see the *File Attributes Reference Manual*.

The following table describes the syntax constructs added because of tape management enhancements:

File Attribute	Meaning
AUTOUNLOAD	Determines whether or not a tape is unloaded when it is released by the system during a reel switch or a file close operation. If the value is ON, the tape is rewound and unloaded. If the value is OFF, the tape is not unloaded. If the value is DONTCARE, or if this attribute is not specified, tape behavior is controlled by the setting of the AUTOUNLOAD option of the MODE (Unit Mode) system command. See "MODE (Unit Mode)" in Section 2 of this <i>Tape Management Guide</i> . If a reel is switched during a COPY and COMPARE operation, intermediate reels are not unloaded, regardless of the AUTOUNLOAD value, until the COMPARE phase is finished.
SCRATCHPOOL	Indicates the scratch pool from which the tape is retrieved when files are copied. The scratch pool name is a 17-character identifier without a default value.

Examples

The following examples illustrate various aspects of the COPY statement with the AUTOUNLOAD and SCRATCHPOOL file attributes:

- This statement copies the file X as A/B from disk PACK to tape T. Because the value of the AUTOUNLOAD file attribute is ON, the tape is unloaded when the operation is completed.

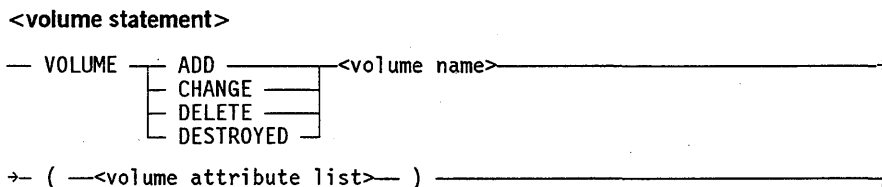
```
COPY X AS A/B FROM PACK TO T(KIND=TAPE, AUTOUNLOAD=ON);
```

- This statement copies the file X as A/B from disk PACK to tape T in the TEST scratch pool:

```
COPY X AS A/B FROM PACK TO T(KIND=TAPE, SCRATCHPOOL=TEST);
```

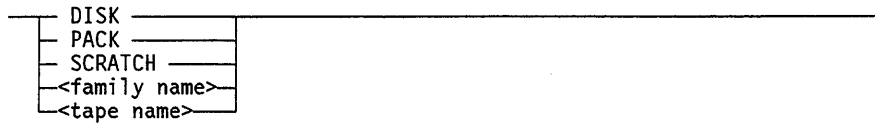
VOLUME Statement

The SECURITYLABELS volume attribute is added to the syntax diagram and text for the volume attribute list of the VOLUME statement in Section 6, "Statements."

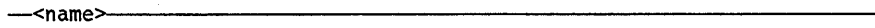


Programming Considerations (cont.)

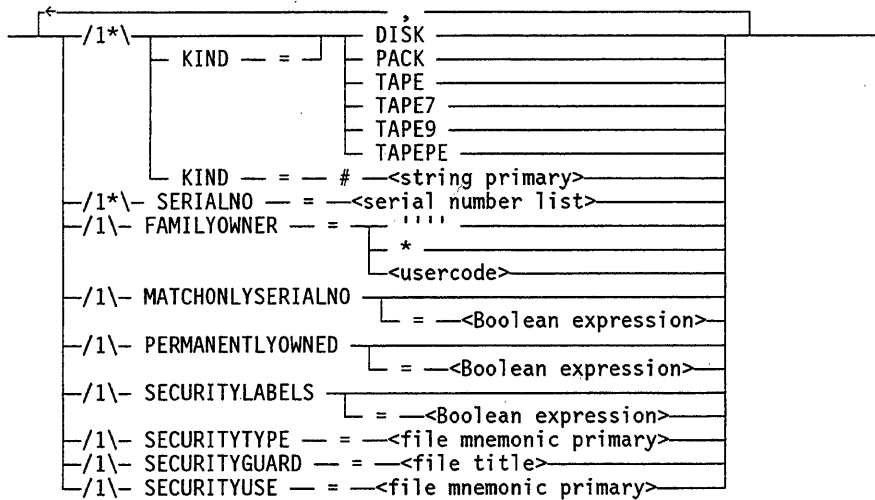
<volume name>



<tape name>



<volume attribute list>



Note: A *VOLUME* statement can set the *SECURITYLABELS* attribute to *TRUE* or the *PERMANENTLYOWNED* attribute to *TRUE* but not both. The system rejects a *VOLUME* statement that specifies both *SECURITYLABELS=TRUE* and *PERMANENTLYOWNED=TRUE*.

Explanation

The following table explains the syntax construct added because of tape management enhancements:

File Attribute	Meaning
SECURITYLABELS	<p>When this attribute is set to TRUE, the security attributes SECURITYTYPE, SECURITYUSE, SECURITYGUARD, and FAMILYOWNER are maintained in the tape volume label. With this information on the tape itself, a tape can be transferred easily among hosts in a multihost environment.</p> <p>At volume creation, the system determines the security attribute values of the first file to be written to the volume. The system then writes those values to the tape label and to the volume directory. Each volume of a multivolume file must have a SECURITYLABELS value that matches the SECURITYLABELS value of the first volume.</p> <p>When a tape with SECURITYLABELS set to TRUE comes online, the system reads the security attribute values from the tape label and updates the volume directory accordingly.</p>

Example

This example establishes a volume directory entry for a tape volume named QFILES under the usercode AEDEPT. The security attribute values are stored in the tape volume label when the volume is created.

```
VOLUME ADD QFILES (TAPE, SERIALNO = (111111), FAMILYOWNER = AEDEPT,
                  SECURITYLABELS = TRUE, SECURITYTYPE = GUARDED,
                  SECURITYGUARD = NEWGUARD);
```

Logging Enhancements

The following pages describe changes to LOGANALYZER and to the system log (SUMLOG).

The LOGANALYZER selection option called VOLUMES now results in a revised output report format.

Two LOGANALYZER configuration and maintenance options have been added for A Series users with a cartridge tape library (CTL) system that includes one or more cartridge library units (CLUs) and a RoboHost A TAPESERVER library. For more information, refer to the *A Series RoboHost A Operation, Installation, and Administration Guide*.

The following system log (SUMLOG) entry formats have been changed:

- Major Type 1–Minor Type 5: File Open Entry (LOGOPENTYPE)
- Major Type 1–Minor Type 6: File Close Entry (LOGCLOSETYPE)
- Major Type 6–Minor Type 3: SETSTATUS Call (LOGSETSTAT)
- Major Type 15–Minor Types 1, 2, 3, 4, 5, 12, 13: Volume Status Entry (LOGMAJVOL)

A complete description of the entry formats is given later in this section.

The following additional system log changes for A Series users with RoboHost A cartridge tape library systems are listed in this section and described in the *A Series RoboHost A Operation, Installation, and Administration Guide*:

- Major Type 2–Minor Type 24: CLU Hardware Report (CLUHWREPORT)
- Major Type 26: Tape Library Entry (LOGMAJ_MEDIALIB)

LOGANALYZER Output Illustration for VOLUMES Option

The VOLUMES option defines the types of log entries that are being output. While the VOLUMES option is not new, the format of the output report has been revised. Figure 3–1 shows a sample report.

LOGANALYZER VERSION: 39.106.0113 SDASUPPORT VERSION: 39.106.0066, JOBFORMATTER VERSION: 39.106.0413.
 MCP *SYSTEM/ASD/MCP/39106/DIAG VERSION: 39.106 (MCP/AS)
 ANALYZED BY A6 (SYSTEM SERIAL = 2345) ON 10/12/1992 10:24:33

REQUEST: 1015 VOLUMES (UNSORTED BY DEFAULT)

SUMLOG #002772 CREATED BY A6 (SYSTEM SERIAL = 2345) ON 10/10/1992 00:01:06
 TITLE = *SYSTEM/SUMLOG ON PACK (CURRENT SUMLOG).

FILE CONTAINS 47782 RECORDS FROM 10/10/1992 00:01:00 TO 10/12/1992 10:25:00

Monday, October 12, 1992

10:18:04	VOL 1579	VOLUME HOLD STATUS CHANGE UNIT 193 [000198] PAY1 (HOLD) WRITE ENABLED FILE ID FILE000 CREATION DATE 92286 CREATION SITE 2345 CYCLE 1 VERSION 0 SAVEFACTOR 30 REEL 1
10:18:17	VOL 1580	VOLUME PURGED UNIT 193 [000198] PAY1 (HOLD) WRITE ENABLED FILE ID FILE000 CREATION DATE 92286 CREATION SITE 2345 CYCLE 1 VERSION 0 SAVEFACTOR 30 REEL 1
10:18:48	VOL 1582	VOLUME PURGED UNIT 193 [009999] S C R A T C H POOL: PAYROLL (HOLD) WRITE ENABLED OLD SERIAL NUMBER WAS [000198]
10:21:25	VOL 1594	VOLUME HOLD STATUS CHANGE UNIT 193 [009999] S C R A T C H POOL: PAYROLL WRITE ENABLED
10:21:29	VOL 1592	NEW FILE CREATED ON VOLUME UNIT 193 [009999] ARS. WRITE ENABLED FILE ID FILE000 CREATION DATE 92286 CREATION SITE 2345 CYCLE 1 VERSION 0 SAVEFACTOR 30 REEL 1
10:21:36	VOL 1595	VOLUME ONLINE UNIT 193 [009999] ARS WRITE ENABLED FILE ID FILE000 CREATION DATE 92286 CREATION SITE 2345 CYCLE 1 VERSION 0 SAVEFACTOR 30 REEL 1
10:22:04	VOL 1598	VOLUME USED UNIT 193 [009999] ARS WRITE ENABLED FILE ID FILE000 CREATION DATE 92286 CREATION SITE 2345 CYCLE 1 VERSION 0 SAVEFACTOR 30 REEL 1
10:22:10	VOL 1599	VOLUME ONLINE UNIT 193 [009999] ARS WRITE ENABLED FILE ID FILE000 CREATION DATE 92286 CREATION SITE 2345 CYCLE 1 VERSION 0 SAVEFACTOR 30 REEL 1

NORMAL TERMINATION FOR LOGANALYZER PROGRAM.

Figure 3-1. VOLUMES Option Output

Major Type 1–Minor Type 5: File Open Entry (LOGOPENTYPE)

The File Open SUMLOG entry has been enhanced in the following ways:

- If the TAPEMANAGER library changed any file attributes of a tape file before the file was opened, then Word 7, Field [44: 1] is set to 1.
- The high-order bits of the open error message number are now stored in Word 7, Field [43: 4]. The low-order bits of the open error message number continue to be stored in Field [15: 8].

Table 3–2 describes the complete format of the File Open entry, including the new field.

Table 3–2. Major Type 1–Minor Type 5

Minor Type = 5 File Open Entry (LOGOPENTYPE)	
Words	Description
0–3	As described in the <i>A Series System Software Support Reference Manual</i> in Table 12–2, “First Four Log Entry Words”
4	File size in units of words (valid for disk and pack only)
5	LINK to internal name
6	LINK to external name
7	Open description word [47: 2] File Access Rule used 1 = Actor rights 2 = Declarer rights 3 = Union of Actor and Declarer rights [45: 1] Value of DUMMYFILE attribute [44: 1] Attributes of the tape file were modified by the TAPEMANAGER library.

continued

Table 3-2. Major Type 1-Minor Type 5 (cont.)

Minor Type = 5 File Open Entry (LOGOPENTYPE)	
Words	Description
	<p>[43: 4] The high-order four bits of the open error message number. The low-order bits are stored in field [15: 8].</p> <p>The open error message number identifies the system message that explains the open error. When an open error occurs, the system issues a Major Type 14-Minor Type 4 (INFO Message) log entry that contains the text of this system message. When such a log entry is printed by LOGANALYZER or appears in a job summary, the message text is preceded by the string <i>MSRFIB</i> and the open error message number. For example, open error message number 74 refers to system message MSRFIB74, "NOT CLOSED".</p> <p>For processes discontinued by open errors, the open error message number also corresponds to the HISTORYREASON task attribute value. If the HISTORYTYPE value is 4 (DSEDV) and the HISTORYCAUSE value is 8 (SOFTIOERRCAUSEV), then the open error message number equals the HISTORYREASON value. If the HISTORYTYPE value is 4 (DSEDV) and the HISTORYCAUSE value is 14 (SOFTIOERR2CAUSEV), then the open error message is equal to the HISTORYREASON value plus 256. For descriptions of the HISTORYTYPE, HISTORYCAUSE, and HISTORYREASON task attributes, refer to the <i>A Series Task Attributes Programming Reference Manual</i>.</p> <p>[31: 8] Logical type (See Table 12-21, "Peripheral Logical Types," in the <i>A Series System Software Support Reference Manual</i>.)</p> <p>[23: 8] Value of the FILEKIND file attribute</p> <p>[15: 8] The low-order eight bits of the open error message number. For further information, refer to the description of Field [43: 4].</p> <p>[7: 7] Value of the MYUSE file attribute:</p> <ul style="list-style-type: none"> 1 = INPUT 2 = OUTPUT 3 = I/O <p>[0: 1] If this bit is 1, then [15: 8] contains an error code.</p>
8	LINK to access code
9	<p>OPEN parameters</p> <p>[46: 1] 1 if file is a valid subfile</p> <p>[41:10] Subfile index, if file is a valid subfile</p>

continued

Table 3-2. Major Type 1-Minor Type 5 (cont.)

Minor Type = 5 File Open Entry (LOGOPENTYPE)	
Words	Description
	[15: 4] Motion
	1 = Positioning requested
	2 = No positioning requested
	[11: 4] Position
	1 = AT FRONT
	2 = AT END
	[7: 8] Open type
	For nonport files:
	1 = OPEN WAIT
	2 = AVAILABLE
	3 = OFFER
	4 = RESIDENT
	5 = PRESENT
	6 = PBT REEL SWITCH OPEN
	7 = STACK ASSIGNED OPEN REVERSE
	For port files:
	1 = WAIT
	2 = AVAILABLE
	3 = RETURN (OFFER)
	4 = DONTWAIT
10	Device number
11	Stack using the file (actor)
	[47:12] Stack number
	[31:16] Job number
	[15:16] Task number
12	Stack owning the file (declarer; same word format as Word 11)

continued

Table 3-2. Major Type 1-Minor Type 5 (cont.)

Minor Type = 5 File Open Entry (LOGOPENTYPE)																											
Words	Description																										
13	<p>LINK to port file information. This LINK is 0 if this entry is not for a port file. If the LINK is nonzero, it points to an area used to contain a block of additional information for port files. The information contained in that block is listed in the following subtable. The location of each item is described by an offset from the start of this block of information.</p> <p>The LINKs within this block of information point to strings of characters. The length of each string is given in characters (not words) in the length field [39:20] of the LINK. The index field [19:20] of the LINK provides the word index in the log entry where the information starts.</p> <table border="1"> <thead> <tr> <th>Offset</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>LINK to YOURHOSTNAME value</td> </tr> <tr> <td>1</td> <td>LINK to YOURNAME value</td> </tr> <tr> <td>2</td> <td>Value of MAXSEGSIZE attribute</td> </tr> <tr> <td>3</td> <td>LINK to YOURUSERCODE value</td> </tr> <tr> <td>4</td> <td>Value of MYPORADDRESS attribute</td> </tr> <tr> <td>5</td> <td>Value of MYSUBPORTADDRESS attribute</td> </tr> <tr> <td>6</td> <td>LINK to SERVICE value</td> </tr> <tr> <td>7</td> <td>LINK to PROVIDERGROUP value</td> </tr> <tr> <td>8</td> <td>LINK to the name of the provider selected, if the OPEN succeeded; otherwise, 0 (zero)</td> </tr> <tr> <td>9</td> <td>CONNECTTIMELIMIT value, if one was specified; otherwise, 0 (zero)</td> </tr> <tr> <td>10</td> <td>Initiation primitive invoked 0 = OPEN 1 = AWAITOPEN</td> </tr> <tr> <td>11</td> <td>RESPOND invocation 1 = RESPOND(ACCEPTCLOSE) 2 = RESPOND(REJECTCLOSE) 31 = RESPOND was not invoked.</td> </tr> </tbody> </table>	Offset	Contents	0	LINK to YOURHOSTNAME value	1	LINK to YOURNAME value	2	Value of MAXSEGSIZE attribute	3	LINK to YOURUSERCODE value	4	Value of MYPORADDRESS attribute	5	Value of MYSUBPORTADDRESS attribute	6	LINK to SERVICE value	7	LINK to PROVIDERGROUP value	8	LINK to the name of the provider selected, if the OPEN succeeded; otherwise, 0 (zero)	9	CONNECTTIMELIMIT value, if one was specified; otherwise, 0 (zero)	10	Initiation primitive invoked 0 = OPEN 1 = AWAITOPEN	11	RESPOND invocation 1 = RESPOND(ACCEPTCLOSE) 2 = RESPOND(REJECTCLOSE) 31 = RESPOND was not invoked.
Offset	Contents																										
0	LINK to YOURHOSTNAME value																										
1	LINK to YOURNAME value																										
2	Value of MAXSEGSIZE attribute																										
3	LINK to YOURUSERCODE value																										
4	Value of MYPORADDRESS attribute																										
5	Value of MYSUBPORTADDRESS attribute																										
6	LINK to SERVICE value																										
7	LINK to PROVIDERGROUP value																										
8	LINK to the name of the provider selected, if the OPEN succeeded; otherwise, 0 (zero)																										
9	CONNECTTIMELIMIT value, if one was specified; otherwise, 0 (zero)																										
10	Initiation primitive invoked 0 = OPEN 1 = AWAITOPEN																										
11	RESPOND invocation 1 = RESPOND(ACCEPTCLOSE) 2 = RESPOND(REJECTCLOSE) 31 = RESPOND was not invoked.																										

continued

Table 3-2. Major Type 1-Minor Type 5 (cont.)

Minor Type = 5 File Open Entry (LOGOPENTYPE)	
Words	Description
18	ASSOCIATED DATA transmission 0 = ASSOCIATED DATA was sent. 1 = ASSOCIATED DATA was not sent.
19	ASSOCIATED DATA receipt 0 = ASSOCIATED DATA was received. 1 = ASSOCIATED DATA was not received.
14	LINK to family name in substandard form
15-end	Variable-length data pointed to by the LINKs in this entry

For information on peripheral types, see Tables 12-21 through 12-24 in the *A Series System Software Support Reference Manual*.

Major Type 1-Minor Type 6: File Close Entry (LOGCLOSETYPE)

The File Close SUMLOG entry has been enhanced so that the high-order bits of the close error message number are now stored in Word 7, Field [43: 4]. The low-order bits of the close error message number continue to be stored in Field [15: 8].

Table 3-3 describes the complete format of the File Close entry, including the new field.

Table 3-3. Major Type 1-Minor Type 6

Minor Type = 6 File Close Entry (LOGCLOSETYPE)	
Words	Description
0-3	As described in the <i>A Series System Software Support Reference Manual</i> in Table 12-2, "First Four Log Entry Words"
4	File size in units of words (valid for disk and pack only)

continued

Table 3-3. Major Type 1–Minor Type 6 (cont.)

Minor Type = 6 File Close Entry (LOGCLOSETYPE)	
Words	Description
5	LINK to internal name (standard form)
6	LINK to external name (standard form)
7	<p>Close description word</p> <p>[47: 2] File Access Rule used</p> <ul style="list-style-type: none"> 1 = Actor rights 2 = Declarer rights 3 = Union of actor and declarer rights <p>[45: 1] Value of DUMMYFILE file attribute</p> <p>[43: 4] The high-order four bits of the close error message number. The low-order bits are stored in field [15: 8].</p> <p>The close error message number identifies the system message that explains the close error. When a close error occurs, the system issues a Major Type 14–Minor Type 4 (INFO Message) log entry that contains the text of this system message. When such a log entry is printed by LOGANALYZER or appears in a job summary, the message text is preceded by the string <i>MSRFIB</i> and the close error message number. For example, close error message number 19 refers to system message MSRFIB16, "ILLEGAL OPTIONAL FILE".</p> <p>For processes discontinued by close errors, the close error message number also corresponds to the HISTORYREASON task attribute value. If the HISTORYTYPE value is 4 (DSEDV) and the HISTORYCAUSE value is 8 (SOFTIOERRCAUSEV), then the close error message number equals the HISTORYREASON value. If the HISTORYTYPE value is 4 (DSEDV) and the HISTORYCAUSE value is 14 (SOFTIOERR2CAUSEV), then the close error message is equal to the HISTORYREASON value plus 256. For descriptions of the HISTORYTYPE, HISTORYCAUSE, and HISTORYREASON task attributes, refer to the <i>A Series Task Attributes Programming Reference Manual</i>.</p> <p>[31: 8] Logical type (See Table 12-21, "Peripheral Logical Types," in the <i>A Series System Software Support Reference Manual</i>.)</p> <p>[23: 8] Value of FILEKIND file attribute</p> <p>[15: 8] The low-order eight bits of the close error message number. For further information, refer to the description of Field [43: 4].</p>

continued

Table 3-3. Major Type 1-Minor Type 6 (cont.)

Minor Type = 6 File Close Entry (LOGCLOSETYPE)	
Words	Description
	[7: 7] Value of MYUSE file attribute 1 = INPUT 2 = OUTPUT 3 = I/O
	[0: 1] If this bit is a 1, then field [15: 8] contains an error code.
8	Name qualification information [42:15] USASI generation number [27:11] SAVEFACTOR [16:17] Creation date in binary Julian format
9	File blocking information [47:16] BLOCKSIZE [27:11] MINRECSIZE [15:16] MAXRECSIZE
10	Transaction count (number of logical I/O operations)
11	Elapsed I/O time
12	Miscellaneous information For a disk file [47: 1] Installation-allocated disk (IAD) file [46: 1] Permanent file [44: 1] Updated file [43: 1] CLOSE with CRUNCH [42: 1] Protected file [40: 1] Interchange disk pack file [32: 1] Duplicated file [31: 4] Number of duplicate copies

continued

Table 3-3. Major Type 1-Minor Type 6 (cont.)

Minor Type = 6 File Close Entry (LOGCLOSETYPE)	
Words	Description
	<p>[23:24] Value of AREASIZE attribute</p> <p>For a tape file</p> <p>[15: 1] Contains 1 if the file was a multivolume file</p> <p>[14:15] Reel number</p> <p>For units other than tape or disk, this word contains zeros.</p>
13	Serial number (six EBCDIC characters) for tape file. This word is 0 for other units.
14	<p>Unit information</p> <p>[47:16] INTMODE</p> <p>[25:14] CYCLE</p> <p>[11: 8] VERSION</p> <p>[0: 1] UNITS</p>
15	<p>Close parameters</p> <p>[46: 1] Valid subfile</p> <p>[39:16] Subfile index, if file is a valid subfile</p> <p>[23: 8] Disposition</p> <p>1 = REWIND</p> <p>2 = NOREWIND</p> <p>3 = SAVE</p> <p>4 = LOCK</p> <p>5 = PURGE</p> <p>6 = CRUNCH</p> <p>7 = HERE</p> <p>8 = BLOCKEXIT</p> <p>[15: 8] Association</p> <p>1 = RELEASE</p> <p>2 = RETAIN</p>

continued

Table 3-3. Major Type 1-Minor Type 6 (cont.)

Minor Type = 6 File Close Entry (LOGCLOSETYPE)											
Words	Description										
	<p>3 = RESERVE</p> <p>4 = DISABLE</p> <p>[7: 8] Close type</p> <p>For nonport files:</p> <p>1 = REGULAR CLOSE</p> <p>2 = REEL CLOSE</p> <p>3 = DONT WAIT</p> <p>For port files (note that disposition and association fields are not used):</p> <p>0 = BLOCKEXIT</p> <p>1 = NORMAL</p> <p>2 = DONTWAIT</p>										
16	<p>Physical I/O count</p> <p>[47:24] Read count</p> <p>[23:24] Write count</p>										
17	LINK to FORMMESSAGE (standard form)										
18	<p>LINK to port file information. This LINK is 0 if this entry is not for a port file. If the LINK is nonzero, it points to an area used to contain a block of additional information for port files. The information contained in that block is listed in the following subtable. The location of each item is described by an offset from the start of this block of information.</p> <p>The LINKS within this block of information point to strings of characters. The length of each string is given in characters (not words) in the length field [39:20] of the LINK. The index field [19:20] of the LINK provides the word index in the log entry where the information starts.</p> <table border="1"> <thead> <tr> <th>Offset</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>LINK to YOURHOSTNAME value</td> </tr> <tr> <td>1</td> <td>LINK to YOURNAME value</td> </tr> <tr> <td>2</td> <td>Value of MAXSEGSIZE attribute</td> </tr> <tr> <td>3</td> <td>LINK to YOURUSERCODE value</td> </tr> </tbody> </table>	Offset	Contents	0	LINK to YOURHOSTNAME value	1	LINK to YOURNAME value	2	Value of MAXSEGSIZE attribute	3	LINK to YOURUSERCODE value
Offset	Contents										
0	LINK to YOURHOSTNAME value										
1	LINK to YOURNAME value										
2	Value of MAXSEGSIZE attribute										
3	LINK to YOURUSERCODE value										

continued

Table 3-3. Major Type 1-Minor Type 6 (cont.)

Minor Type = 6 File Close Entry (LOGCLOSETYPE)	
Words	Description
4	Number of messages sent
5	Number of messages received
6	Number of segments sent
7	Number of segments received
8	Number of retransmissions
9	Number of control frames sent
10	Value of MYPORTADDRESS attribute
11	Value of MYSUBPORTADDRESS attribute
12	LINK to SERVICE value
13	LINK to PROVIDERGROUP value
14	LINK to provider selected, if the CLOSE succeeded; otherwise, 0 (zero)
15	CLOSE type 0 = ABORT 1 = REQUEST
16	INITIATOR 0 = Locally initiated 1 = Correspondent initiated 2 = Provider initiated
17	RESPOND invocation 1 = RESPOND(ACCEPTCLOSE) 2 = RESPOND(REJECTCLOSE) 31 = RESPOND was not invoked.
18	ASSOCIATED DATA transmission 0 = ASSOCIATED DATA was sent. 1 = ASSOCIATED DATA was not sent.

continued

Table 3-3. Major Type 1-Minor Type 6 (cont.)

Minor Type = 6 File Close Entry (LOGCLOSETYPE)	
Words	Description
	19 ASSOCIATED DATA receipt 0 = ASSOCIATED DATA was received. 1 = ASSOCIATED DATA was not received.
19	External device number
20	Stack using the file (actor) [47:12] Stack number [31:16] Job number [15:16] Task number
21	Stack owning the file (declarer; same field format as Word 20)
22	LINK to family name in substandard form
23	Number of physical read operations initiated for the file
24	Number of physical write operations initiated for the file
25	FILESTRUCTURE for disk files. This word is -1 for other kinds of files. 0 = ALIGNED180 1 = STREAM 5 = BLOCKED
26	Buffer size in words
27	Buffer size source 1 The buffer size for the file was based on the BLOCKSIZE file attribute because the BUFFERSIZE attribute is not applicable to this file. 2 The buffer size for the file was specified through an assignment to the BUFFERSIZE file attribute. 3 The buffer size for the file was based on the BUFFERGOAL memory management parameter of the SF (Set Factors) system command.

continued

Table 3-3. Major Type 1–Minor Type 6 (cont.)

Minor Type = 6 File Close Entry (LOGCLOSETYPE)	
Words	Description
28	Density for tape files. This word is –1 for other units. 0 = BPI800 1 = BPI556 2 = BPI200 3 = BPI1600 4 = BPI6250 5 = BPI38000 6 = BPI1250 7 = BPI11000
29	Physical Data Representation [47:16] EXTMODE [31:16] CCSVERSION [15:01] CCSVERSION validity bit (1 if CCSVERSION contains a valid value)
30–end	Variable-length data pointed to by the LINKs in this entry

For information on peripheral types, see Tables 12-21 through 12-24 in the *A Series System Software Support Reference Manual*.

Major Type 2–Minor Type 24: CLU Hardware Report (CLUHWREPORT)

A new log entry minor type has been added to record cartridge library unit (CLU) hardware activity. This entry is relevant only to RoboHost A users. Refer to the *A Series RoboHost A Operations, Installation, and Administration Guide* for a description of the entry format.

Major Type 6–Minor Type 3: SETSTATUS Call (LOGSETSTAT)

The SETSTATUS call log entry has been enhanced so that word $n + 9$ stores the value of the V parameter of SETSTATUS.

Table 3–4 describes the complete format of this log entry, including the new field.

Table 3–4. Major Type 6–Minor Type 3

Minor Type = 3 SETSTATUS Call Entry (LOGSETSTAT)	
Words	Description
0–3	As described in the <i>A Series System Software Support Reference Manual</i> in Table 12–2, “First Four Log Entry Words”
4	Information regarding parameters to SETSTATUS call [47: 8] TYPE, the first parameter [39: 8] SUBTYPE, the second parameter [31: 8] V, the third parameter [15:16] Number of words to follow in the entry
5 for n words	Contents of the array parameter to SETSTATUS, where n is the number of words specified in field [15:16] of Word 4
If the length of the entire log record is greater than $n + 5$ words, the remaining words have the following meaning:	
$n + 5$	Identity information code [47:06] Length of fixed part of entry in words [41:10] Entire length of identity information
$n + 6$	Miscellaneous identity information [07:08] Source of SETSTATUS command 1 = From ODT 2 = From remote ODT 3 = From BNA connected host 4 = From DCKEYIN 5 = From MCS
$n + 7$	LINK to SOURCENAME task attribute value

continued

Table 3-4. Major Type 6-Minor Type 3 (cont.)

Minor Type = 3 SETSTATUS Call Entry (LOGSETSTAT)	
Words	Description
$n + 8$	LINK to USERCODE task attribute value. This word is used only if the source of the SETSTATUS command was a BNA connected host (that is, field [07:08] of Word $n + 6$ stores a 3).
$n + 9$	The V (third parameter) value of SETSTATUS

Note that the field [31: 8] of Word 4 differs from the third parameter value (V) in the following cases:

Case	Value in Word 4 [31: 8]
AD or DD	V & (V.[10: 1]) [1: 1] & (V.[39: 1]) [2: 1]
PP or XP	IF V = -1 THEN 6 ELSE V
SW-	IF V = -1 THEN 6 ELSE V

Note: The SETSTATUS call entry records a call to SETSTATUS, which is the MCP procedure called to carry out operator input commands. Thus, these records show most operator input that changes the status of the system, such as time and date changes, tapes purged, and tasks discontinued.

Major Type 15–Minor Types 1–5, 12, 13: Volume Status Entry (LOGMAJVOL)

The Volume Status SUMLOG entries have been enhanced in the following areas:

- The Minor Type 12 (Tape Volume Used) entry has been added. This log entry is issued when a process first opens a tape file on a particular tape volume. This log entry is not issued if the same process opens subsequent tape files on the same tape volume.
- The Minor Type 13 (Tape Volume Hold) entry has been added. This log entry is issued when a HOLD (Hold Unit for Operator Use) system command changes the HOLD status of a tape unit.
- The Minor Type 5 (Tape Volume NEWFILE) is now issued for all cases in which a tape file is assigned to a new tape file on a tape volume. Formerly, this log entry was not issued if the new tape file was unlabeled.
- Word 7, Field [42: 1] is now set if the volume is unavailable for file assignment because of a HOLD command in effect for the unit.
- For Minor Type 3 (Tape Volume Purged) log entries, Words 22 through 24 store the scratch pool name in EBCDIC characters. The first byte is a binary count of the number of characters in the scratch pool name.

Table 3–5 describes the complete format of the Volume Status log entries for Minor Types 1 through 5, 12, and 13.

Table 3–5. Major Type 15–Minor Types 1 through 5, 12, and 13

Minor Type = 1 Volume Online (LOGVOLON) Minor Type = 2 Volume Offline (LOGVOLOFF) Minor Type = 3 Tape Volume Purged (LOGVOLPG) Minor Type = 4 Tape Volume Expired (LOGVOLEXP) Minor Type = 5 Tape Volume NEWFILE (LOGVOLNEW) Minor Type = 12 Tape Volume Used File (LOGVOLUSED) Minor Type = 13 Tape Volume Hold (LOGVOLHOLD)	
Words	Description
0–3	As described in the <i>A Series System Software Support Reference Manual</i> in Table 12–2, “First Four Log Entry Words”
4	External device number
5	Peripheral Description [47: 8] Peripheral Type. (See Table 12–21, “Peripheral Logical Types,” in the <i>A Series System Software Support Reference Manual</i> .)

continued

Table 3-5. Major Type 15—Minor Types 1 through 5, 12, and 13 (cont.)

Words	Description
<p>Minor Type = 1 Volume Online (LOGVOLON) Minor Type = 2 Volume Offline (LOGVOLOFF) Minor Type = 3 Tape Volume Purged (LOGVOLPG) Minor Type = 4 Tape Volume Expired (LOGVOLEXP) Minor Type = 5 Tape Volume NEWFILE (LOGVOLNEW) Minor Type = 12 Tape Volume Used File (LOGVOLUSED) Minor Type = 13 Tape Volume Hold (LOGVOLHOLD)</p>	
<p>[39: 8] Peripheral subtype. (See Table 12-22, "Peripheral Subtypes," in the <i>A Series System Software Support Reference Manual</i>.) [31: 9] Density. (See Table 12-23, "Peripheral Densities," in the <i>A Series System Software Support Reference Manual</i>.)</p>	
6	Serial Number in EBCDIC characters
7	<p>Volume Status</p> <p>[47: 1] SET if volume is a scratch volume</p> <p>[46: 1] SET if error is detected in label (parts of log entry might be missing or invalid)</p> <p>[45: 1] SET if volume is listed in the volume library</p> <p>[44: 1] Has a write ring</p> <p>[43: 1] SET if volume is listed in the volume directory</p> <p>[42: 1] SET if volume is unavailable for file assignment because of a HOLD command in effect for the unit</p>
<p>If the volume is a scratch volume, the following rules determine which of Words 8 through 24 are filled in:</p>	
<ul style="list-style-type: none"> ● For Minor Types 1, 2, 4, and 5, only words 22 through 24 (scratch pool) are filled in. ● For Minor Type 3 (Tape Volume Purged) entries, only the following words are filled in: 17 (new serial number), 18 (new density), and 22 through 24 (scratch pool). 	
8-10	Volume or family name (first byte is binary number of characters that follow)
<p>The remaining words of the log entry differ depending on whether the volume is a disk or tape volume.</p>	

continued

Table 3-5. Major Type 15-Minor Types 1 through 5, 12, and 13 (cont.)

Minor Type = 1 Volume Online (LOGVOLON) Minor Type = 2 Volume Offline (LOGVOLOFF) Minor Type = 3 Tape Volume Purged (LOGVOLPG) Minor Type = 4 Tape Volume Expired (LOGVOLEXP) Minor Type = 5 Tape Volume NEWFILE (LOGVOLNEW) Minor Type = 12 Tape Volume Used File (LOGVOLUSED) Minor Type = 13 Tape Volume Hold (LOGVOLHOLD)	
Words	Description
For disk volumes:	
11	Status fields: [47: 1] SET if disk has directory (for example, base pack) [46: 1] SET if disk is interchange mode [45: 1] SET if disk is IAD [44: 1] SET if disk is mirrored
The remainder of the disk log entry is not filled in for interchange-mode packs.	
12	Family index number
13	Family serial number in EBCDIC characters (serial number of original base pack for family)
14	Family creation date (Julian = YYDDD)
15	Family creation time (where time is the number of 2.4 microsecond clock ticks since midnight)
16	System serial number of site where family was originally created
For tape volumes:	
11	[20: 1] Parity [16: 1] SET if LABELKIND field is valid [15: 4] LABELKIND attribute
12	[47: 8] Label level 3 = ANSI69 label

continued

Table 3-5. Major Type 15-Minor Types 1 through 5, 12, and 13 (cont.)

<p>Minor Type = 1 Volume Online (LOGVOLON) Minor Type = 2 Volume Offline (LOGVOLOFF) Minor Type = 3 Tape Volume Purged (LOGVOLPG) Minor Type = 4 Tape Volume Expired (LOGVOLEXP) Minor Type = 5 Tape Volume NEWFILE (LOGVOLNEW) Minor Type = 12 Tape Volume Used File (LOGVOLUSED) Minor Type = 13 Tape Volume Hold (LOGVOLHOLD)</p>	
Words	Description
	<p>4 = ANSI87 label</p> <p><i>Note: Prior to the Mark 3.9.2 release, the label level was stored in field [47: 9].</i></p> <p>[37:14] Cycle</p> <p>[23: 8] Version</p> <p>[14:15] FILESECTION number</p>
13	<p>[47:16] System serial number of system that created file</p> <p>[31: 4] GENERATION attribute</p> <p>[27:11] SAVEFACTOR attribute</p> <p>[16:17] CREATIONDATE (Julian = YYDDD)</p>
14-16	Tape usercode or zeros (first byte is binary count of characters in the usercode)
17	New serial number of tape volume. (This can differ from original serial number above for SN MT entry.)
18	<p>New density</p> <p>[31: 9] New density of tape volume. This can differ from original density above for SN MT entry. (See Table 12-23, "Peripheral Densities," in the <i>A Series System Software Support Reference Manual</i>)</p>
19-21	File name identifier (for the first file on the tape volume). The first byte of the name is a binary number that counts characters in the name that follows.
22-24	<p>If the tape is a scratch tape that belongs to a scratch pool, these words contain the scratch pool name in EBCDIC characters. The first byte is a binary count of the number of characters in the scratch pool name.</p> <p>If the tape does not belong to a scratch pool, these words contain zeros.</p>

continued

Table 3-5. Major Type 15–Minor Types 1 through 5, 12, and 13 (cont.)

Minor Type = 1 Volume Online (LOGVOLON) Minor Type = 2 Volume Offline (LOGVOLOFF) Minor Type = 3 Tape Volume Purged (LOGVOLPG) Minor Type = 4 Tape Volume Expired (LOGVOLEXP) Minor Type = 5 Tape Volume NEWFILE (LOGVOLNEW) Minor Type = 12 Tape Volume Used File (LOGVOLUSED) Minor Type = 13 Tape Volume Hold (LOGVOLHOLD)	
Words	Description
25–end	Variable-length data pointed to by the LINKs in this entry

Major Type 26: Tape Library Entry (LOGMAJ_MEDIALIB)

Major Type 26 (Tape Library Entry) has been added to document log entries for media libraries. This entry is relevant only to RoboHost A users. Refer to the *A Series RoboHost A Operations, Installation, and Administration Guide* for a description of the entry format.

Section 4

DMSII Considerations

This section describes how tape management enhancements affect DMSII utilities. Unless otherwise stated, the section references in the following text refer to the DMSII manual under which the reference is located. The following manuals are affected:

- *A Series DMSII Technical Overview*
- *A Series DMSII DASDL Reference Manual*
- *A Series DMSII Utilities Operations Guide*

Table 4-1 summarizes DMSII information affected by the tape management enhancements. DMSII information that is not described in the table remains unchanged.

Table 4-1. Tape Management Enhancements to DMSII

Change	DMSII Manual Affected	Topics Affected
Recommendations for dumps	<i>DMSII Technical Overview</i>	Recommendations for maintaining a dump tape directory
	<i>DMSII Utilities Operations Guide</i>	Dump recommendations
SCRATCHPOOL file attribute for tape files	<i>DMSII Utilities Operations Guide</i>	Table of contents information for the control file, UPDATE option explanation for the DMCONTROL statement, dump tape syntax for the DUMP statement, tape medium syntax for the COPYAUDIT commands, and syntax for the VISIBLE DBS utility

For more information on the SCRATCHPOOL file attribute, see Section 3, "Programming Considerations" of this guide.

Tape Management Enhancements in the Technical Overview

The tape management enhancements affect how a dump directory is maintained as described in the *DMSII Technical Overview*.

The following text is added to "Backing Up a Database" in Section 6, "Restoring a DMSII Database."

Tapes can have the same name but must have unique serial numbers. When retrieving tapes for input, the Cartridge Tape Library (CTL) needs this serial number to mount the correct tape on a Cartridge Library Unit (CLU) device. Use one of the following techniques to provide this number:

- Define a serial number in DMSII.
- Use the DMDUMPDIR utility of DMSII.
- Use a Tape Management System (TMS) on your site that can provide a serial number.

If you do not use any of these techniques, the system can accept an incorrect tape with the same name as the correct one or suspend the task as it waits for an operator response.

For more information on CTL and CLU, see the *A Series RoboHost A Programming Guide*.

Tape Management Enhancements in DMSII Utilities

The tape management enhancements affect DMSII utilities and tape files through the SCRATCHPOOL file attribute, additional dump recommendations, and the SCRATCHPOOL command. The SCRATCHPOOL file attribute is further described under “DMUTILITY DUMP Statement” later in this section of the guide.

Table of Contents

The scratch pool name is added to the description of audit control information in the table of contents. The following text is changed in Section 2, “Control File.”

Audit Control

Audit control information includes the current audit file number, the current audit block serial number, and the limits and current values for the primary and secondary audit tape serial numbers (when auditing to designated serial numbers) or designated scratch pool, if specified.

DMCONTROL Statement

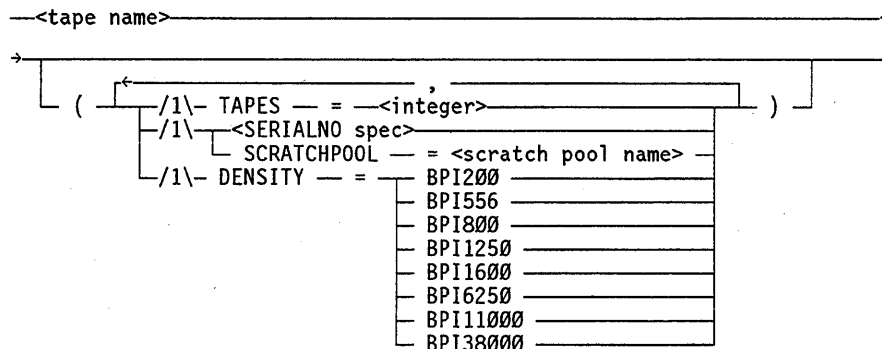
The scratch pool name is also transferred when you create a control file from an existing one. The syntax explanation for the UPDATE option is changed under “Initializing and Maintaining the Control File” in Section 5, “Initializing and Maintaining.”

Option	Explanation
UPDATE	<p data-bbox="695 323 1377 380">Causes a new control file to be created from the existing control file and the new DASDL description file after a DASDL update run.</p> <p data-bbox="695 401 1382 604">The DASDL update level must be greater than 1. The update level in the old control file must be no more than one update level less than that in the database description file. If the old control file format level is greater than the current level, an error results. If the old control file format level is less than the current format level, it is still accepted. The new control file has the update level of the database description file, and the current control file format level.</p> <p data-bbox="695 625 1360 709">If REORGANIZATION is in progress, the control file program terminates with an error; otherwise, the state flags are transferred from the old control file to the new control file.</p> <p data-bbox="695 730 1382 1016">The dynamic database parameters are transferred from the new database description file. The ranges for audit to designated serial numbers are also transferred from the description file. The dynamic audit information is transferred from the old control file. If the current tape serial number is not within the new range of designated serial numbers, it is changed to the start of the new range. Deleted structures are dropped out of the control file. Partitioned records relating to a deleted partitioned structure are marked as being deleted within the control file. Added structures are marked as INITIALIZATION REQUIRED.</p> <p data-bbox="695 1037 1300 1064">The scratch pool name for the audit tape is also transferred.</p> <p data-bbox="695 1085 1377 1283">For a reorganized structure, REORGANIZATION alters the creation timestamp when it reorganizes the structure. In addition, REORGANIZATION alters the format timestamp appropriately if it changes the format of the structure. Because partitions cannot be reorganized, the partition records section of the control file remains unaltered. If the family change bit is 1, the family specifications in the new control file remain as they were in the old control file.</p>

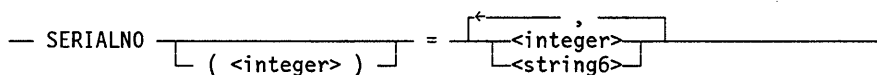
DMUTILITY DUMP Statement

The SCRATCHPOOL file attribute is added as an option to the syntax diagram and text for dump media under “Dumping a Database” in Section 6, “Backing Up the Database.”

<dump tape>



<serialno spec>



The following text is added to the syntax explanation for the <serialno spec>.

Option	Explanation
<serialno spec>	<p>Tapes can have the same name but must have unique serial numbers. When retrieving tapes for input, the Cartridge Tape Library (CTL) needs this serial number to mount the correct tape on a Cartridge Library Unit (CLU) device. Use one of the following techniques to provide this number:</p> <ul style="list-style-type: none"> ● Define a serial number in DMSII. ● Use the DMDUMPDIR utility of DMSII. ● Use a Tape Management System (TMS) on your site that can provide a serial number.

If you do not use any of these techniques, the system can accept an incorrect tape with the same name as the correct one or suspend the task as it waits for an operator response.

For more information on CTL and CLU, see the *A Series RoboHost A Programming Guide*.

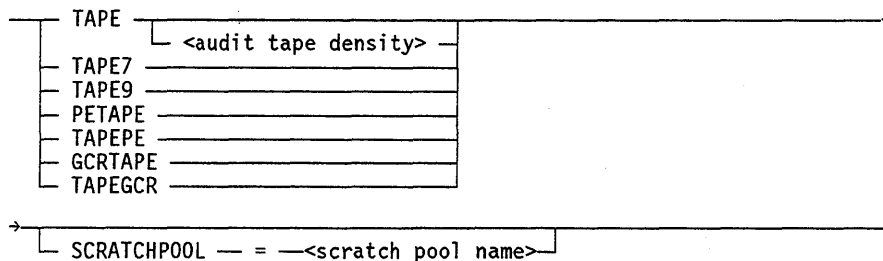
The following table explains the syntax construct added because of tape management enhancements:

Option	Explanation
SCRATCHPOOL	Designates the name of the scratch pool from which the tape is retrieved. The <scratch pool name> construct is a 17-character identifier.

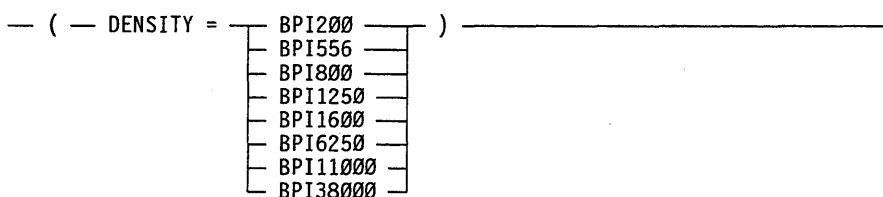
COPYAUDIT Commands

The SCRATCHPOOL file attribute is added as an option to the syntax diagram and text of tape medium for the COPY command in Section 9, "Copying and Printing Audit Files."

<tape medium>



<audit tape density>



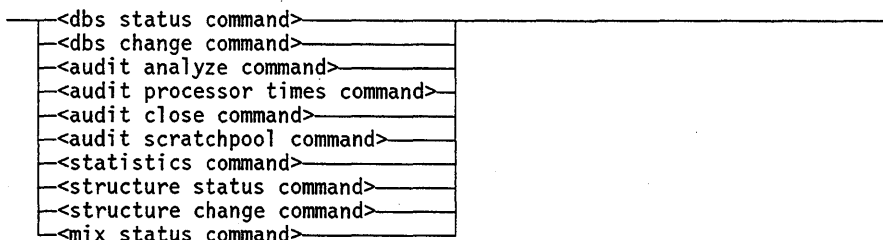
The following table explains the syntax construct added because of tape management enhancements:

Option	Explanation
SCRATCHPOOL	Used only with the COPY and QUICKCOPY commands. This option restricts output to a tape from the specified scratch pool. The <scratch pool name> construct is a 17-character identifier.

VISIBLE DBS Commands

The SCRATCHPOOL file attribute is added as a command to the syntax diagram and text of the VISIBLE DBS utility in Section 11, "Communicating with the Database."

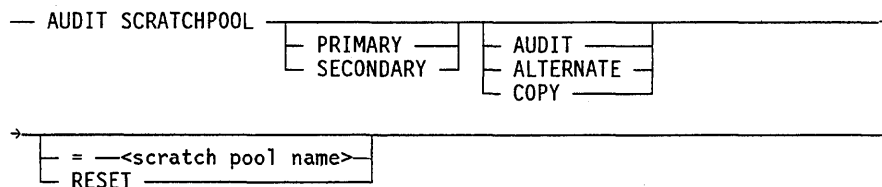
<visible dbs command>



AUDIT SCRATCHPOOL Command

The AUDIT SCRATCHPOOL command identifies the name of the scratch pool from which a tape is selected.

AUDIT SCRATCHPOOL command



The AUDIT SCRATCHPOOL command without any options displays information about the scratch pool such as its name.

Option	Explanation
PRIMARY or SECONDARY	Designates that the scratch pool tape is selected for the primary or secondary audit, alternate, or copy.
AUDIT, ALTERNATE, or COPY	Designates that the scratch pool tape is selected for the audit file, alternate audit file, or audit file copy. If none of these options are designated, the scratch pool name is assigned to all audit files even if a scratch pool name was previously defined for a particular audit file. For example, if the following command previously defined the scratch pool name for primary audits to XTEST: AUDIT SCRATCHPOOL PRIMARY = XTEST Then the following command would define the scratch pool names to YTEST, including the primary audits previously defined to XTEST: AUDIT SCRATCHPOOL = YTEST
<scratch pool name>	A 17-character identifier that names the scratch pool form which a tape is selected.
RESET	Designates no scratch pool attribute. If a scratch pool name was previously assigned, that name is deleted.

Section 5

TAPEMANAGER Library Implementation

A tape management system (TMS) is any application that is designed to record or control the use of tapes on a system. The A Series operating system now provides additional MCP support for some functions of a TMS. To take advantage of this support, you must implement these TMS functions as a support library with the function name TAPEMANAGER. You must include certain standard procedure declarations in the TAPEMANAGER library, but you supply the code for these procedures. The MCP calls these procedures whenever certain tape-open requests or tape-purge requests occur, and the TAPEMANAGER library has the opportunity to permit, deny, or modify the actions.

A Series systems do not require a TAPEMANAGER library. However, you might want to develop and install a TAPEMANAGER library because it can control the use of tapes. A TAPEMANAGER library receives the following information about tape-open and tape-purge requests:

- The file name and other file attributes of any tape file that is being opened or closed
- The serial number and label of any tape volume that is being purged

The role of the TAPEMANAGER library is to control tape file and volume access and usage through these tape-open and tape-purge requests. A tape management system can also track and record current events and the status of tapes in a database, and initiate actions through requests and responses to and from the A Series operating system.

Figure 5-1 illustrates the interaction of a TMS in an A Series operating system environment.

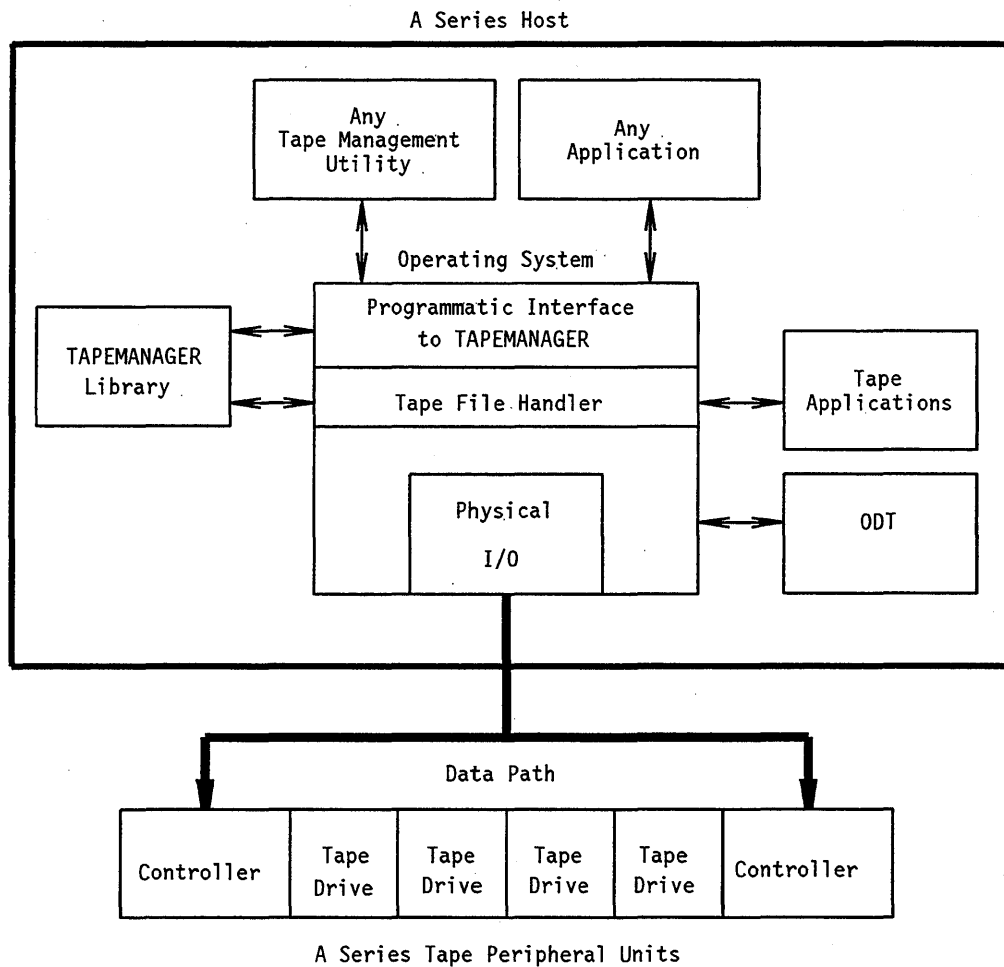


Figure 5-1. The Tape Management System (TMS) Environment

This section discusses the following topics:

Topic	Refer to...
Overview of the TAPEMANAGER library functions	"TAPEMANAGER Functions for TMS Design"
Information sources outside the TAPEMANAGER library	"Event and Status Information Retrieval for TMS Design"
Overall issues to consider when you implement a TAPEMANAGER library	"General Considerations for TAPEMANAGER Libraries"

continued

continued

Topic	Refer to...
Security issues related to TAPEMANAGER libraries	"Security Considerations"
The circumstances that cause the MCP to call various TAPEMANAGER procedures, and details about their interaction	"Interaction with the MCP"
The declarations and parameter descriptions for TAPEMANAGER procedures	"Exported TAPEMANAGER Procedures"
Procedures exported by the MCP that are related to tape management	"Exported MCP Procedures for Tape Management"
The procedures system operators use to install, initiate, and terminate the TAPEMANAGER library	"Operations Interface to the TAPEMANAGER Library"

TAPEMANAGER Functions for TMS Design

The MCP supports six functions as entry points that the TAPEMANAGER library can include to aid in the design and implementation of tape management systems (TMSs). These functions can help record or control the use of tapes on the system for the following tasks:

- Opening tape files
- Purging tape volumes
- Handling programmatic calls to the TAPEMANAGER library

Opening a Tape File

When a process attempts to open a tape file and the TAPEMANAGER library is running, the TAPEMANAGER library might call any of three functions:

- **CHECK_TAPE_ASSIGNMENT**
 This function determines whether the process is permitted to open a requested tape file when the file is not assigned to a tape volume, and the UNITNO file attribute is not assigned a tape unit number. CHECK_TAPE_ASSIGNMENT can inspect the file attribute and task attribute parameter values, and modify certain file attributes. The procedure must return an indication of whether the tape volume assignment is allowed or denied.
- **CHECK_TAPE_UNIT_ASSIGNMENT**
 This function determines whether the process is permitted to open a requested tape file when the open request specifies a tape unit number.

- **CHECK_TAPE_REOPEN**

This function determines whether the process is permitted to reopen a requested tape file that is already assigned to a tape volume.

Purging a Tape Volume

When a process attempts to purge a tape volume and the TAPEMANAGER library is running, the TAPEMANAGER library calls the CHECK_TAPE_PURGE function. This function determines whether the process is permitted to purge the tape.

Handling Programmatic Calls

When a process calls the CALL_TAPEMANAGER exported MCP procedure and the TAPEMANAGER library is running, CALL_TAPEMANAGER in turn calls the PROGRAM_INTERFACE function in the TAPEMANAGER library. This interface enables application programs to communicate with the TAPEMANAGER library. It also provides a method for a TMS that includes additional programs to enable those programs to communicate with the TAPEMANAGER library.

The MCP calls the CONTROL_INTERFACE function to initiate and to terminate the TAPEMANAGER library. An operator can use the system commands SEND TM START and SEND TM QUIT to start up and shut down the TAPEMANAGER library.

For details about the interaction between the MCP and the TAPEMANAGER library, refer to “Interaction with the MCP” later in this section. For details about each TAPEMANAGER procedure, refer to “Exported TAPEMANAGER Procedures” later in this section.

Event and Status Information Retrieval for TMS Design

The MCP interface to TAPEMANAGER is just one of the features provided by the A Series operating system that can help you design and implement tape management systems (TMSs).

The TAPEMANAGER library designates and restricts tape usage based on file assignment, reopen, and purge requests. The TAPEMANAGER interface is particularly suited to enforcing restrictions on tape usage. TAPEMANAGER can enforce such limits effectively because the MCP asks the permission of TAPEMANAGER before it opens any tape file or purges any tape. However, TAPEMANAGER is less suited to monitoring or recording successful tape actions, because the MCP notifies TAPEMANAGER of requested actions rather than successfully completed actions.

All A Series systems automatically track the status and activity of their peripheral units, and allow programs to initiate certain system actions. You can retrieve this event and status information to monitor current tape usage throughout the system, and initiate control actions as needed.

The following system features can play a role in the design of a TMS:

- Event reports in the system log
- Tape label creation events
- Status reports about tapes
- Programmatic initiation of tape operations

A complete TMS might include multiple programs that make use of these features for a variety of purposes. The following information describes the features briefly, with notes on how they might be used in a TMS.

Monitoring Tape Usage Events

The system log (SUMLOG) maintains a log entry for each system event that affects a task, file, unit, or volume. One way for a TMS to monitor successful tape actions is by calling the exported MCP procedure `REPORT_LOG_ENTRIES`. This procedure requests the MCP to forward copies of selected types of log entries to a DCALGOL queue whenever those entries are logged. Table 5-1 lists the log entry types relevant to tape management.

Table 5-1. Log Entries for Tape Management

Major Type	Minor Type	Name	Event Type
1	1	BOJ †	Task
1	2	EOJ †	Task
1	3	BOT †	Task
1	4	EOT †	Task
1	5	File open *	File and task
1	6	File close *	File and task
2	16	IOP I/O error †	Unit and task
2	21	I/O exception †	Unit and task
2	24	CLU hardware report ‡	TAPESERVER
15	All	Volume status *	Volume
26	1	Tape library ‡	TAPESERVER

Legend

- * Documented in Section 3, "Programming Considerations"
- † Documented in the A Series System Software Support Reference Manual
- ‡ Documented in the A Series RoboHost A Operations, Installation, and Administration Guide

Note that each of these log entry types is relevant to tape management only part of the time. For example, a File Open entry might record that a tape file is open, or it might record that a different kind of file is open.

A TMS is concerned with both tape files and tape volumes. One subtype of a volume status entry reports when a volume is labeled. A file open entry reports when a file is written to a specific volume. After a task finishes, the volume might be taken offline. The volume could be purged, which would be reported as a volume status change. A TMS can direct a file to be written to a specific physical volume as a control measure.

For a description of `REPORT_LOG_ENTRIES` and documentation of log entry formats, refer to the *A Series System Software Support Reference Manual*.

Another way a TMS can obtain information about tape events is by calling the exported MCP procedure `STATUS_CHANGE_REQUEST` to monitor tape label creation. This procedure requests the MCP to write information about selected types of system events to a DCALGOL queue whenever those events occur. Tape label creation is the only tape management event that is reported; it is specified to `STATUS_CHANGE_REQUEST` as event type 7. For a description of `STATUS_CHANGE_REQUEST`, refer to the *A Series System Software Support Reference Manual*.

If you design a TMS to monitor tape activity with `REPORT_LOG_ENTRIES` and `STATUS_CHANGE_REQUEST` calls, you might want to place these calls in a process that runs independently from the TAPEMANAGER library. This technique can be desirable because TAPEMANAGER is subject to termination and restart by the operator, and so might not be able to monitor tape activity continuously.

Monitoring Current Tape Status

A TMS can use the DCALGOL `GETSTATUS` function to obtain information about the status of tape units on the system. The `GETSTATUS` type 5 call (Unit Requests) can return information for tapes, including the `CYCLE`, `DENSITY`, `SAVEFACTOR`, `SERIALNO`, and `VERSION` attributes. Also, the type 5 call can return the file name and mix number of the process assigned to the tape unit. For more information, refer to the *A Series GETSTATUS/SETSTATUS Programming Reference Manual*.

A TMS can also use the DCALGOL `SYSTEMSTATUS` function to determine the number and type of tape units available on the system. The Type 13, 21, and 22 `SYSTEMSTATUS` calls return a unit information word that includes the tape subtype and density for all tape units. The Type 21 call also includes information on tape file attributes such as `CREATIONDATE`, `CYCLE`, `GENERATION`, `SAVEFACTOR`, `SERIALNO`, and `VERSION`. For further information, refer to the *A Series SYSTEMSTATUS Programming Reference Manual*.

Initiating Programmatic Tape Operations

A TMS can use the DCALGOL `SETSTATUS` function to hold, rewind, or purge tape units, and to assign serial numbers to tape volumes. Refer to "SETSTATUS Changes" in Section 3 for complete details.

The DCALGOL *DCKEYIN* statement enables a program to pass a request for a system command or job to the operating system and receive a response. As an alternative to SETSTATUS, a TMS can use DCKEYIN to initiate tape operations such as holding, rewinding, or purging tape units through system commands.

Note: The system response to a SETSTATUS request is encoded so that a program can interpret the data. The system response to a DCKEYIN request is alphanumeric text designed so that a system operator can interpret it.

CALL_TAPEMANAGER Procedure

If a TMS includes multiple programs, these programs can communicate with the TAPEMANAGER library by using the MCP exported procedure CALL_TAPEMANAGER. This procedure could also serve as an interface between TAPEMANAGER and any application program that uses tapes. For further information about this procedure, refer to "CALL_TAPEMANAGER" later in this section.

General Considerations for TAPEMANAGER Libraries

Because the TAPEMANAGER procedures require features, such as file parameters and return values, that are not available in most programming languages, you can use only ALGOL, BDMSALGOL, DCALGOL, DMALGOL, and NEWP for the procedures exported by the TAPEMANAGER library. However, you could use the Binder to bind procedures written in these languages into a library program written in another language. For information about using the Binder, refer to the *A Series Binder Programming Reference Manual*.

Only one TAPEMANAGER library can be installed on the system at a time. The TAPEMANAGER library must be associated with the function name TAPEMANAGER, as described under "Installing the TAPEMANAGER Library" later in this section.

TAPEMANAGER should be a SHARED BY ALL library to ensure that all user processes share the same instance of the library. The freeze should typically be of type TEMPORARY. If you use a CONTROL or PERMANENT freeze, the library or the system operator must take extra steps to cause the SEND TM QUIT system command to have the desired effect. Refer to "Shutting Down" later in this section.

An operator can terminate the TAPEMANAGER library at any time with the SEND TM QUIT form of the SEND (Send Message) system command. Any information the TAPEMANAGER library stores in memory is lost when TAPEMANAGER terminates. Therefore, any information of permanent value that is passed to a TAPEMANAGER procedure should be written to some permanent storage medium before the procedure exits.

Unisys recommends that you use the MP (Mark Program) system command to mark as private and assign an identity to the TAPEMANAGER library object code file. This identity appears in any system messages or display messages generated by a process that is executing TAPEMANAGER code.

You might want to avoid the use of statements that enter into transaction state in TAPEMANAGER exported procedures. Transaction state is normally invoked by processes before they attempt to update records in a Data Management System II (DMSII), Semantic Information Manager (SIM), or Structured Query Language Database (SQLDB) database.

Problems can arise because TAPEMANAGER exported procedures execute on user process stacks, and any of these user processes might already have a database open and in transaction state. It would be unusual for a process to attempt to open a tape while the process is in transaction state against a database, because database transactions are usually time-critical. However, if such a user process is already in transaction state, and the TAPEMANAGER exported procedure itself enters transaction state, the user process encounters an error. If the user process does not handle the error itself, the user process becomes suspended as described under "Error Handling by the MCP" later in this section.

If TAPEMANAGER exported procedures simply read database records, rather than updating them, then the procedures do not need to enter into transaction state. However, even in this case problems can arise if TAPEMANAGER is a SHARED BY ALL library (which it typically is) and the database is declared globally to the exported procedures. A variety of undefined results can occur if two or more user processes of this library have transactions in progress against the database at the same time.

For example, if two user processes are executing FIND NEXT statements at the same time without locks, each user process is likely to get every other record. Therefore, if the TAPEMANAGER exported procedures access a database, the library should use events to ensure that only one user process can have a transaction in progress against the database at a time.

Any TAPEMANAGER exported procedure that procures globally-declared events should include an EXCEPTION procedure to liberate these events if the user process terminates abnormally. Without the EXCEPTION procedure, the MCP cannot perform cleanup processing after an abnormal exit from a TAPEMANAGER procedure. EXCEPTION procedures are available in DCALGOL, DMALGOL, and NEWP.

For more information about using libraries on A Series systems, refer to the *A Series Task Management Programming Guide*.

TAPEMANAGER Error Handling

When a user process opens or purges a tape file, the MCP invokes one or more TAPEMANAGER procedures that execute on the user process stack. Thus, the MCP considers any errors in TAPEMANAGER procedures to be errors committed by the user process. However, the MCP takes special steps to ensure that it does not discontinue processes that use tapes because of errors in the TAPEMANAGER code. Additionally, you can design TAPEMANAGER to handle certain types of errors on its own.

Fault Handling within TAPEMANAGER

Faults are those program errors that receive a HISTORYCAUSE task attribute value of 4 (FAULTCAUSEV) and that normally cause the process to be discontinued with the F-DS message. In any TAPEMANAGER exported procedure that you write in ALGOL, you can include an ON statement to specify special handling for any faults committed by that procedure. For example, the ON statement could modify the procedure result to indicate whether the requested action is to be allowed or denied (whichever is the preferred default at your site).

Note: If TAPEMANAGER handles faults with an ON statement, then TAPEMANAGER should also provide reasonable diagnostic features to enable support personnel to determine which fault occurred, and why. For example, TAPEMANAGER could display a message describing the fault, or could provide an option that enables operators to request a program dump.

Error Handling by the MCP

While a user process is executing TAPEMANAGER code, the only cause that can discontinue the user process is a DS (Discontinue) system command. The MCP protects the user process from any faults that are not handled by ON statements in TAPEMANAGER, as well as from all other types of program errors. If any fault or error occurs that would normally terminate the user process, the MCP takes the following actions:

1. Issues display messages that show the current values of the HISTORYTYPE, HISTORYCAUSE, HISTORYREASON, and STOPPOINT task attributes of the user process. These messages appear in the MSG (Display Messages) system command output, and are also written to the system log as Major Type 3, Minor Type 7 (Display Message) entries. You can use these values to help debug the TAPEMANAGER library.
2. Resets the HISTORY, HISTORYTYPE, HISTORYCAUSE, and HISTORYREASON task attributes of the user process to zero.
3. Suspends the user process and displays an RSVP request, system message MSRMISC221: "USER PROCESS FAULTED OR DSED WHILE IN TAPEMANAGER LIBRARY CODE."

The following table shows the possible operator responses to this RSVP request and the effects of these responses:

Response	Effect
OK	The MCP invokes the same library procedure again.
DS	The MCP discontinues the user process.

continued

continued

Response	Effect
OF	<p>The open or purge request continues as if the TAPEMANAGER library were not installed.</p> <p>Because TAPEMANAGER might be used to enforce tape security at some sites, the MCP limits the use of the OF response. The OF response can be submitted from the following sources:</p> <ul style="list-style-type: none">● Operator display terminals (ODTs)● Remote ODTs; that is, remote terminals granted ODT status by the REMOTESPO (Activate REMOTESPO) system command● Nonusercoded Menu-Assisted Resource Control (MARC) sessions● DCKEYIN functions in nonusercoded DCALGOL processes <p>The MCP ignores the OF response if it is submitted from CANDE sessions, usercoded MARC sessions, or usercoded DCALGOL processes.</p>

If the TAPEMANAGER library is discontinued by a DS command, then the MCP also protects any user processes linked to the library and suspends them with the "USER PROCESS FAULTED OR DSED WHILE IN TAPEMANAGER LIBRARY CODE" message. If the TAPEMANAGER library is discontinued with the *SEND TM QUIT* form of the SEND (Send Message) system command, any user processes linked to the library continue processing, and the library terminates only after all user processes have exited the TAPEMANAGER procedures.

If the user process was initiated by a job, and you use the DS command to discontinue that job, the MCP suspends the user process with the "USER PROCESS FAULTED OR DSED WHILE IN TAPEMANAGER LIBRARY CODE" message described previously. The job remains in the A (Active Entries) mix display, but the MCP has already set the history task attributes of the job as follows: HISTORYTYPE = 4 (DSEDV), HISTORYCAUSE = 1 (OPERATORCAUSEV), and HISTORYREASON = 2 (JUSTDSEDV). The job actually terminates only when the user process terminates.

Security Considerations

The MCP sets the following Boolean library attributes to TRUE for the TAPEMANAGER function name:

- MCPINIT. This attribute ensures that only the MCP can initiate the library by way of the TAPEMANAGER function name.
- ONEONLY. This attribute ensures that only one version of the library code file can be active at a time. In other words, the old version of the library must be terminated before a new version can be initiated. Refer to "Replacing the TAPEMANAGER Library with a Newer Version" later in this section.

After the MCP initiates the library, the library linkage class determines which user processes are allowed to link to the library. The MCP associates a linkage class of 1 with the TAPEMANAGER library function name. This linkage class ensures that only the MCP and selected system libraries can link to the library by using a LIBACCESS

value of BYFUNCTION. Furthermore, after the MCP links to the TAPEMANAGER library, the MCP also prevents other processes from linking to the library BYTITLE. For a discussion of library linkage classes, refer to the *Task Management Programming Guide*.

It is possible for any process to link to the library by using a LIBACCESS value of BYTITLE, if the process makes the linkage before the MCP does. To prevent such unauthorized accesses, you can assign the library code file a SECURITYTYPE file attribute of PRIVATE. Even when this file attribute is set, privileged user processes or user processes with the same usercode as the TAPEMANAGER code file can link to the library BYTITLE.

Note: Normally the MCP should be the only user linked to the TAPEMANAGER library. If other user processes are linked to the library, the SEND TM QUIT form of the SEND (Send Message) system command cannot successfully cause the library to terminate.

For further information about TAPEMANAGER library initialization, refer to “Starting Up” later in this section.

Interaction with the MCP

The MCP calls TAPEMANAGER exported procedures in response to a number of types of events that involve tape activity. The following pages discuss the situations that result in such procedure calls, and the types of interaction that can take place in each of these situations. For detailed descriptions of the various exported TAPEMANAGER procedures, refer to “Exported TAPEMANAGER Procedures” later in this section.

TAPEMANAGER Library Start Up and Shut Down

Figure 5-2 shows the status changes of the TAPEMANAGER library during start up and shut down. In this figure, the following conventions apply:

- The elliptical boxes contain the TAPEMANAGER library status value, which changes from INACTIVE to ACTIVE at start up, and from ACTIVE to INACTIVE at shut down.
- The command or event that causes a status change is specified between the two boxes.
- The action taken as a result of the status change is enclosed in square brackets.

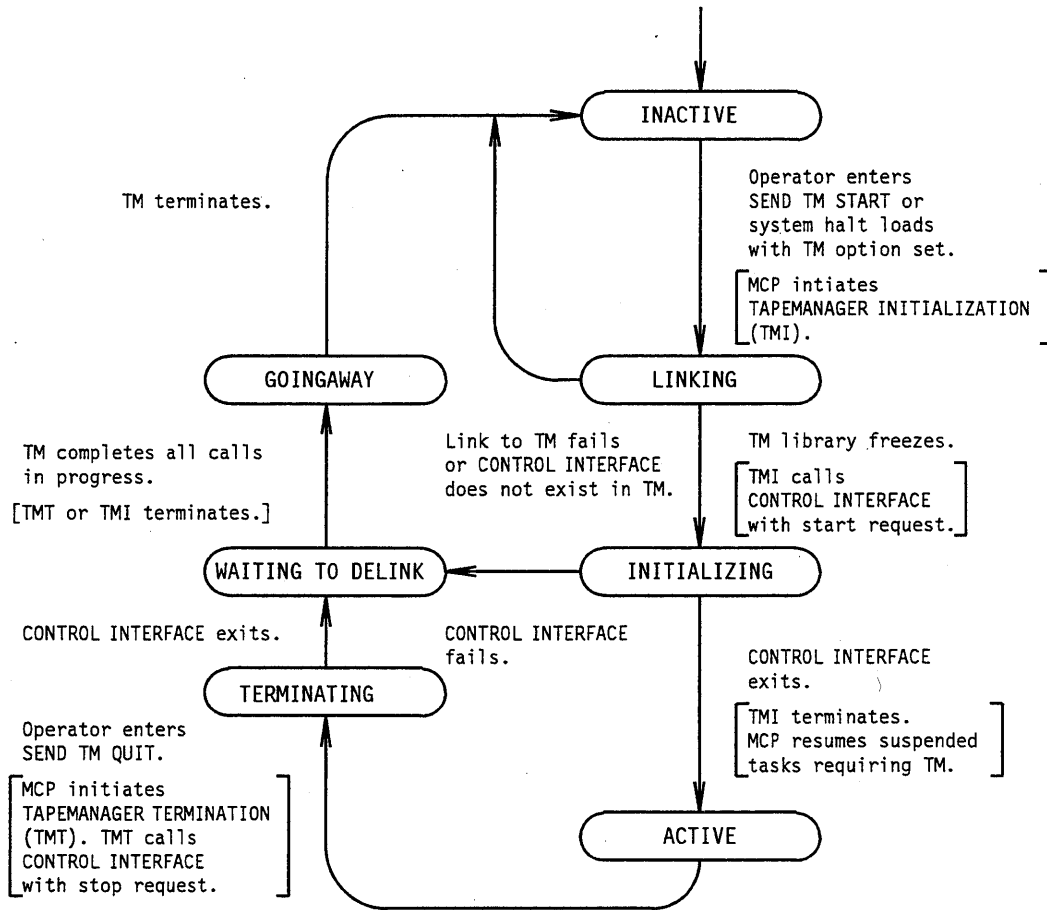


Figure 5-2. TAPEMANAGER Library Status Change Diagram

Starting Up

When the MCP starts up the TAPEMANAGER library, its state changes from inactive to active. The MCP suspends any task that calls TAPEMANAGER until the library becomes active.

Before you can initiate the TAPEMANAGER library, you must do the following:

- Define the TAPEMANAGER function name with the SL (Support Library) system command.
- Set the TAPEMANAGER system option with the SYSOPS (System Options) system command.

Initiating TAPEMANAGER

To initiate TAPEMANAGER, either send a SEND TM START system command or perform a system halt/load. In either of these cases, the MCP changes the TAPEMANAGER status from INACTIVE to LINKING. Refer to "Inspecting the TAPEMANAGER Status" later in this section. The MCP then initiates the visible independent runner TAPEMANAGER_INITIALIZATION to initiate the library.

Linking to the TAPEMANAGER Library

TAPEMANAGER_INITIALIZATION attempts to link to the TAPEMANAGER library with the LINKLIBRARY function. Library linkage can fail if the TAPEMANAGER function is not defined, if the library code file is missing or is not a library, and so on. If library linkage fails, TAPEMANAGER_INITIALIZATION becomes suspended with the RSVP request, system message MSRMISC204: "UNABLE TO LINK TO TAPEMANAGER. ENTER OK TO RETRY." After you attempt to remedy the situation, for example by copying a corrected version of the library to its intended location, you can enter an OK (Reactivate) system command to cause TAPEMANAGER_INITIALIZATION to retry the linkage.

If the linkage does not immediately fail, the MCP initiates the TAPEMANAGER library code file. Before TAPEMANAGER executes a FREEZE statement, it can perform preparatory functions such as opening a database.

Verifying TAPEMANAGER Procedures

When TAPEMANAGER freezes, the MCP determines whether the standard TAPEMANAGER procedures are exported by the library. These procedures are described under "Exported TAPEMANAGER Procedures" later in this section. If one or more of these procedures are missing, the MCP displays the system warning message MSRMISC209: "WARNING: TAPEMANAGER LIBRARY IS MISSING ONE OR MORE OBJECTS."

If the CONTROL_INTERFACE procedure is missing, the MCP displays the following additional system messages MSRMISC210 and MSRMISC208:

```
TAPEMANAGER LIBRARY IS MISSING CONTROL_INTERFACE PROCEDURE
```

```
TAPEMANAGER LIBRARY WILL NOT BE USED
```

If CONTROL_INTERFACE is missing, the MCP does not use the TAPEMANAGER library. If CONTROL_INTERFACE is present, but other procedures are missing, the MCP uses only the procedures that are present. For example, if the CHECK_TAPE_PURGE procedure is missing, the MCP does not consult the TAPEMANAGER library when purge requests occur.

When the library freezes, the MCP changes the TAPEMANAGER status to INITIALIZING, and TAPEMANAGER_INITIALIZATION calls the CONTROL_INTERFACE library procedure with a start request. The CONTROL_INTERFACE procedure can perform initialization functions that are

specific to the TAPEMANAGER implementation, such as opening a database if it has not already done so.

Completing Initialization

When `CONTROL_INTERFACE` exits, the TAPEMANAGER status changes to `ACTIVE` and the `TAPEMANAGER_INITIALIZATION` independent runner terminates. At this point, the library is ready for use. When the TAPEMANAGER library is active, the MCP allows any task that calls the library to proceed.

If you enter a `SEND TM START` command while the TAPEMANAGER status is `LINKING` or `INITIALIZING`, the system displays the system message `MSRMISC222: "TAPEMANAGER LIBRARY IS ALREADY ACTIVE."`

Shutting Down

When the MCP shuts down the TAPEMANAGER library, its state changes from active to inactive, as shown in Figure 5-2. When the TAPEMANAGER library is in any state other than active, the MCP suspends any task that calls the library.

When you enter the `SEND TM QUIT` form of the `SEND` (Send Message) system command, the MCP takes the following actions:

- Changes the status of the library to `TERMINATING`. You can display this status, as described under "Inspecting the TAPEMANAGER Status" later in this section.
- Allows any processes that are currently executing TAPEMANAGER procedures to continue normally.
- Suspends any processes that attempt tape actions that would require new calls on the TAPEMANAGER library. Refer to "Responding to Processes Waiting on the TAPEMANAGER Library" later in this section.
- Initiates the `TAPEMANAGER_TERMINATION` visible independent runner, which in turn invokes the `CONTROL_INTERFACE` procedure of the TAPEMANAGER library with a stop request. If you used the `SEND TM QUIT <text>` command, then the text is passed to the `REQUEST` parameter of `CONTROL_INTERFACE`. The meaning of this text is determined by the TAPEMANAGER library implementation.

If the library has a `CONTROL` freeze, the library should exit the control procedure when the `CONTROL_INTERFACE` procedure receives the stop request. `CONTROL_INTERFACE` can notify the control procedure to exit by causing some event in the library that is declared global to both procedures. Exiting the control procedure changes the library to a `TEMPORARY` library.

The `TAPEMANAGER_TERMINATION` process terminates after the `CONTROL_INTERFACE` procedure returns. The MCP changes the TAPEMANAGER status to `WAITING TO DELINK`. TAPEMANAGER remains in this state until it completes all calls currently in process. At this point, the MCP delinks from the library and changes the TAPEMANAGER status to `GOINGAWAY`.

The exact effects of the GOINGAWAY assignment vary, depending on the freeze type of the library. These varying effects are as follows:

- If the library freeze is TEMPORARY, then the library unfreezes and resumes execution when the MCP sets the TAPEMANAGER status to GOINGAWAY. The TAPEMANAGER status remains GOINGAWAY until the TAPEMANAGER process terminates.
- If the library originally had a CONTROL freeze, CONTROL_INTERFACE should have already notified the control procedure to exit so the freeze could become TEMPORARY.
- If the library freeze is PERMANENT, it remains frozen until the operator enters a THAW (Thaw Frozen Library) system command. The THAW command changes the library freeze to TEMPORARY and thus enables it to unfreeze and resume execution. The TAPEMANAGER status remains GOINGAWAY until the TAPEMANAGER process terminates.

If you enter a SEND TM START system command while the library status is TERMINATING, WAITING TO DELINK, or GOINGAWAY, the MCP delays the request until the library has terminated.

Refer to “Terminating the TAPEMANAGER Library” later in this section for information about measures you can take if the library takes longer than expected to terminate.

Opening Files

When a process attempts to open a tape file and the TAPEMANAGER library is installed and running, the MCP might call any of the following procedures in the TAPEMANAGER library, depending on the circumstances:

- CHECK_TAPE_ASSIGNMENT
- CHECK_TAPE_UNIT_ASSIGNMENT
- CHECK_TAPE_REOPEN

The MCP passes information to the TAPEMANAGER procedures in parameters, and the TAPEMANAGER procedures return results that either accept, deny, or modify the request. If you modify an open request with system commands such as FA (File Attribute) or OU (Output Unit), the MCP invokes the TAPEMANAGER procedures again to verify that the modified request is acceptable.

CHECK_TAPE_ASSIGNMENT

The MCP calls the CHECK_TAPE_ASSIGNMENT procedure when the following conditions are all true:

- A process attempts to open a file with a KIND file attribute value equal to TAPE or DONTCARE. Files with KIND equal to DONTCARE can be assigned to tape units.
- The file has not already been assigned to a tape volume.

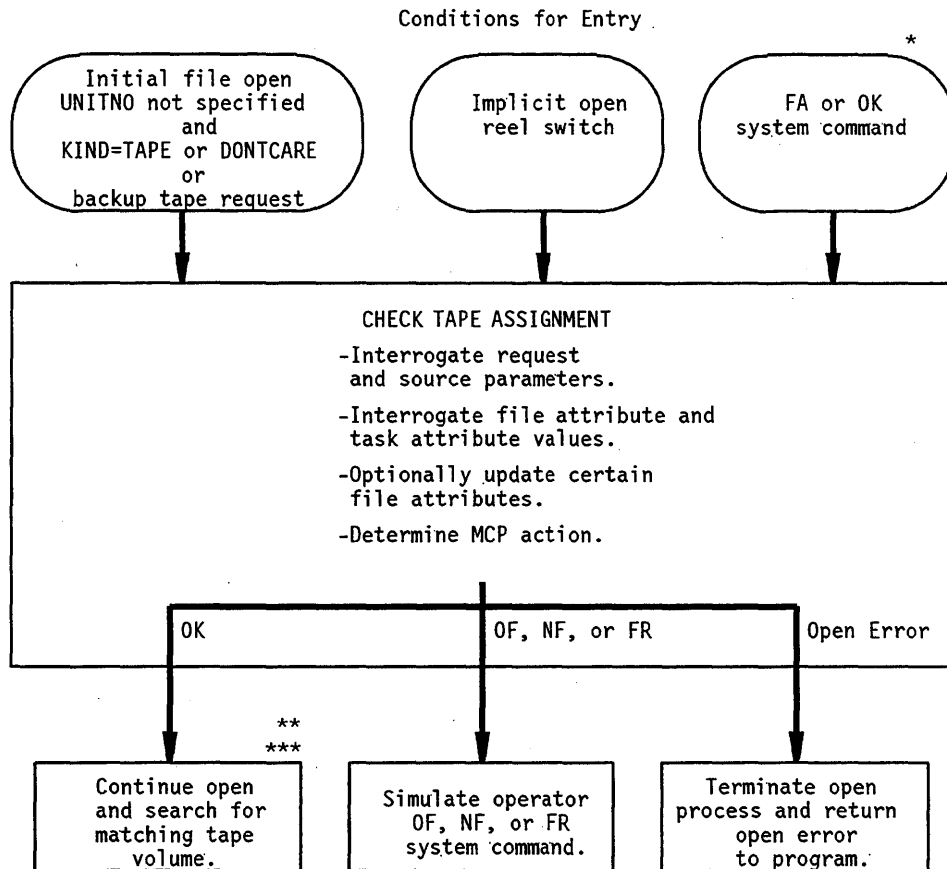
- The tape unit number has not been specified.

The MCP also calls the CHECK_TAPE_ASSIGNMENT procedure after an initial open request for a specific tape unit was denied; the MCP then determines whether the open request can proceed with no tape unit specified.

The MCP calls CHECK_TAPE_ASSIGNMENT under two additional circumstances:

- A reel switch is handled as an implicit open request.
- The FA and OK system commands indicate an operator request to proceed with the file open request.

Figure 5-3 illustrates the flow of information between the CHECK_TAPE_ASSIGNMENT procedure and the operating system.



- * CHECK TAPE ASSIGNMENT or CHECK TAPE UNIT ASSIGNMENT might have been called previously.
- ** CHECK TAPE ASSIGNMENT or CHECK TAPE UNIT ASSIGNMENT might be called again.
- *** Action taken if entry point missing is from TAPEMANAGER.

Figure 5-3. Information Flow in CHECK_TAPE_ASSIGNMENT

When a process requests a reel switch, it is attempting a tape I/O operation that cannot proceed until the logical file is reassigned to the next volume of a multivolume set. For output tapes, a scratch tape can serve as the continuation reel. For input tapes, the combination of the FILESECTION file attribute and the SERIALNO file attribute identifies the tape that is needed. Before searching for the continuation volume, the MCP invokes the CHECK_TAPE_ASSIGNMENT procedure to determine if TAPEMANAGER permits this reel switch.

If the open request is for a tape file on an unspecified unit, the MCP invokes the CHECK_TAPE_ASSIGNMENT procedure. The MCP passes various parameters, including the file variable of the tape file and the task variable of the process that declared the file.

CHECK_TAPE_ASSIGNMENT can inspect file and task attributes, and can modify many of the file attributes.

The CHECK_TAPE_ASSIGNMENT procedure result can specify any of several responses to the open request:

- If the procedure rejects the open request, it returns an open error to the opening process.
- If the procedure accepts the open request, the MCP determines whether the requested tape is available.
- If the procedure returns an OF result, the MCP takes the same action as if an OF (Optional File) system command had been entered. The MCP determines whether the file that received the OF result was originally declared with the OPTIONAL file attribute set to TRUE.

If OPTIONAL is TRUE, then the MCP permits the open operation to continue. However, the first time the process attempts to read from or write to the file, the MCP returns an end of file indication.

If OPTIONAL is FALSE, the MCP returns an open error.

- If the procedure returns an NF result, the MCP takes the same action as if the NF (No File) system command had been entered. The open operation completes, and the MCP returns an end-of-file result the first time the process reads from or writes to the file.
- If the procedure returns an FR result, the MCP takes the same action as if the FR (Final Reel) system command had been entered. The open operation completes, and the MCP returns an end-of-file result the first time the process reads from or writes to the file.
- If the MCP returns a nontape result, the MCP continues with the open operation, but excludes tape units from the search. You might want CHECK_TAPE_ASSIGNMENT to return this response when the file being opened has a KIND value of DONTCARE.

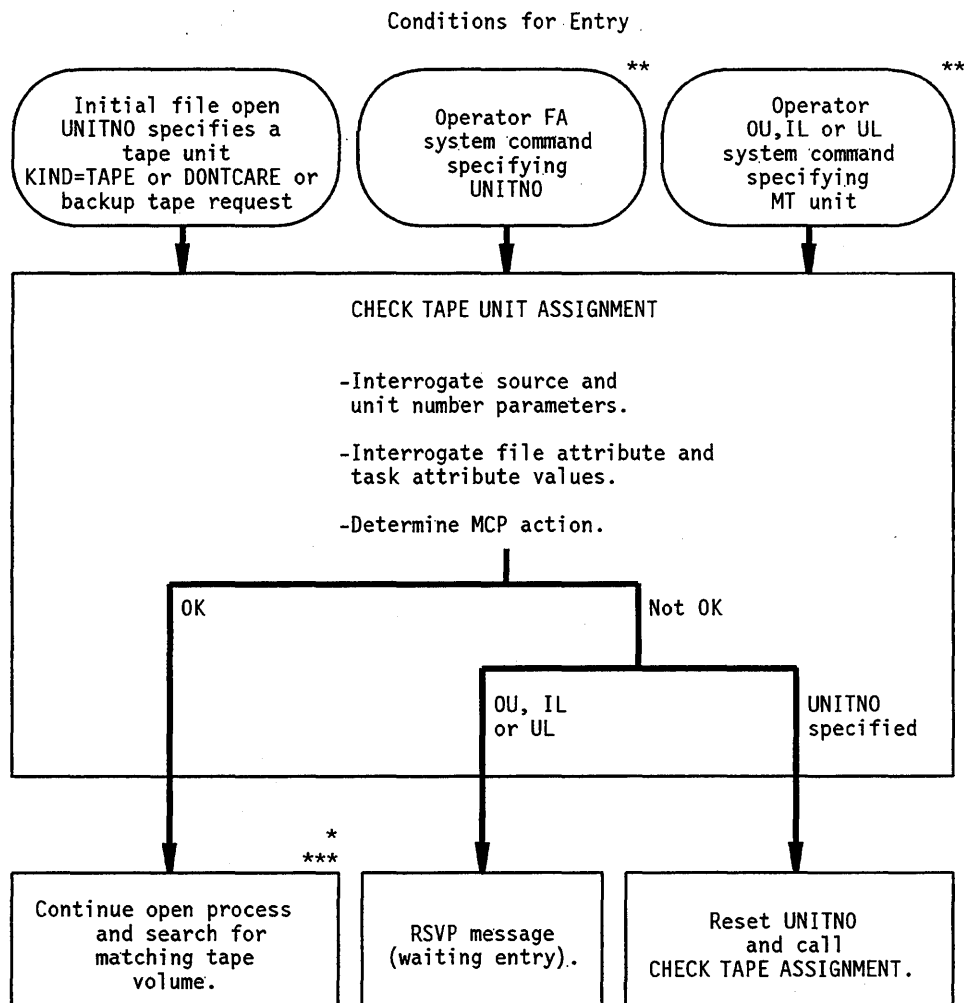
CHECK_TAPE_UNIT_ASSIGNMENT

The MCP calls CHECK_TAPE_UNIT_ASSIGNMENT when a process attempts to open a tape file (KIND=TAPE) that has not yet been assigned to a tape volume, and the UNITNO file attribute has been assigned a tape unit number.

The MCP also calls this procedure if the file KIND is equal to DONTCARE and the UNITNO is a tape unit.

The MCP calls CHECK_TAPE_UNIT_ASSIGNMENT regardless of whether UNITNO was assigned by the opening process or by a system command such as IL (Ignore Label), OU (Output Unit), or UL (Unlabeled).

Figure 5-4 illustrates the flow of information between the CHECK_TAPE_UNIT_ASSIGNMENT procedure and the operating system.



- * Action taken if entry point is missing from TAPEMANAGER.
- ** CHECK TAPE ASSIGNMENT or CHECK TAPE UNIT ASSIGNMENT might have been called previously.
- *** CHECK TAPE ASSIGNMENT or CHECK TAPE UNIT ASSIGNMENT might be called again.

Figure 5-4. Information Flow in CHECK_TAPE_UNIT_ASSIGNMENT

If the unit number is that of a nontape unit, the MCP suspends the opening process with the RSVP request, system message MSRFOUT23: "<unit identifier> TYPE DISAGREES WITH KIND, REQUIRES MT # <unit number>." The unit type is the unit type of the unit with the specified unit number.

The possible operator responses to the RSVP message are the FA (File Attributes), OU (Output Unit), and DS (Discontinue) system commands.

If a tape file open request is waiting for an operator response, the operator can enter one of the following system commands: IL (Ignore Label), OU (Output Unit), or UL (Unlabeled Unit). In those cases, the MCP passes a parameter to CHECK_TAPE_UNIT_ASSIGNMENT that identifies which of the system commands (IL, OU, or UL) assigned the unit number. These commands cause the MCP to attempt to open the tape file on the tape unit with the unit number specified in the command.

The MCP passes CHECK_TAPE_UNIT_ASSIGNMENT various parameters, including the file variable of the tape file and the task variable of the process that declared the file.

CHECK_TAPE_UNIT_ASSIGNMENT can inspect task attributes or file attributes. Note that this procedure cannot modify file attributes.

The CHECK_TAPE_UNIT_ASSIGNMENT procedure result specifies whether the file can be opened on the requested tape unit.

If CHECK_TAPE_UNIT_ASSIGNMENT denies an open request that specifies UNITNO, then the MCP resets UNITNO and invokes the CHECK_TAPE_ASSIGNMENT procedure, to see if the open operation can proceed without a specific unit being requested.

If the procedure denies an open request that does not specify UNITNO, the process is suspended with an operator RSVP system message.

The MCP determines whether the requested tape is available on the requested unit. If the requested tape is available, the MCP continues with the tape file open operation. Otherwise, the process opening the tape file becomes suspended.

CHECK_TAPE_REOPEN

The MCP calls the CHECK_TAPE_REOPEN procedure when a process attempts to open a tape file that was previously assigned to a tape volume and was closed with retention.

Figure 5-5 illustrates the flow of information between the CHECK_TAPE_REOPEN procedure and the operating system.

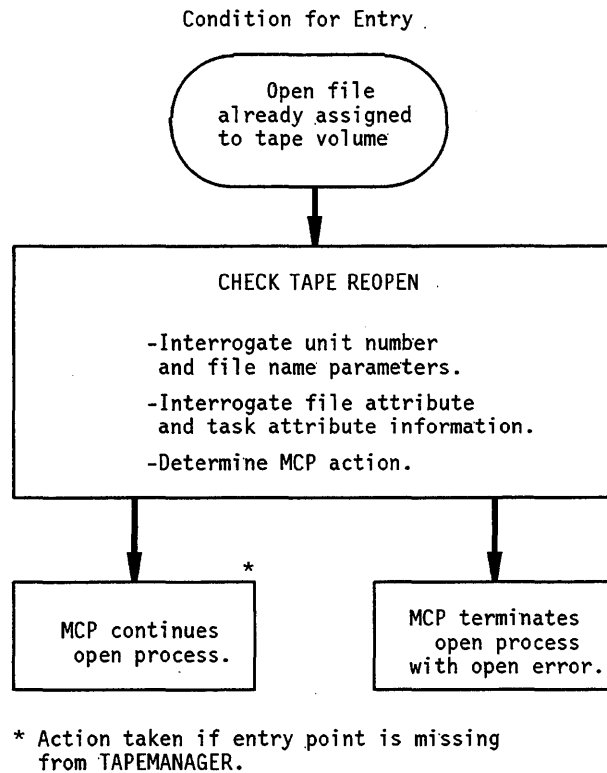


Figure 5-5. Information Flow in CHECK_TAPE_REOPEN

The MCP determines whether the tape file is already assigned to a tape volume. This is the case if the tape file was previously opened and then closed with retention.

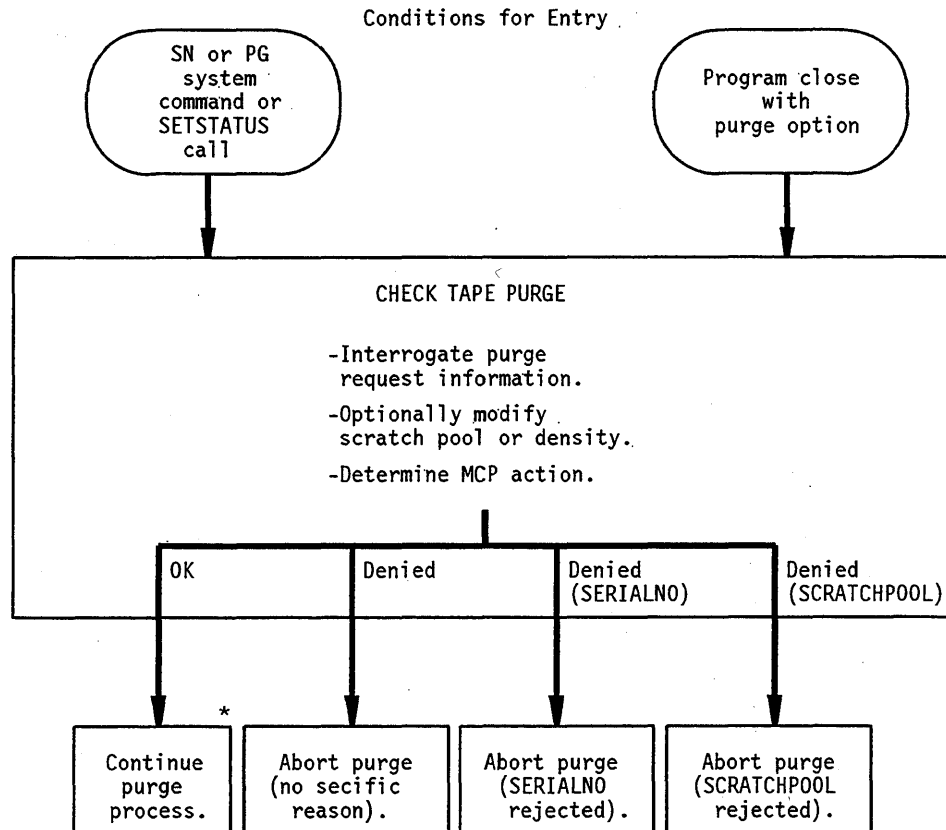
If the tape volume is already assigned, the MCP invokes the CHECK_TAPE_REOPEN procedure of the TAPEMANAGER library. CHECK_TAPE_REOPEN receives various parameters, including the file variable of the tape file and the task variable of the process that declared the file.

The CHECK_TAPE_REOPEN procedure result specifies whether the open can proceed. If CHECK_TAPE_REOPEN approves the open request, then the open request proceeds. Otherwise, the open request fails.

Purging Tape Volumes

Before it executes any purge request, the MCP calls the CHECK_TAPE_PURGE procedure in the TAPEMANAGER library.

Figure 5-6 illustrates the flow of information between the CHECK_TAPE_PURGE procedure and the operating system.



* Action taken if entry point missing from TAPEMANAGER.

Figure 5-6. Information Flow in CHECK_TAPE_PURGE

The CHECK_TAPE_PURGE procedure receives a SOURCE parameter, which specifies whether the purge request resulted from a close with purge or a PG (Purge) or SN (Serial Number) system command. The CHECK_TAPE_PURGE procedure also receives a REQUEST_AND_RESPONSE parameter containing most of a Major Type 15, Minor Type 3 (Tape Volume Purged) log entry.

The CHECK_TAPE_PURGE procedure can modify the scratch pool value, the density value, or both, in the REQUEST_AND_RESPONSE procedure. The CHECK_TAPE_PURGE procedure can also indicate, in its return value, whether the purge should be allowed to proceed.

Note: *By default, the MCP regards a tape volume as expired if the tape is mounted with a write ring and today's date is equal to, or later than, the expiration date in the tape label. The MCP treats expired tapes like scratch tapes that output files can be assigned to. In this way, a tape volume can become a scratch tape without the CHECK_TAPE_PURGE entry point in TAPEMANAGER ever being invoked.*

A tape-monitoring program can monitor this implicit creation of scratch tapes by requesting Major Type 15, Minor Type 4 (Tape Volume Expired) log entries through the REPORT_LOG_ENTRIES function. For a description of REPORT_LOG_ENTRIES, refer to the A Series System Software Support Reference Manual. For a description of the Tape Volume Expired log entries, refer to "Logging Enhancements" in Section 3, "Programming Considerations." Alternatively, you can prevent the implicit creation of scratch tapes by setting the TAPEEXPIRATION option with the SYSOPS (System Options) system command. When TAPEEXPIRATION is set, the MCP does not treat an expired tape as a scratch tape.

Handling Programmatic Calls to the TAPEMANAGER Library

Application programs that use tapes can communicate with the TAPEMANAGER library by invoking the CALL_TAPEMANAGER exported MCP procedure, which in turn calls the PROGRAM_INTERFACE procedure in the TAPEMANAGER library. Also, if a TMS includes other programs in addition to the TAPEMANAGER library, those programs could use the CALL_TAPEMANAGER procedure to communicate with TAPEMANAGER.

PROGRAM_INTERFACE receives the following two parameters:

- CALLER. This parameter indicates whether the calling program is privileged.
- REQUEST_AND_RESPONSE. This parameter is an array whose contents are defined by the TAPEMANAGER implementation.

PROGRAM_INTERFACE can return information to the caller either by modifying the REQUEST_AND_RESPONSE array, or by modifying the procedure return value.

Exported TAPEMANAGER Procedures

The MCP is designed to invoke a particular set of procedures in the TAPEMANAGER library, which must have the expected names and the expected parameter types. For an overview of the situations that cause the MCP to invoke these procedures and how they interact, refer to "Interaction with the MCP" earlier in this section.

Table 5-2 provides information about these procedures.

Table 5-2. Exported TAPEMANAGER Procedures

Procedure	Optional or Required	Purpose
CHECK_TAPE_ASSIGNMENT	Optional	Used in file open operations
CHECK_TAPE_UNIT_ASSIGNMENT	Optional	Used in file open operations
CHECK_TAPE_REOPEN	Optional	Used in file open operations
CHECK_TAPE_PURGE	Optional	Used in tape volume purge operations
PROGRAM_INTERFACE	Optional	Used by CALL_TAPEMANAGER procedure
CONTROL_INTERFACE	Required	Used by MCP to initialize library

Of these procedures, the CONTROL_INTERFACE procedure must be present for the MCP to consider the library valid. The other procedures are optional and are not invoked if they are not present.

If the PROGRAM_INTERFACE procedure is not present, and a user program invokes the CALL_TAPEMANAGER procedure, the user program proceeds normally and CALL_TAPEMANAGER returns a procedure result value of 0. Refer to "CALL_TAPEMANAGER" and "PROGRAM_INTERFACE" later in this section.

The following pages provide detailed explanations of these procedures. A sample TAPEMANAGER library coded in DCALGOL is also provided. The library includes sample code for each TAPEMANAGER procedure as well as the exported MCP procedures CALL_TAPESERVER and CALL_TAPEMANAGER, described later in this section.

CHECK_TAPE_ASSIGNMENT

The MCP calls the CHECK_TAPE_ASSIGNMENT procedure in the TAPEMANAGER library when a process attempts to open a tape file that is not assigned to a tape volume, and the UNITNO file attribute is not assigned a tape unit number.

The MCP passes parameters to CHECK_TAPE_ASSIGNMENT, including the values of those file attributes most likely to be of interest, the file variable itself, and the task variable of the process opening the file.

The parameters can specify a labeled or unlabeled tape, a backup or non-backup tape, a new or existing tape volume, and acceptance of either a nontape file or a tape file only.

The CANDIDATE_FILE is the file variable of the tape file being opened. This parameter determines which file attributes can be modified by CHECK_TAPE_ASSIGNMENT to affect the current open request.

Note that the TAPEMANAGER library should not open the CANDIDATE_FILE, because an open operation is already underway for that file. If TAPEMANAGER attempts to open the file, the open operation fails. The open result or AVAILABLE file attribute value is 234. If the opening process does not trap the open error, the system discontinues the process and displays system message MSRFIB270: "LIBRARY CANNOT OPEN THIS FILE."

CHECK_TAPE_ASSIGNMENT can inspect the values passed, and modify the file attributes of the file variable if desired. The procedure also must return an indication of whether the tape volume assignment is to be permitted or denied.

If this entry point is missing in the TAPEMANAGER library, the result is the same as if permission to open the tape file had been granted.

Compare this procedure with the CHECK_TAPE_UNIT_ASSIGNMENT and CHECK_TAPE_REOPEN procedures discussed later in this section.

Syntax

Specify the procedure heading as follows:

```
REAL PROCEDURE CHECK_TAPE_ASSIGNMENT (SOURCE,
                                     CANDIDATE_FILE,
                                     CANDIDATE_FILE_TASK,
                                     REQUEST_AND_RESPONSE);

VALUE SOURCE;
REAL SOURCE;
FILE CANDIDATE_FILE;
TASK CANDIDATE_FILE_TASK;
ARRAY REQUEST_AND_RESPONSE [0];
```

Explanation

The description of the parameters follows.

SOURCE (Real, Input Only)

The source of the request is laid out as follows:

Field	Value	Meaning
[47: 1]	0	The opening process and the declaring process are the same.
	1	The opening process is different from the declaring process.
[46: 1]	0	The request is for a labeled tape.
	1	The request is for an unlabeled tape.
[45: 1]	0	Only a tape file can satisfy the request.
	1	The request can be satisfied by a tape file or a nontape file (for example, a logical file with KIND = DONTCARE).
[44: 1]	0	The request is not for a backup tape.
	1	The request is for a backup tape.
[43:36]		This field is reserved for future use.
[7: 8]	1	The request is for a new tape volume.
	2	The request is for an existing tape volume.

CANDIDATE_FILE (File, Input/Output)

CANDIDATE_FILE is the file variable of the tape file being opened. The following rules determine which file attributes CHECK_TAPE_ASSIGNMENT can modify to affect the current open request:

- For tapes that are not library maintenance tapes, CHECK_TAPE_ASSIGNMENT can modify the same file attributes that can be modified by an FA (File Attributes) system command.
- For files assigned to the first volume of library maintenance tapes, the FA command can modify only the SCRATCHPOOL and SERIALNO file attributes. However, CHECK_TAPE_ASSIGNMENT can modify all the following file attributes: CYCLE, DENSITY, FILENAME, SAVEFACTOR, SCRATCHPOOL, SERIALNO, and VERSION. If FILENAME is modified, the new FILENAME value must end in /FILE000.

- For the second and later volumes of library maintenance tapes, both the FA command and CHECK_TAPE_ASSIGNMENT can modify only the SERIALNO file attribute.

If CHECK_TAPE_ASSIGNMENT modifies other file attribute values, the changes do not affect the open request that is currently in progress. However, if the same logical file is reused in a later open request, the file attribute assignments might affect that request.

If CHECK_TAPE_ASSIGNMENT does change any file attribute values, this fact is noted in Word 7, Field [44: 1] of the Major Type 1, Minor Type 5 (File Open) log entry. Changing attributes during the call does not result in another immediate call; only an operator FA response does that.

Note: When you assign the SERIALNO attribute, make sure to supply as a parameter the file section number found in the REQUEST_AND_RESPONSE array word [9].[15.16].

CANDIDATE_TASK (Task, Input/Output)

The task variable of the process that declared the file. CHECK_TAPE_ASSIGNMENT can interrogate or modify task attribute values. For related information, refer to the discussion of Field [47: 1] of the SOURCE parameter.

REQUEST_AND_RESPONSE (Real Array [0], Input/Output)

This parameter stores selected information about the tape file. CHECK_TAPE_ASSIGNMENT can inspect this information, and modify it if desired.

The LINK words contained in this parameter have the following format:

Field	Meaning
[39:20]	Length of item in words.
[19:20]	Index of the word in which the item starts. This index points somewhere in the variable part of the array.

In the following table, the column labeled "Input or Output" indicates whether this parameter is input to the procedure, output from the procedure, or both.

Input or Output	Word/Field	Value	Meaning
I/O	[0].[47: 8]		Length of fixed part of record in words
I/O	[0].[39:12]		Length of entire record in words
	[0].[27:28]		Field reserved for future use
I	[1]		Serial number in EBCDIC characters (binary zeroes indicate no specification)
I	[2]		Density requested for new volume
	[2].[47: 1]	0	Density is not specified.

continued

TAPEMANAGER Library Implementation (cont.)

continued

Input or Output	Word/Field	Value	Meaning
		1	Density is specified.
	[2].[46:39]		Field reserved for future use
	[2].[7: 8]		Density code (same range of values as DENSITY file attribute)
I	[3]-[5]		Requested scratch pool (in EBCDIC characters) for new volume, or zeroes. (First byte is binary count of characters in the scratch pool name; the count byte is not counted.)
I	[6]		LINK to the standard form name of the file (with the usercode removed).
I	[7]		Volume/catalog information
	[7].[47: 2]		Volume status for new volume
		0	Do not care
		1	Tape must be nonvolumed.
		2	Tape must be volumed.
	[7].[45:46]		Field reserved for future use
I	[8]		Genealogy (for existing volume)
	[8].[47: 1]	0	The permanent file genealogy can differ from the logical file genealogy.
		1	The permanent file genealogy must match the logical file genealogy.
	[8].[46: 7]		Field reserved for future use
	[8].[39:20]		CYCLE. Default value is 1.
	[8].[19:20]		VERSION. Default value is 0.
I	[9]		Volume Numbering
	[9].[47:32]		Field reserved for future use
	[9].[15:16]		FILESECTION (REEL)
	[10]-[end]		Variable length data pointed to by the LINKs in this record

Procedure Value (Real, Output Only)

The following is the format of the value returned by the procedure:

Field	Value	Meaning
[47: 1]	0	Operator intervention is required for any tape subsequently mounted by the TAPESERVER library to be assigned to a request for an existing unlabeled tape.
	1	A tape subsequently mounted by the TAPESERVER library should be assigned to a request for an existing unlabeled tape, without operator intervention. This value is overridden, and operator intervention is required, if the S1RESTRICTIONS or S2RESTRICTIONS system security option is set or if the CLASS system security option has a value of S1 or S2. System security options are controlled through the SECOPT (Security Options) system command.
[46:38]		Field reserved for future use
[7: 8]	0	Continue search.
	1	Return open error and abandon file search. The open result or AVAILABLE file attribute value is 236. If the opening process does not trap the open error, the system discontinues the process and displays system message MSRFIB271: "ASSIGNMENT STOPPED BY TAPEMANAGER LIBRARY."
	2	Do not include tape units. The file search continues, but tape files are not considered. This value can be a useful response when the MCP invokes CHECK_TAPE_ASSIGNMENT for a file whose KIND file attribute has a value of DONTCARE.
	3	Return the same value as if an NF (No File) system command were entered (abandon file search).
	4	Return the same value as if an OF (Optional File) system command were entered (abandon file search). If the OPTIONAL file attribute does not have the value TRUE, an open error results. The open result and the AVAILABLE file attribute value are each 44 (ERRORINOPENRSLT). If the opening process does not trap the error, the system suspends the process and displays system message MSRFIB16: "ILLEGAL OPTIONAL FILE."
	5	Return the same value as if an FR (Final Reel) system command were entered (abandon file search).

CHECK_TAPE_PURGE

The CHECK_TAPE_PURGE procedure in the TAPEMANAGER library is called by the MCP when an attempt is made to purge a tape. Actions that result in a tape purge are the PG (Purge) system command, the SN (Serial Number) system command, and tape file close operations with the PURGE option specified.

The MCP passes CHECK_TAPE_PURGE information, including the source of the purge request, the unit number of the tape unit, the serial number, and the scratch pool to be assigned.

CHECK_TAPE_PURGE can inspect the values passed and modify the scratch pool or the density. The procedure also must return an indication of whether the purge request is to be allowed or denied.

If this entry point is missing from the TAPEMANAGER library, the result is the same as if the purge had been checked and allowed.

Syntax

Specify the procedure heading as follows:

```
BOOLEAN PROCEDURE CHECK_TAPE_PURGE (SOURCE,  
                                     REQUEST_AND_RESPONSE);  
    VALUE SOURCE;  
    REAL SOURCE;  
    ARRAY REQUEST_AND_RESPONSE [0];
```

Explanation

The description of the parameters follows.

SOURCE (Real, Input Only)

The source of the request is laid out as follows:

Field	Value	Meaning
[47:40]		Field reserved for future use
[7: 8]	1	The purge is because of a PG (Purge) or SN (Serial Number) system command, or a SETSTATUS call corresponding to one of these commands.
	2	The tape was closed with the PURGE option.

REQUEST_AND_RESPONSE (Real Array [0], Input/Output)

This array has the same layout as a Major Type 15, Minor Type 3 (Tape Volume Purged) log entry, except that Words 0 through 2 are not valid. For a description of the log entry format, refer to "Major Type 15-Minor Types 1 through 5, 12, and 13: Volume Status Entry (LOGMAJVOL)" in Section 3, "Programming Considerations."

The CHECK_TAPE_PURGE procedure can change the value of the scratch pool or the density. If CHECK_TAPE_PURGE changes the scratch pool, CHECK_TAPE_PURGE should also set bit [2: 1] in the procedure value. If a TAPESERVER library is in use, it can override such changes to the scratch pool. Further, if CHECK_TAPE_PURGE requests a density that is not supported by the tape unit, the MCP proceeds with the purge action but does not assign the requested density.

Procedure Value (Boolean, Output Only)

The contents of the procedure value are as follows:

Field	Value	Meaning
[47: 8]		Reason for denying attempt
	0	No reason given. The MCP displays system message MSRDPC177: "PURGE REQUEST REJECTED BY TAPEMANAGER."
	1	Rejected SERIALNO value. The MCP displays system message MSRDPC181: "SERIAL NUMBER REJECTED BY TAPEMANAGER."
	2	Rejected SCRATCHPOOL value. The MCP displays system message MSRDPC183: "SCRATCHPOOL REJECTED BY TAPEMANAGER."
[39:37]		Field reserved for future use
[2: 1]	0	CHECK_TAPE_PURGE did not modify the scratch pool value.
	1	CHECK_TAPE_PURGE modified the scratch pool value.
[1: 1]		Field reserved for future use
[0: 1]	0	The purge is allowed.
	1	The attempt is denied. In this case, no Major Type 15, Minor Type 3 (Tape Volume Purged) log entry is issued.

CHECK_TAPE_REOPEN

The MCP calls the CHECK_TAPE_REOPEN procedure in the TAPEMANAGER library when a process attempts to open a tape file that was previously assigned to a tape volume and was closed with retention.

The MCP passes parameters to CHECK_TAPE_REOPEN, including the file variable, the file name, and the unit number of the tape unit.

CHECK_TAPE_REOPEN must return a result indicating whether the file open is to be allowed or denied.

If this entry point is missing from the TAPEMANAGER library, the result is the same as if the tape reopen request were accepted.

Compare this procedure with the CHECK_TAPE_ASSIGNMENT and CHECK_TAPE_UNIT_ASSIGNMENT procedures discussed elsewhere in this section.

Syntax

Specify the procedure heading as follows:

```
BOOLEAN PROCEDURE CHECK_TAPE_REOPEN (RETAINED_UNIT,  
                                     RETAINED_FILE,  
                                     RETAINED_FILE_TASK,  
                                     NEW_FILENAME);  
  
VALUE RETAINED_UNIT;  
REAL RETAINED_UNIT;  
FILE RETAINED_FILE;  
TASK RETAINED_FILE_TASK;  
ARRAY NEW_FILENAME [0];
```

Explanation

The description of the parameters follows.

RETAINED_UNIT (Real, Input Only)

The unit number (external device number) of the tape unit that is to be reused.

RETAINED_FILE (File, Input Only)

The file variable of the tape file. Any attempt to modify a file attribute results in an attribute error.

Note that the TAPEMANAGER library should not open the RETAINED_FILE, because an open action is already underway for that file. If TAPEMANAGER attempts to open the file, the open will fail. The open result or AVAILABLE file attribute value is 234. If the opening process does not trap the open error, the system discontinues the

process and displays system message MSRFIB270: "LIBRARY CANNOT OPEN THIS FILE."

RETAINED_FILE_TASK (Task, Input Only)

The task variable of the process that declared the file. CHECK_TAPE_REOPEN can interrogate or modify task attribute values. Refer to the discussion of Field [47: 1] of the SOURCE parameter of the CHECK_TAPE_ASSIGNMENT procedure.

NEW_FILENAME (Real Array [0], Input Only)

The file name of the file being opened. By contrast, the FILENAME file attribute reflects the current label of the tape unit. The current label is displayed in the output from an OL (Display Label and Paths) system command. The current label might reflect the FILENAME of a file that was previously opened on the same tape volume.

Procedure Value (Boolean, Output Only)

The contents of the procedure value are as follows:

Field	Value	Meaning
[47:47]		Field reserved for future use
[0: 1]	0	Directs the MCP to continue processing the file open.
	1	Directs the MCP to terminate the file open with an open error. The open result or AVAILABLE file attribute value is 230. If the opening process does not trap the open error, the system discontinues the process and displays system message MSRFIB268: "REOPEN STOPPED BY TAPEMANAGER."

CHECK_TAPE_UNIT_ASSIGNMENT

The MCP calls the CHECK_TAPE_UNIT_ASSIGNMENT procedure in the TAPEMANAGER library when a process attempts to open a tape file that has not yet been assigned to a tape volume, and the UNITNO file attribute has been assigned a tape unit number. CHECK_TAPE_UNIT_ASSIGNMENT is called regardless of whether UNITNO was assigned by the opening process or by a system command such as IL (Ignore Label), OU (Output Unit), or UL (Unlabeled).

The MCP passes parameters to CHECK_TAPE_UNIT_ASSIGNMENT, including the specified unit number, the file variable, and the task variable of the declaring process, along with other information. Some of the options specify a labeled or unlabeled tape, a backup or non-backup tape, an operator entry of a parameter, a tape file only or nontape file, and whether the process opening the file is the same as the process declaring the file.

The CANDIDATE_FILE is the file variable of the tape file being opened. Note that the TAPEMANAGER library should not open the CANDIDATE_FILE, since an open operation is already underway for that file. If TAPEMANAGER attempts to open the file, the open operation fails. The open result or AVAILABLE file attribute value is 234. If the opening process does not trap the open error, the system discontinues the process and displays system message MSRFIB270: "LIBRARY CANNOT OPEN THIS FILE."

CHECK_TAPE_UNIT_ASSIGNMENT can inspect the values passed, and modify the file attributes of the file variable if desired. The procedure must return an indication of whether the tape volume assignment is to be allowed or denied. If the assignment is denied, one of the following actions results:

- If the requested UNITNO was assigned programmatically to the UNITNO file attribute, the MCP invokes the CHECK_TAPE_ASSIGNMENT procedure to determine whether TAPEMANAGER allows the file open on an unspecified unit.
- If the requested UNITNO was assigned by an IL, OU, or UL system command, the MCP suspends the opening process.

The procedure can also indicate whether an unlabeled tape request can be handled by the TAPESERVER library without an operator response of a UL system command to a request for an unlabeled tape.

If this entry point is missing from the TAPEMANAGER library, the result is the same as if the tape unit request were allowed.

Compare this procedure with the CHECK_TAPE_ASSIGNMENT and CHECK_TAPE_REOPEN procedures discussed earlier in this section.

Syntax

Specify the procedure heading as follows:

```

BOOLEAN PROCEDURE CHECK_TAPE_UNIT_ASSIGNMENT (SOURCE,
                                                CANDIDATE_FILE,
                                                CANDIDATE_FILE_TASK,
                                                CANDIDATE_UNIT);

VALUE SOURCE,
        CANDIDATE_UNIT;
REAL SOURCE,
        CANDIDATE_UNIT;
FILE CANDIDATE_FILE;
TASK CANDIDATE_FILE_TASK;
    
```

Explanation

The description of the parameters follows.

SOURCE (Real, Input Only)

The source of the request. This array is laid out as follows:

Field	Value	Meaning
[47: 1]	0	Indicates whether the process opening the file is the same as the process declaring the file. The task variable of the declaring process is passed in the CANDIDATE_TASK parameter. The task variable of the opening process can be accessed by way of the MYSELF predeclared task variable.
	1	The opening process and the declaring process are the same.
	1	The opening process is different from the declaring process.
[46: 1]	0	The request is for a labeled tape.
	1	The request is for an unlabeled tape.
[45: 1]	0	Only a tape file can satisfy the request.
	1	The request can be satisfied by a tape file or a nontape file (for example, a logical file with KIND = DONTCARE).
[44: 1]	0	The request is not for a backup tape.
	1	The request is for a backup tape.
[44:37]		Field reserved for future use
[7: 8]	3	The UNITNO file attribute was specified or an operator entered the OU (Output Unit) system command.

continued

continued

Field	Value	Meaning
	4	An operator entered on IL (Ignore Label) system command.
	5	An operator entered a UL (Unlabeled) system command.

CANDIDATE_FILE (File, Input/Output)

CANDIDATE_FILE is the file variable of the tape file being opened. CHECK_TAPE_UNIT_ASSIGNMENT can interrogate the file attribute values, but cannot modify them. An attempt to modify file attribute values results in an attribute error. The ATERR file attribute is set to 177, and the system displays system error message MSRATTERR177: "ERROR: ATTRIBUTE MODIFICATION PROHIBITED FOR THIS FILE PARAMETER."

CANDIDATE_TASK (Task, Input Only)

The task variable of the process that declared the file. CHECK_TAPE_UNIT_ASSIGNMENT can interrogate task attribute values. For related information, refer to the discussion of Field [47: 1] of the SOURCE parameter.

CANDIDATE_UNIT (Real, Input Only)

The unit number (external device number) of the requested unit in binary.

Procedure Value (Real, Output Only)

The procedure value is divided into the following fields:

Field	Value	Meaning
[47:47]		Field reserved for future use.
[0: 1]	0	Assignment to this tape unit is approved.
	1	Assignment to this tape unit is rejected.
		If CHECK_TAPE_UNIT_ASSIGNMENT is invoked because the UNITNO file attribute is specified, the MCP resets the UNITNO file attribute to a null value and invokes the CHECK_TAPE_ASSIGNMENT procedure to reevaluate the open request.
		If CHECK_TAPE_UNIT_ASSIGNMENT is invoked because the operator enters an IL, OU, or UL system command, the opening process remains suspended with a condition that does not request a specific tape unit.

CONTROL_INTERFACE

The TAPEMANAGER_INITIALIZATION independent runner calls the CONTROL_INTERFACE procedure in the TAPEMANAGER library when the library is being initialized. Refer to "Initiating the TAPEMANAGER Library" later in this section.

The TAPEMANAGER_TERMINATION independent runner calls the CONTROL_INTERFACE procedure when the library is being terminated by the SEND TM QUIT form of the SEND (Send Message) system command. Refer to "Terminating the TAPEMANAGER Library" later in this section. If the SEND TM QUIT <text> command is used, the text is passed to the REQUEST parameter of the procedure.

The MCP generates the response to the SEND TM STATUS command. It is unnecessary to include any code in the TAPEMANAGER library to generate this response.

CONTROL_INTERFACE can be designed to return a Boolean procedure value to cause the initiation or termination request to be terminated.

If this entry point is missing from the TAPEMANAGER library, the library cannot be initialized.

Syntax

Specify the procedure heading as follows:

```

BOOLEAN PROCEDURE CONTROL_INTERFACE (SOURCE,
                                     REQUEST);
    VALUE SOURCE;
    REAL SOURCE;
    ARRAY REQUEST [0];
    
```

Explanation

The description of the parameters follows.

SOURCE (Real, Input Only)

The type of request. The following are the possible values:

Value	Meaning
1	Start library.
2	Stop library.

REQUEST (Real Array [0], Input Only)

This array contains any text specified by the operator in a *SEND TM QUIT <text>* system command. The format of the array is as follows:

Word	Meaning
0	The length of the input text, not counting this word. Although a full word is used to store the length, the length itself is expressed in units of bytes.
1-end	The input text. The meaning of this text is determined by the TAPEMANAGER implementation.

Procedure Value (Boolean, Output Only)

The `CONTROL_INTERFACE` procedure can reject the start or stop request by returning a value of `TRUE` in the procedure value.

If `CONTROL_INTERFACE` rejects a start request, the MCP terminates the TAPEMANAGER library and displays system message `MSRMISC214`:
“TAPEMANAGER FAILED IN INITIALIZATION PROCEDURE.”

If `CONTROL_INTERFACE` rejects a stop request, the MCP terminates the TAPEMANAGER library anyway, but displays system message `MSRMISC215`:
“TAPEMANAGER FAILED IN TERMINATION PROCEDURE.”

PROGRAM_INTERFACE

The MCP calls the PROGRAM_INTERFACE procedure in the TAPEMANAGER library when a user process invokes the exported MCP procedure CALL_TAPEMANAGER. The PROGRAM_INTERFACE and CALL_TAPEMANAGER procedures are designed to make it possible for user programs to communicate with the library without linking to it directly. The use of CALL_TAPEMANAGER as a jacket procedure is necessary because only the MCP is allowed to link to the TAPEMANAGER library. For further information, refer to "CALL_TAPEMANAGER" later in this section.

The PROGRAM_INTERFACE procedure receives an indication of whether the user program is privileged, and an array containing input from the user program. The format and meaning of this information is defined by the TAPEMANAGER library implementation.

The PROGRAM_INTERFACE procedure can modify the input array to record its response. PROGRAM_INTERFACE can also return an integer procedure value whose meaning is defined by the implementation. Values less than or equal to 0 are reserved for use by the MCP to record CALL_TAPEMANAGER errors.

If the PROGRAM_INTERFACE procedure is not present in the TAPEMANAGER library, and a user program invokes the CALL_TAPEMANAGER procedure, the user program proceeds normally and CALL_TAPEMANAGER returns a procedure result value of 0.

Syntax

Specify the procedure heading as follows:

```
REAL PROCEDURE PROGRAM_INTERFACE (CALLER,
                                  REQUEST_AND_RESPONSE);
VALUE CALLER;
REAL CALLER;
ARRAY REQUEST_AND_RESPONSE [*];
```

Explanation

The description of the parameters follows.

CALLER (Real, Input Only)

The contents of the CALLER parameter are as follows:

Field	Value	Meaning
[47: 1]	1	The caller is privileged.
[46:47]		Field reserved for future use

REQUEST_AND_RESPONSE (Real Array [*, Input/Output])

Defined by the library implementation

Procedure Value (Real, Output Only)

This value is copied to the `CALL_TAPEMANAGER` procedure result, where it can be read by the user program. Values greater than 0 have meanings defined by the library implementation.

A value of 0 is ambiguous to the user program, because `CALL_TAPEMANAGER` also returns a value of 0 if `PROGRAM_INTERFACE` is not present in the `TAPEMANAGER` library. Therefore, `PROGRAM_INTERFACE` should not use a 0 value to convey information to the user program.

If the value is less than 0, the constant value -2, indicating an incorrect value, is returned to the caller of `CALL_TAPEMANAGER`.

Sample TAPEMANAGER Library

Two sample programs, both written in DCALGOL, are provided in this section to demonstrate some of the features available with the TAPEMANAGER interface. They are not intended to be used as an actual TMS.

The first program comprises a TAPEMANAGER library, and includes sample code for each supported TAPEMANAGER procedure as well as an example of a call to the exported MCP procedure CALL TAPEMANAGER. The first program refers to another program called CT/SET/VOLFAM/SERIALNO, which is the second sample program and is an example of a TMS program that uses the interface to the TAPEMANAGER library.

```

$ SET INSTALLATION 1
BEGIN
  LABEL AWAY;
  LIBRARY MCPSUPPORT (LIBACCESS=BYFUNCTION,FUNCTIONNAME="MCPSUPPORT");
  ARRAY VOLFAM_SERIALNOS [0:9,0:9];

  REAL PROCEDURE CALL_TAPESERVER (UNLINKED_ACTION, A);
    VALUE UNLINKED_ACTION;
    REAL UNLINKED_ACTION;
    ARRAY A[*];
    LIBRARY MCPSUPPORT;

  PROCEDURE CHECK_HISTORY;
    BEGIN
      REAL TYPE, CAUSE, REASON;
      TYPE := MYSELF.HISTORYTYPE;
      IF TYPE NEQ VALUE(NORMALV) THEN
        BEGIN
          CAUSE := MYSELF.HISTORYCAUSE;
          REASON := MYSELF.HISTORYREASON;
          DISPLAY ("HISTORY: " !! STRING (TYPE,*) !! ", " !!
            STRING (CAUSE,*) !! ", " !! STRING(REASON,*));
        END;
    END CHECK_HISTORY;

  REAL PROCEDURE CHECK_TAPE_ASSIGNMENT (SOURCE, F, DECLARER, A);
    VALUE SOURCE;
    REAL SOURCE;
    TASK DECLARER;
    ARRAY A[0];
    FILE F;
    BEGIN
      ARRAY FN[0:43]; % FILENAME FROM FILE
      ARRAY AFN[0:43]; % FILENAME FROM ARRAY
      ARRAY MSG[0:30];
      REAL FILE_SECTION;
      POINTER P, AFNP;
      EBCDIC ARRAY EA[0:71];
    END
  
```

TAPEMANAGER Library Implementation (cont.)

```
REAL RETURNVAL;
LABEL BACK;
ON ANYFAULT,
  BEGIN
    DISPLAY ("PDUMP");
    PROGRAMDUMP;
    GO BACK;
  END;

%
%Display the parameters of the request
%
REPLACE POINTER(FN) BY F.TITLE;
REPLACE P:POINTER(MSG) BY "GET ";
CASE A[7].[47:2] OF
  BEGIN
    Ø: ;
    1: REPLACE P:P BY "NON-VOLUMED ";
    2: REPLACE P:P BY "VOLUMED ";
  END;
IF BOOLEAN(SOURCE.[46:1]) THEN
  REPLACE P:P BY "UNLABELED ";
CASE SOURCE.[7:8] OF
  BEGIN
    (1): REPLACE P:P BY "OUTPUT ";
    (2): REPLACE P:P BY "INPUT ";
    ELSE: REPLACE P:P BY
      "(SOURCE=",SOURCE.[7:8] FOR * DIGITS,") ";
  END CASES;
IF BOOLEAN(SOURCE.[44:1]) THEN
  REPLACE P:P BY "BACKUP ";
IF BOOLEAN(SOURCE.[45:1]) THEN
  REPLACE P:P BY "FILE "
ELSE
  REPLACE P:P BY "TAPE ";
AFNP := POINTER (AFN);
STANDARDTODISPLAY(POINTER(A[A[6]]),AFNP);
REPLACE P:P BY POINTER(AFN) FOR 3Ø UNTIL=".";
IF A[1] ISNT Ø THEN
  REPLACE P:P BY " [" ,POINTER(A[1]) FOR 6, "]" ;
FILE_SECTION := A[9].[15:16];
REPLACE P:P BY " #", FILE_SECTION FOR * DIGITS;
IF SOURCE.[7:8] = 2 THEN
  IF BOOLEAN(A[8].[47:1]) THEN % EXACT
    REPLACE P:P BY " ",A[8].[39:2Ø] FOR * DIGITS, % CYCLE
      ":",A[8].[19:2Ø] FOR * DIGITS % VERSION
  ELSE
    REPLACE P:P BY " (" ,A[8].[39:2Ø] FOR * DIGITS, % CYCLE
      ":",A[8].[19:2Ø] FOR * DIGITS,% VERSION
      ")";
IF BOOLEAN(A[2].[47:1]) THEN
  CASE A[2].[7:8] OF
    BEGIN
```

```

0: REPLACE P:P BY " (800)";
1: REPLACE P:P BY " (556)";
2: REPLACE P:P BY " (200)";
3: REPLACE P:P BY " (1600)";
4: REPLACE P:P BY " (6250)";
5: REPLACE P:P BY " (38000)";
6: REPLACE P:P BY " (1250)";
END;
IF A[3].[47:8] GTR 0 THEN
  REPLACE P:P BY " POOL=", POINTER(A[3])+1 FOR A[3].[47:8];
REPLACE P:P BY 48"00";
DISPLAY(MSG);

%The remainder of this procedure is a set of examples
%that perform different actions based on the file name.

IF POINTER(FN) = "GET_VOLFAM_" THEN
  BEGIN
  %Change SERIALNO to that provided by CT/SET/VOLFAM/SERIALNO
  TRUTHSET DIGIT_CHARACTERS ("0123456789");
  REAL FAM, SNTOUSE;
  IF POINTER(FN)+11 IN DIGIT_CHARACTERS FOR 1 THEN
    BEGIN
    FAM:=INTEGER(POINTER(FN)+11,1);
    IF SNTOUSE := VOLFAM_SERIALNOS [FAM,FILE_SECTION]
    ISNT 0 THEN
      BEGIN
      F(FILE_SECTION).SERIALNO := SNTOUSE;
      REPLACE P:POINTER(MSG) BY
        "SERIALNO (", FILE_SECTION FOR * DIGITS, ")",
        " SET TO ", SNTOUSE, 48"00";
      DISPLAY(MSG);
      END;
    IF FAM = 9 THEN
      BEGIN
      %Allow unlabeled assignment to tape mounted
      %by TAPESERVER library
      RETURNVAL := 0 & 1 [47:1]; % OK TO UL
      DISPLAY ("WILL UL SERVER MOUNT");
      END;
    END;
  END;

IF POINTER(FN) = "ORACLE." THEN
  BEGIN
  %Display information about the tasks involved with the file
  DEFINE TELL_ABOUT (TSK,ID) =
    BEGIN
    REPLACE P:POINTER(MSG) BY ID, ": ";
    REPLACE P:P BY TSK.STACKNO FOR 4 DIGITS;
    REPLACE P:P BY " (", TSK.USERCODE;
    REPLACE P:P-1 BY ") ", TSK.NAME;
  
```

TAPEMANAGER Library Implementation (cont.)

```
        REPLACE P:P-1 BY 48"00";
        DISPLAY (MSG);
        END #;
    TELL_ABOUT (MYSELF,"MYSELF");
    TELL_ABOUT (MYJOB,"MYJOB");
    IF BOOLEAN(SOURCE.[47:1]) THEN
        TELL_ABOUT (DECLARER,"DECLARER");
    END;

IF POINTER(FN) = "AUTOUNLOAD_" THEN
    BEGIN
        %Set the AUTOUNLOAD attribute
        LABEL SKIPMSG;
        IF POINTER(FN)+11 ="ON" THEN
            F.AUTOUNLOAD := VALUE(ON)
        ELSE
            IF POINTER(FN)+11 ="OF" THEN
                F.AUTOUNLOAD := VALUE(OFF)
            ELSE
                IF POINTER(FN)+11 ="DO" THEN
                    F.AUTOUNLOAD := VALUE(DONTCARE)
                ELSE
                    GO SKIPMSG;
                DISPLAY ("AUTOUNLOAD CHANGED");
            SKIPMSG:
            END;
        IF POINTER(FN) = "WILL" THEN
            BEGIN
                DISPLAY("CHANGING FILE");
                IF POINTER(FN)+5 ="Y." THEN
                    REPLACE POINTER(FN) BY "NILLY.";
                IF POINTER(FN)+5 ="Y/FILE000." THEN
                    REPLACE POINTER(FN) BY "NILLY/FILE000.";
                IF POINTER(FN)+5 ="YNILLY/FILE000." THEN
                    REPLACE POINTER(FN) BY "WILLYNILLY.";
                F.DIOFILESTRUCTURE := VALUE (ALIGNED180);
                F.SAVEFACTOR := 123;
                F.VERSION := 2;
                F.CYCLE := 3;
                F.DENSITY := VALUE(BPI1600);
                REPLACE F.SCRATCHPOOL BY "DUMMY.";
                REPLACE F.FILENAME BY POINTER(FN);
            END;
        IF POINTER(FN) = "FILM." THEN
            BEGIN
                %Prevent assignment to a tape unit
                DISPLAY ("TAPE WILL BE IGNORED");
                RETURNVAL := 2;
            END;
        IF POINTER(FN) = "OPEN_F." THEN
            BEGIN
                %Attempt to open the file passed in
```

```

OPEN (F);                                % ERROR TEST
END;

%Test the various return values
IF POINTER(FN) = "OK_TO_UL" THEN
    RETURNVAL := 0 & 1 [47:1]
ELSE IF POINTER(FN) = "CAN_NOT_OPEN" THEN
    RETURNVAL := 1
ELSE IF POINTER (FN) = "OF_ME" THEN
    RETURNVAL := 4
ELSE IF POINTER (FN) = "NF_ME" THEN
    RETURNVAL := 3
ELSE IF POINTER (FN) = "FR_ME" AND F.FILESECTION = 2 THEN
    RETURNVAL := 5;

BACK:
CHECK_HISTORY;
DISPLAY ("RETURNING NOW, [47:1] = " !! STRING(RETURNVAL.[47:1],*)
        !! ", [7:8] = " !! STRING(RETURNVAL.[ 7:8],*));
CHECK_TAPE_ASSIGNMENT := RETURNVAL;
END CHECK_TAPE_ASSIGNMENT;

BOOLEAN PROCEDURE CHECK_TAPE_REOPEN(U,F,TSK,NEW_FN);
VALUE U;
REAL U;
FILE F;
TASK TSK;          % DECLARER TASK
ARRAY NEW_FN [0]; % NEW FILE NAME IN STANDARD FORM
BEGIN
ARRAY FN[0:9];
ARRAY NFN[0:9];   % NEW FILE NAME IN DISPLAY FORM
POINTER NFNPN;
POINTER P;
REAL TSKMIX;
ARRAY MSG[0:20];
REPLACE POINTER(FN) BY F.FILENAME;
NFNPN := POINTER(NFN);
ON ANYFAULT,
    BEGIN
        DISPLAY ("FAULT");
        PROGRAMDUMP(ARRAYS);
    END;
STANDARDTODISPLAY (POINTER(NEW_FN),NFNPN);
IF FN = "CAN_NOT_REOPEN" THEN
    BEGIN
        %Deny the reopen
        DISPLAY ("REOPEN DENIED");
        CHECK_TAPE_REOPEN := TRUE;
    END
ELSE
    BEGIN
        %Display the parameters of the request and grant permission
        IF POINTER(NFN) = "." FOR 1 THEN

```

TAPEMANAGER Library Implementation (cont.)

```
        PROGRAMDUMP (ARRAYS,FILES);
        REPLACE P:POINTER(MSG) BY "REOPEN OF ",
            POINTER(FN) UNTIL = ".",
            " ON MT", U FOR * DIGITS,
            " AS ",
            POINTER(NFN) UNTIL = ".";

        TSKMIX := TSK.STACKNO;
        IF TSKMIX NEQ MYSELF.STACKNO THEN
            REPLACE P:P BY " FOR MIX ", TSKMIX FOR * DIGITS;
        REPLACE P:P BY " OK"48"00";
        DISPLAY (MSG);
        END;
    IF FN = "TRY_TO_CHANGE" THEN
        BEGIN
            %Attempt to modify a file attribute
            F.SAVEFACTOR := 123;                                %ERROR TEST
        END CHECK_TAPE_REOPEN;

BOOLEAN PROCEDURE CHECK_TAPE_UNIT_ASSIGNMENT (SOURCE,F,DECLARER,U);
    VALUE SOURCE,U;
    REAL SOURCE,U;
    FILE F;
    TASK DECLARER;
    BEGIN
        EBCDIC ARRAY EA [0:71];
        ARRAY FN [0:42];
        POINTER P;
        %
        %Display the parameters of the request and
        %request permission to continue
        %
        REPLACE POINTER(FN) BY F.TITLE;
        REPLACE P:EA BY "OK TO PUT ";
        IF BOOLEAN(SOURCE.[46:1]) THEN
            REPLACE P:P BY "UNLABELED ";
        IF BOOLEAN(SOURCE.[44:1]) THEN
            REPLACE P:P BY "BACKUP ";
        IF BOOLEAN(SOURCE.[45:1]) THEN
            REPLACE P:P BY "FILE "
        ELSE
            REPLACE P:P BY "TAPE ";
        REPLACE P:P BY POINTER(FN) FOR 30 UNTIL=".", " ";
        IF BOOLEAN(SOURCE.[47:1]) THEN
            REPLACE P:P BY "FOR TASK ",DECLARER.STACKNO FOR * DIGITS, " ";
        REPLACE P:P BY "ON ";
        CASE SOURCE.[7:8] OF
            BEGIN
                3:  IF U NEQ F.UNITNO THEN
                    REPLACE P:P BY "OU-ED ";
                4:  REPLACE P:P BY "IL-ED ";
                5:  REPLACE P:P BY "UL-ED ";
                ELSE:REPLACE P:P BY
```

```

        "(SOURCE=",SOURCE.[7:8] FOR * DIGITS,") ";
    END;
    REPLACE P:P BY "UNIT ",U FOR * DIGITS, "?48"00";
    %Request action from operator
    ACCEPT(EA);
    IF EA = "N" THEN
        CHECK_TAPE_UNIT_ASSIGNMENT := TRUE;
    END;

BOOLEAN PROCEDURE CONTROL_INTERFACE (WHAT,A);
    VALUE WHAT;
    REAL WHAT;
    ARRAY A[0];
    BEGIN
        ARRAY SHOW_MESSAGE [0:7];
        ON ANYFAULT,
            BEGIN
                DISPLAY ("FAULT");
                PROGRAMDUMP (ARRAYS);
            END;
        DISPLAY ("TAPEMGR CONTROL " !! STRING (WHAT,*) !! " RECEIVED");
        REPLACE POINTER (SHOW_MESSAGE) BY "A[0] = #",
            POINTER (A[0],4) FOR 12
            WITH HEXTOEBCDIC;
        DISPLAY (POINTER (SHOW_MESSAGE));
        IF A[0] > 0 THEN
            BEGIN
                %Handle optional text
                REPLACE POINTER (SHOW_MESSAGE) BY "TEXT: ",
                    POINTER (A[1])
                    FOR MIN (41,A[0]),
                    48"00";
                DISPLAY (POINTER (SHOW_MESSAGE));
                IF A[0] = 5 THEN
                    IF POINTER (A[1]) = "FAULT" THEN
                        BEGIN
                            INTEGER I;
                            DISPLAY ("WILL CAUSE FAULT");           % ERROR TEST
                            I:= I/I;
                            END;
                        END;
                    END;
                END CONTROL_INTERFACE;

REAL PROCEDURE PROGRAM_INTERFACE (CALLER,A);
    VALUE CALLER;
    REAL CALLER;
    ARRAY A[*];
    BEGIN
        DEFINE PRIV_CALLER = BOOLEAN (CALLER.[47:1])#;
        REAL SR;
        ARRAY MSG[0:14];
        POINTER P;

```


TAPEMANAGER Library Implementation (cont.)

```
LABEL AWAY;

IF A[0] = "VOLFAM" THEN % FOR CT/SET/VOLFAM/SERIALNO
  BEGIN
    %Update entry in VOLFAM_SERIALNOS array
    IF A[3]="000000" THEN
      A[3]:=0;
      VOLFAM_SERIALNOS [A[1],A[2]]:=A[3];
      GO AWAY;
    END;
  %
  %Display the parameters of the request
  %
  P:= POINTER(MSG);
  IF PRIV_CALLER THEN
    REPLACE P:P BY "PRIVILEGED ";
  REPLACE P:P BY "PROGRAM SAYS: ";
  REPLACE P:P BY POINTER(A) FOR 30 UNTIL = 48"00", 48"00";
  DISPLAY (POINTER(MSG));
  IF P:POINTER(A) = "SERVER " THEN
    BEGIN
      %Pass remainder of message to TAPESERVER library
      ARRAY SVRMSG [0:14];
      REPLACE POINTER(SVRMSG) BY P FOR 82 UNTIL = 48"00",48"00";
      DISPLAY ("CALLING SERVER");
      SR:=CALL_TAPESERVER(0,SVRMSG);
      END;
    %Return a result to the caller
    PROGRAM_INTERFACE := SR * 100 + 3;
  AWAY:
    END PROGRAM_INTERFACE;

BOOLEAN PROCEDURE CHECK_TAPE_PURGE (SOURCE,A);
  VALUE SOURCE;
  REAL SOURCE;
  ARRAY A[0];
  BEGIN
    DEFINE OLDSN = A[ 6] #,
      NEWSN = A[17] #,
      NEWDEN= A[18].[31:9] #,
      POOL = A[22] #; % 3 WORDS LONG
    POINTER P;
    ARRAY MSG[0:11];
    REAL REASON;
    %
    %Display the parameters of the request
    %
    P:=POINTER(MSG);
    CASE SOURCE.[7:8] OF
      BEGIN
        1: ; % SETSTATUS CALL
        2: REPLACE P:P BY "CLOSEPG, ";
```

```

ELSE: REPLACE P:P BY "SOURCE=", SOURCE.[7:8] FOR * DIGITS;
END;
IF OLDSN IS Ø THEN
    REPLACE P:P BY "NO OLD SN, "
ELSE
    REPLACE P:P BY "OLD SN = ", POINTER(OLDSN) FOR 6, ", ";
IF NEWSN IS Ø THEN
    REPLACE P:P BY "NO NEW SN"
ELSE
    REPLACE P:P BY "NEW SN = ", POINTER(NEWSN) FOR 6;
IF POOL.[47:8] > Ø THEN
    REPLACE P:P BY ", POOL = ", POINTER(POOL)+1 FOR POOL.[47:8];
REPLACE P:P BY 48"ØØ";
DISPLAY (MSG);
%
%Take various actions depending on serial number
%and/or pool name
%
IF OLDSN = "PERMIT" THEN
    DISPLAY ("ANYTHING GOES")
ELSE IF NEWSN.[47:4Ø] = "MPOOL" THEN
    CHECK_TAPE_PURGE := TRUE & BOOLEAN(2) [47:Ø8]
ELSE IF NEWSN.[47:4Ø] = "MSNUM" THEN
    CHECK_TAPE_PURGE := TRUE & BOOLEAN(1) [47:Ø8]
ELSE IF NEWSN.[47:4Ø] = "MTELL" THEN
    BEGIN
        REPLACE POINTER(MSG) BY "MT", A[4] FOR * DIGITS,
        " NOT PURGED BECAUSE YOU TOLD ME NOT TO"48"ØØ";
        DISPLAY (MSG);
        CHECK_TAPE_PURGE := TRUE;
    END
ELSE IF NEWSN.[47:4Ø] = "MDENY" THEN
    CHECK_TAPE_PURGE := TRUE
ELSE IF NEWSN.[47:4Ø] = "MDSME" THEN
    MYSELF.STATUS := VALUE(TERMINATED)
ELSE IF NEWSN.[47:4Ø] = "M16ØØ" THEN
    NEWDEN := VALUE(BPI16ØØ)
ELSE IF NEWSN.[47:4Ø] = "M125Ø" THEN
    NEWDEN := VALUE(BPI125Ø)
ELSE IF POINTER(POOL) = 48"ØA"8"CHANGE_MGR" THEN
    BEGIN
        REPLACE POINTER(POOL) BY
            48"Ø7"8"CHANGED";
        CHECK_TAPE_PURGE := FALSE & BOOLEAN(1) [2:1];
    END
ELSE IF POINTER(POOL) = 48"ØD"8"CHANGE_ME_TRY" THEN
    BEGIN
        % ERROR TEST
        DISPLAY ("TRYING CHANGE WITHOUT BIT");
        REPLACE POINTER(POOL) BY
            48"Ø7"8"CHANGED";
        % FORGET TO CHANGE BIT 2 AND SEE WHAT HAPPENS
    END

```

TAPEMANAGER Library Implementation (cont.)

```
ELSE IF POINTER(POOL) = 48"ØD"8"CHANGE_ME_BAD" THEN
  BEGIN                                     % ERROR TEST
  DISPLAY ("TRYING BIGGER NAME");
  REPLACE POINTER(POOL) BY
    48"12"8"LENGTH_TOO_BIG";
    % WILL FAULT IF REALLY MAKE TOO BIG:
    % ONLY 3 WORDS LEFT IN ARRAY
  CHECK_TAPE_PURGE := FALSE & BOOLEAN(1) [2:1];
  END;
END CHECK_TAPE_PURGE;

EXPORT CHECK_TAPE_ASSIGNMENT,
  CHECK_TAPE_UNIT_ASSIGNMENT,
  CHECK_TAPE_REOPEN,
  CHECK_TAPE_PURGE,
  PROGRAM_INTERFACE,
  CONTROL_INTERFACE;
FREEZE(TEMPORARY);
AWAY:
END.
```

The following sample program is called CT/SET/VOLFAM/SERIALNO. It is referenced by the previous sample program, and is an example of a program that calls the CALL_TAPEMANAGER procedure.

```

$ SET LEVEL 2
PROCEDURE SETSN(A);
ARRAY A[*]; % A IS STRING OF UPTO 9 SERIALNOS, STARTING WITH FS 1
            % VALUE IS VOLFAM FOR WHICH IT IS BEING SET
BEGIN
LIBRARY MCP (LIBACCESS=BYFUNCTION,FUNCTIONNAME="MCPSUPPORT");
REAL PROCEDURE CALL_TAPEMANAGER (UNLINKED_ACTION, A);
    VALUE UNLINKED_ACTION;
    REAL UNLINKED_ACTION;
    ARRAY A[*];
    LIBRARY MCP;
ARRAY INPUTMSG [0:14];
POINTER PA;
INTEGER FS;
REAL R;
INPUTMSG[0]:="VOLFAM";
INPUTMSG[1]:=MYSELF.TASKVALUE;
PA:=POINTER(A);
WHILE PA NEQ 48"00" DO
    BEGIN
        INPUTMSG[2]:=*+1;
        REPLACE POINTER(INPUTMSG[3]) BY PA:PA FOR 6;
        R:=CALL_TAPEMANAGER(0,INPUTMSG);
    END;
END.

```

Exported MCP Procedures for Tape Management

The following exported MCP procedures provide functions related to tape management:

- **CALL_TAPEMANAGER.** Tape management utilities can use this procedure to communicate with the TAPEMANAGER library. The CALL_TAPEMANAGER procedure is described in the following pages.
- **CALL_TAPESERVER.** The TAPEMANAGER library or other software can use this procedure to communicate with the TAPESERVER library. For more information about the A Series RoboHost A TAPESERVER library features and functions, refer to the *A Series RoboHost A Programming Guide*.
- **REPORT_LOG_ENTRIES.** A TMS can use this procedure to monitor successful tape actions. Refer to the *A Series System Software Support Reference Manual* for the detailed procedure.
- **STATUS_CHANGE_REQUEST.** A TMS can use this procedure to obtain information about tape events. Refer to the *A Series System Software Support Reference Manual* for the detailed procedure.
- **GETSTATUS** and **SETSTATUS.** A TMS can use these functions to obtain information about the status of tape units on the system. Refer to the *A Series GETSTATUS/SETSTATUS Programming Reference Manual*.
- **SYSTEMSTATUS.** A TMS can use this function to determine the number and type of tape units available on the system. Refer to the *A Series SYSTEMSTATUS Programming Reference Manual*.

CALL_TAPEMANAGER Procedure

The CALL_TAPEMANAGER procedure in the MCPSUPPORT library can be called by any program that needs to communicate with the TAPEMANAGER library. The CALL_TAPEMANAGER procedure in turn invokes the PROGRAM_INTERFACE procedure in TAPEMANAGER. CALL_TAPEMANAGER is provided so that user programs can communicate with the TAPEMANAGER library, even though only the MCP is allowed to link to TAPEMANAGER directly.

CALL_TAPEMANAGER accepts an array parameter that it passes through to PROGRAM_INTERFACE. CALL_TAPEMANAGER also accepts a parameter indicating the action to be taken if the TAPEMANAGER library status is not ACTIVE.

The CALL_TAPEMANAGER procedure returns immediately if the TAPEMANAGER system option has not been set by the SYSOPS (System Options) system command. If the TAPEMANAGER library status is not ACTIVE, the CALL_TAPEMANAGER procedure informs the operator that the library is not ACTIVE, and either waits until it becomes ACTIVE or returns immediately, depending on the caller's request. A CALL_TAPEMANAGER call does not itself cause the library to be initiated.

If PROGRAM_INTERFACE modifies the array parameter passed to it, these changes are reflected when the parameter is returned by CALL_TAPEMANAGER. Also, CALL_TAPEMANAGER returns a procedure value that either duplicates the procedure value returned by PROGRAM_INTERFACE, or indicates that the MCP is unable to invoke PROGRAM_INTERFACE.

For further information, refer to "PROGRAM_INTERFACE" earlier in this section.

Syntax

Specify the procedure heading and the MCPSUPPORT library as follows:

```

LIBRARY MCPSUPPORT(LIBACCESS = BYFUNCTION,FUNCTIONNAME = "MCPSUPPORT.");

REAL PROCEDURE CALL_TAPEMANAGER (UNLINKED_ACTION,
                                REQUEST_AND_RESPONSE);
    VALUE UNLINKED_ACTION;
    REAL UNLINKED_ACTION;
    ARRAY REQUEST_AND_RESPONSE [*];
LIBRARY MCPSUPPORT;
```

Explanation

The description of the parameters follows.

UNLINKED_ACTION (Real, Input Only)

This parameter is divided into the following fields:

Field	Value	Meaning
[47:40]		Field reserved for future use
[7: 8]		Action to be taken if the library status is not ACTIVE at the time of the call
	0	Wait until the library status is ACTIVE or an operator enters an OF (Optional File) or DS (Discontinue) response.
	1	Return immediately.

Note: *This action is taken, regardless of the value of UNLINKED_ACTION, if the SYSOPS TAPEMANAGER system option is not set.*

REQUEST_AND_RESPONSE (Real Array [*], Input/Output)

The contents and meaning of this parameter are defined by the TAPEMANAGER library implementation.

Procedure Value (Real, Output Only)

The following table describes the meaning of the procedure values:

Value	Meaning
> 0	The meaning is defined by the TAPEMANAGER library implementation.
0	Either the PROGRAM_INTERFACE procedure was not present in TAPEMANAGER, or PROGRAM_INTERFACE returned a value of 0 for implementation-defined reasons.
-1	The TAPEMANAGER library was not called because the library state was not ACTIVE.
-2	The TAPEMANAGER library was called but returned an invalid result (that is, a result less than 0).
-3	The TAPEMANAGER library was not called because the TAPEMANAGER system option was not set by the SYSOPS (System Options) system command.

Operations Interface to the TAPEMANAGER Library

The following pages discuss only those aspects of the TAPEMANAGER library that are of interest to system operators.

The TAPEMANAGER library is an optional system library that is implemented by the customer site or a third-party vendor. If a TAPEMANAGER library is installed on an A Series system, and contains the standard set of exported procedures, then the MCP invokes TAPEMANAGER procedures whenever any of the following events occurs:

- A process attempts to open a tape file.
- A reel switch occurs for a tape file.
- A purge request occurs for a tape volume. Purge requests result from PG (Purge) and SN (Serial Number) system commands, or from program statements that close a file with a disposition of purge.

The MCP uses parameters to pass TAPEMANAGER information about the open, reel switch, or purge request. The TAPEMANAGER procedures return results that either accept, deny, or modify the request. If an operator modifies an open request with system commands such as FA (File Attribute) or OU (Output Unit), the MCP invokes the TAPEMANAGER procedures again to verify that the modified request is acceptable.

For more detailed information about the interactions between application programs and the TAPEMANAGER library, refer to "Interaction with the Operating System" earlier in this section.

The following paragraphs discuss the steps involved in installing, initiating, terminating, and inspecting the status of the TAPEMANAGER library.

Installing the TAPEMANAGER Library

Use the following steps to install a TAPEMANAGER library:

1. Use the SL (Support Library) system command to define the function name TAPEMANAGER and to associate the function name with a library code file. The following is an example of this command:

```
SL TAPEMANAGER = (JASMITH)OBJECT/OURTAPEMANAGER ON DISK
```

2. Use the SYSOPS (System Options) system command to set the TAPEMANAGER option. The following is an example of this command:

```
SYSOPS TAPEMANAGER +
```


Unisys recommends the following additional actions for TAPEMANAGER installation, but they are not essential:

- Make the library code file private for the reasons discussed under “Security Considerations” earlier in this section. For example, you could enter the following command at the ODT:

```
SECURITY *SYSTEM/TAPEMANAGER ON DISK PRIVATE IO
```

- Assign an identity to the library code file. This identity appears in any system messages or display messages generated by a process that is executing TAPEMANAGER code. You can assign the identity with the MP (Mark Program) system command, as in the following example:

```
MP *SYSTEM/TAPEMANAGER ON DISK + IDENTITY = TML
```

Initiating the TAPEMANAGER Library

If the TAPEMANAGER function name is defined and the TAPEMANAGER system option is set, then you can initiate the TAPEMANAGER library in either of the following ways:

- Enter the *SEND TAPEMANAGER START* form of the SEND (Send Message) system command. This command can be abbreviated to *SEND TM START*.
- Wait for the next halt/load. When the system reinitializes, it automatically initiates the TAPEMANAGER library.

For further information about TAPEMANAGER library initiation, refer to “Starting Up” earlier in this section.

Inspecting the TAPEMANAGER Status

You can determine the status of the TAPEMANAGER library by entering the *SEND TM STATUS* form of the SEND (Send Message) system command. The following table describes the states that the system can display in response to this command.

State	Meaning
INACTIVE	The library is not linked, nor is initiation or termination underway.
LINKING	The MCP is in the process of linking to the library, and the library has not yet frozen.
INITIALIZING	The MCP is linked to the library, and the CONTROL_INTERFACE procedure is currently executing the start request.
ACTIVE	The library is frozen, linked, and available for use.
TERMINATING	A SEND TM QUIT system command has been entered. The MCP has begun suspending any new processes that attempt tape actions requiring calls on the library. The CONTROL_INTERFACE procedure is currently executing the stop request.

continued

continued

State	Meaning
WAITING TO DELINK	The CONTROL_INTERFACE procedure has exited after executing the stop request. The library remains in the WAITING TO DELINK state until the MCP delinks from the library, which the MCP does after all procedure calls that are currently in progress have completed.
GOINGAWAY	The MCP has delinked from the library. If the library is a TEMPORARY library, it unfreezes at this point and resumes execution as a normal process. The library remains in the GOINGAWAY state until it terminates.

Terminating the TAPEMANAGER Library

You can terminate the TAPEMANAGER library by entering one of the following forms of the SEND command:

```
SEND TAPEMANAGER QUIT
SEND TAPEMANAGER QUIT <text>
```

These commands can also be abbreviated as follows:

```
SEND TM QUIT
SEND TM QUIT <text>
```

If text is specified in the command, the text is passed to the RESPONSE parameter of the CONTROL_INTERFACE procedure in the TAPEMANAGER library. The meaning of the text is determined by the TAPEMANAGER implementation.

You can also terminate the TAPEMANAGER library process with a DS (Discontinue) system command. However, you must bear the following considerations in mind:

- The DS command has no effect on the TAPEMANAGER library process unless you first use the <mix number> -LP form of the LP (Lock Program) system command to unlock the library process. This step is necessary because the SEND TM START command implicitly locks the library process.
- The DS command is the nonpreferred method of terminating TAPEMANAGER, because it prevents user processes from finishing any open operations that are in progress, and prevents TAPEMANAGER from performing any cleanup actions that it normally might perform before terminating.

If the TAPEMANAGER library takes unexpectedly long to terminate, you can check the status as described under “Inspecting the TAPEMANAGER Status” earlier in this section. The following table lists TAPEMANAGER status values, and actions that the operator can take if TAPEMANAGER becomes stuck in one of these states.

Status	Operator Response
TERMINATING	The CONTROL_INTERFACE procedure is executing, and might have a bug that is causing it to loop. You can unlock the library with the LP (Lock Program) system command and then discontinue the library with the DS (Discontinue) system command.
WAITING TO DELINK	It is possible that one of the user processes is suspended or is looping because of a bug in one of the TAPEMANAGER exported procedures. If you can locate the offending user process, you can discontinue that process with a DS command, and the library should then terminate normally. If you cannot locate the offending user process, you can simply unlock the library with an LP command and then discontinue the library with a DS command.
GOINGAWAY	<p>It is possible that the library has a PERMANENT freeze that keeps the library from unfreezing and resuming execution. You can use the LIBS (Library Mix Entries) system command to determine the library freeze type. If the freeze is permanent, you can use the THAW (Thaw Frozen Library) system command to change the freeze to TEMPORARY, and the library should unfreeze and terminate sometime thereafter.</p> <p>If the TAPEMANAGER library does not appear in the LIBS display, then the library has unfrozen and is executing as a normal process. There might be a bug in the TAPEMANAGER code that causes it to loop. You can unlock the library with an LP command and then discontinue the library with a DS command.</p>

For further information about TAPEMANAGER library termination, refer to “Shutting Down” earlier in this section.

Terminating the TAPEMANAGER library can cause processes that use tapes to become suspended. This issue is discussed under the following heading.

Temporarily Disabling Use of the TAPEMANAGER Library

If you enter the `SYSOPS TAPEMANAGER -` command after TAPEMANAGER is initiated, the system temporarily disables use of the TAPEMANAGER library. The MCP resumes use of the TAPEMANAGER library when you enter the `SYSOPS TAPEMANAGER +` form of the SYSOPS command.

Note: *While TAPEMANAGER is disabled, the MCP allows processes to use tape files as if the TAPEMANAGER library were not installed. This behavior is not desirable if you are relying on the TAPEMANAGER library to enforce tape security. You might prefer to terminate TAPEMANAGER instead, by entering the SEND TM QUIT form of the SEND (Send Message) system command. While TAPEMANAGER is terminated, processes that attempt to open a tape file or purge a tape become suspended, and the operator can respond to the waiting entries individually. Refer to “Terminating the TAPEMANAGER Library” earlier in this section.*

Responding to Processes Waiting on the TAPEMANAGER Library

If the TAPEMANAGER system option is set, then the system attempts to invoke the relevant TAPEMANAGER procedures whenever a process attempts to open or purge a tape. If the TAPEMANAGER library is not currently linked, for example because an operator has terminated the library with a SEND TM QUIT command, then processes attempting to open or purge tape files become suspended. The RSVP request, system message (MSRMISC220), is: "TAPEMANAGER REQUIRED." The following table describes the possible operator responses and their effects:

Response	Effect
OK	The MCP attempts to call the TAPEMANAGER library again.
DS	The MCP discontinues the process attempting to open the file.
OF	The open or purge request continues as if the TAPEMANAGER library were not installed.

Because TAPEMANAGER might be used to enforce tape security at some sites, the MCP limits the use of the OF response. The OF response can be submitted from the following sources:

- ODTs
- Remote ODTs; that is, remote terminals granted ODT status by the REMOTESPO (Activate REMOTESPO) system command
- Nonusercoded MARC sessions
- DCKEYIN functions in nonusercoded DCALGOL processes

The MCP ignores the OF response if it is submitted from CANDE sessions, usercoded MARC sessions, or usercoded DCALGOL processes.

The MCP also resumes execution of the process automatically if TAPEMANAGER library linkage completes while the process is suspended.

A user process can also become suspended if it encounters a fault while executing a TAPEMANAGER procedure. The RSVP request in this case is system message MSRMISC221: "USER PROCESS FAULTED OR DSED WHILE IN TAPEMANAGER LIBRARY CODE." The possible responses are the same as described previously: OK, DS, or OF. For further information, refer to "TAPEMANAGER Error Handling" earlier in this section.

Replacing the TAPEMANAGER Library with a Newer Version

Use the following steps to replace the currently running TAPEMANAGER library with a newer version:

1. Use a *WFL COPY* statement to copy the new version of the library object code file to the desired file name and family. You can assign the same file name and family name as the old version has. In this case, the new version of the code file becomes a permanent disk file, and the old version of the code file becomes a temporary disk file but continues to be used for the moment.
2. Enter the *SL TAPEMANAGER = <code file title>* form of the SL (Support Library) system command. Use this command even if the new version of the code file has the same title as the old version. This command has no immediate effect on the library instance that is running, but it does specify the code file to be used the next time the library is initiated.
3. Enter a *SEND TM QUIT* or *SEND TM QUIT <text>* command to terminate the current TAPEMANAGER library instance.
4. Wait until the current TAPEMANAGER library instance terminates. TAPEMANAGER has terminated if the *SEND TM STATUS* command returns a status of *INACTIVE*.
5. Enter a *SEND TM START* command to initiate the new TAPEMANAGER library instance.

Appendix A

System Messages

This appendix presents selected A Series system messages generated by the master control program (MCP)—specifically, system messages related to A Series tape management facilities.

This appendix does not include any messages generated by system libraries, message control systems (MCSs), database management software, or application programs.

Note: *The information in this appendix is based on the A Series System Messages Support Reference Manual.*

This appendix follows the basic style and conventions of the *A Series System Messages Support Reference Manual*. That is

- The explanation of each message consists of two bulleted items, the first describing the cause of the error and the second presenting a suggested response.
- The associated message number follows the text of the message, and is enclosed in parentheses. When a message appears in the log, the message number precedes the message text. All system message numbers begin with the characters *MSR*.
- Message presentations include metatokens. Metatokens are variables whose values change according to the current situation and the current user. In this appendix, metatokens are surrounded by angle brackets (< >). Table A-1 presents some of the metatokens that appear in system messages throughout this appendix. A metatoken that appears in a limited number of messages is defined with those individual messages. When they are included, metatoken definitions follow the message text and precede the probable cause of the message.
- Messages are presented in alphabetic order. In locating a message alphabetically, do not consider the following information and characters at the beginning of a message:
 - Metatokens such as <file name> or <unit name>
 - Device acronyms such as MT or PK
 - Parentheses and the information they enclose; brackets and the information they enclose; and nonalphabetic characters such as numbers, commas, periods, and question marks

Table A-1. Metatokens Appearing in Message Listings

Metatoken	Definition
<family name>	Name of a family.
<FIB error prefix>	The following information: <div style="text-align: center;">FILE <file name> <error category></div> The variable <error category> is one of the following: <div style="text-align: center;"> OPEN <error type> CLOSE <error type> I/O <error type> INPUT REEL SWITCH: OUTPUT REEL SWITCH: I/O ERROR: NO FILES COPIED RESPOND: </div> The variable <error type> is one of the following: <div style="text-align: center;"> WARNING: ERROR: </div>
<file name>	Name of a file, with no family name appended.
<file title>	Complete identifier for a file, which consists of the file name, the word <i>ON</i> , and the family name.
<host name>	Name of a system as established with the HOSTNAME (Hostname) system command.
<mix number>	Mix number of the program that encountered the error described by the message. The mix number appears in parentheses.
<number>	An integer. This value can be, for example, hours, minutes, and seconds, or it can be the record number of a directory record. In some cases, integers are represented in the form <i>nnn</i> and hexadecimal numbers are represented in the form <i>hhh</i> .
<percentage>	Percentage of a process that has completed.
<rcw>	Return control word. This value indicates where in the process an interrupt or procedure call occurred.
<rrr>	Row number in a file.
<serial number>	Either the 6-digit serial number of a disk or the 6-character serial number of a tape.
<serial number list>	One or more volume serial numbers. If the list includes more than two serial numbers, they might appear as follows: [nnnnnn, . . . nnnnnn]
<unit identifier>	Either a unit name or a unit number.

continued

Table A-1. Metatokens Appearing in Message Listings (cont.)

Metatoken	Definition
<unit kind>	Device acronym, such as PK or MT.
<unit name>	Device acronym and unit number, such as PK 97 or MT 80.
<unit number>	Unit number of a peripheral, such 97 or 80.
<volume description>	A volume name, a volume serial number, or both.
<vol dir extension>	An indicator of the reason a volume directory action was taken. ADD BUILD CHANGE DELETE DESTROYED FINDINPUT FINDOUTPUT OPEN PG MT READALABEL REBUILD RESTRICT SEARCH SETSTATUS ADD SN MT
<volume family>	PACK, DISK, SCRATCH, or a family name.

See the *System Messages Manual* for details on how system messages are documented.

**PK<unit number> ARCHIVE BACKUP ENTRY DELETED
 <file name> ON <family name>, FOR TAPE <tape name>
 [<serial number>] (MSRLM92)**

- The system erased from the family's archive directory the archive backup information for a file previously reported as *COPIED AND ARCHIVED*. Archive information created by an archive backup or rollout operation is erased in either of the following situations:
 - A tape is purged in response to a VERIFY error.
 - The archive process is initiated with the & DSONERROR option and any error occurs.

System Messages

- Determine the cause of the original error. Correct the error and resubmit the WFL *ARCHIVE* statement.

<FIB error prefix> ASSIGNMENT STOPPED BY TAPEMANAGER
<rcw> (MSRFIB271)

- A process attempted to open the indicated tape file. However, the TAPEMANAGER system library specified that this open request should be terminated with an open error.
- Contact your site's support personnel or the developer of your TAPEMANAGER library to determine why the request was denied. See Section 5, "TAPEMANAGER Library Implementation," for more information on TAPEMANAGER libraries.

MT<unit number> BACKSPACE FAILED TO REACH TAPEMARK
<file name> (MSRLM27)

- The library maintenance procedure was unable to recopy or compare a disk file on a library maintenance tape because the procedure was unable to locate a tape mark. If the error occurred on an input source tape, the procedure discontinues copying files. If the error occurred on a destination tape, the procedure discontinues copying files to the affected tape.
- Resubmit the WFL *COPY* or *ARCHIVE* statement on a different tape volume or tape drive.

MT<unit number> BAD DISK HEADER ON TAPE <file name>
(MSRLM12)

- You attempted to copy a file from tape with the WFL *COPY* statement. However, the disk file header read from the tape was either corrupted or the incorrect size. The system discontinued copying files.
- Perform as many of the following steps as necessary:
 - Clean the tape drive and resubmit the *COPY* statement.
 - If the operation fails again, move the tape volume to another tape unit and resubmit the *COPY* statement.
 - If the operation still fails, the problem is likely to be with the tape itself. Attempt to investigate how the the tape was created and correct any inappropriate characteristics.

**CANNOT QUIT TAPEMANAGER, CURRENTLY <status>
(MSRMISC213)**

The variable <status> is one of the following:

INACTIVE
LINKING
TERMINATING
WAITING TO DELINK
GOING AWAY

- The TAPEMANAGER is in a state that is invalid for a SEND TM QUIT command to be processed.
- If the status is LINKING, wait for the TAPEMANAGER to become INITIATING or ACTIVE and then resubmit the request. In all other cases, the TAPEMANAGER will eventually become INACTIVE.

**CANNOT START TAPEMANAGER, CURRENTLY <status>
(MSRMISC212)**

The variable <status> is one of the following:

ACTIVE
LINKING
INITIALIZING
TERMINATING
WAITING TO DELINK
GOING AWAY

- The TAPEMANAGER is currently in a state other than INACTIVE.
- Contact your site's support personnel or the developer of your TAPEMANAGER library to determine why this situation exists. See Section 5, "TAPEMANAGER Library Implementation," for more information on TAPEMANAGER libraries.

**DESTROYED VOLUME NOT ADDED IN SCRATCH POOL (MT)
[<volume family>] (MSRVL19)**

- While initiating a restart of a program that used the checkpoint facility, the restart procedure attempted to release certain volumed tapes as scratch tapes. However, these volumed tapes were marked as DESTROYED after the checkpoint was taken. The restarting task is unaffected, and the restart process continues.
- No action is required.

**MT<unit number> DISK FILE HEADER VERSION ERROR
<file name> (MSRLM37)**

- The input tape contains a disk file that was created by a newer version of the operating system. The current operating system is unable to copy the file.
- Use the CM (Change MCP) system command to change to the advanced Mark release version of the operating system; then resubmit the COPY request.

**MT<unit number> END-OF-FILE READING HDR1 LABEL
<file name> (MSRLM31)**

- The library maintenance procedure encountered a tape mark where it expected a tape label. The procedure discontinues copying files.
- If the input tape was created correctly—that is, the library maintenance operation that wrote the input tape completed successfully—resubmit the request with a different tape drive.

**MT<unit number> ERROR CLOSING TAPE FILE <file name>
(MSRLM26)**

- During the processing of a WFL *COPY* or *ARCHIVE* statement, the system CLOSE routine returned an error indication between files. If the error occurred on a source tape, the system is unable to copy any more files. If the error occurred on a destination tape, the system is unable to copy any more files to that tape.
- Resubmit the COPY or ARCHIVE statement with a different tape volume or tape drive.

**MT<unit number> ERROR OPENING TAPE FILE <file name>
(MSRLM29)**

- The system OPEN routine either failed to correctly locate an input tape file or it failed to correctly allocate an output tape file. The system is unable to copy any additional files.
- Determine the cause of the open error and resolve that problem. Resubmit the COPY request with different tapes or tape drives.

**MT<unit number> ERROR READING TAPE LABEL <file name>
(MSRLM21)**

- An I/O error occurred while the system was reading either a tape HDR1, HDR2, EOF1, EOF2, or EOVI label, or a tape mark in response to a WFL *COPY* statement. The system discontinues copying files.
- Resubmit the COPY statement with a different tape drive or tape volume.

**MT<unit number> ERROR WRITING TAPE LABEL OR TAPE
MARK <file name> (MSRLM8)**

- An I/O error occurred while the system was performing one of the following operations:
 - Writing a tape HDR1, HDR2, EOF1, EOF2, or EOVI label with the WFL *COPY* or *ARCHIVE* statement
 - Writing a tape mark on the output tapeThe system is unable to copy any additional files.
- Resubmit the request on a different tape drive or tape volume.

**MT<unit number> EXPECTED RECORD ID <number> BUT
WORDS 1&2 OF RECORD = <number> <number> (MSRLM39)**

- The library maintenance procedure read an incorrect record identifier from the tape. The procedure is unable to copy any more files to or from the specified unit.
- If the procedure follows this message with a “RECOPY REQD” RSVP request, see the description of that RSVP request later in this appendix for a discussion of your options; enter the response of your choice.

If appropriate, resubmit the WFL *COPY* or *ARCHIVE* statement with a different I/O unit.

**<unit identifier> FAMILYOWNER MISMATCH OR SECURITY
ERROR: <file name> (MSRFOUT9)**

- The system has rejected a requested tape operation involving the indicated file on the indicated tape drive because one of the following situations exists:
 - The FAMILYOWNER values on the tape label and in the volume directory do not match.
 - A security error occurred.

System Messages

- Determine the cause of the error, take appropriate corrective measures, and resubmit the request.

FAULT REASON = <fault number> <rcw> (MSRMISC223)

The variable <fault number> is the number of a fault condition, as follows:

Fault Number	Fault Name
1	ZERODIVIDE
2	EXPONENTOVERFLOW
3	EXPONENTUNDERFLOW
4	INVALIDINDEX
5	INTEGEROVERFLOW
7	MEMORYPROTECT
8	INVALIDOP
9	LOOP
10	MEMORYPARITY
11	SCANPARITY
12	INVALIDADDRESS
14	STRINGPROTECT
15	PROGRAMMEDOPERATOR
16	ASSERTIONFAILURE
18	INVALIDPROGRAMWORD

This message is always preceded by the following RSVP request:

USER PROCESS FAULTED OR DSED WHILE IN TAPEMANAGER LIBRARY CODE

This RSVP request is discussed in its respective alphabetical position among the other message descriptions in this appendix.

- The indicated fault condition occurred. The return control word (rcw) is a pointer to the stack history array.
- Contact your site's support personnel or the developer of your TAPEMANAGER library to correct the problem. See Section 5, "TAPEMANAGER Library Implementation," for more information on TAPEMANAGER libraries.

FILE <file name> <error category> . . .

Note: *A number of system messages begin with this information, which is represented in this appendix by the metatoken <FIB error prefix>. See Table A-1, "Metatokens Appearing in Message Listings," for the values the <error category> metatoken can have.*

This appendix lists these messages in alphabetic order according to the text that follows the <FIB error prefix> metatoken. For example, the complete format of system message MSRFIB271 is: "FILE <file name> <error category> ASSIGNMENT STOPPED BY TAPEMANAGER <rcw>."

However, this appendix replaces the "FILE <file name> <error category>" information with the <FIB error prefix> metatoken. And since this appendix ignores metatokens in determining the alphabetic order of messages, MSRFIB271 is considered to begin with ASSIGNMENT rather than FILE. Thus, the entry for MSRFIB271 appears between the entries for MSRLM92, "ARCHIVE BACKUP ENTRY DELETED," and MSRLM12, "BAD DISK HEADER ON TAPE."

MT<unit number> FILE NOT VERIFIED <file name> (MSRLM88)

- A file that was copied to tape by a WFL *COPY* or *ARCHIVE* statement with the & *VERIFY* clause was incorrectly verified. This message is usually preceded by other error messages describing the reasons for the failure. The library maintenance utility continues with the next file.

If the file name is unknown to the system, no file name appears in the message. Instead, the message includes the file number in the following format:

FILE # nnn

To determine the name of the file in question, correlate the file names in *VERIFIED* and *NOT VERIFIED* messages with the file names that appeared in *COPIED* or *NOT COPIED* messages.

- Note the name of the file and any preceding error messages. Decide whether to recopy the file with a different unit or a different tape volume.

MT <unit number> FILE TITLE ERROR <file name> (MSRLM11)

- During a library maintenance COPY operation from tape, the library maintenance procedure found that the file title of a file is other than the expected title. The operation copies no more files.
- Determine whether the input tape was created correctly by examining the job log output generated by the WFL COPY or ARCHIVE statement that originally created the tape. If the tape was created correctly, resubmit the request with a different tape drive.

MT<unit number> FILE VERIFIED <file name> (MSRLM86)

- This message is issued for each output tape file that is successfully copied and verified.
- No action is required.

HISTORY REASON = <reason>, CAUSE = <cause>, TYPE = <type> <rcw> (MSRMISC224)

The variable <reason> is an integer indicating the specific reason for the abnormal termination or suspension of a task. This value corresponds to the value of field [23:08] of the HISTORY task attribute and to the HISTORYREASON task attribute value.

The variable <cause> is an integer indicating the general type of condition that caused the task to terminate abnormally or be suspended. This value corresponds to the value of field [15:08] of the HISTORY task attribute and to the HISTORYCAUSE task attribute value.

The variable <type> is an integer indicating the type of termination that occurred for a process. This value corresponds to the value of field [07:08] of the HISTORY task attribute and to the HISTORYTYPE task attribute value.

This message is always preceded by the following RSVP request:

```
USER PROCESS FAULTED OR DSED WHILE IN TAPEMANAGER LIBRARY CODE
```

This RSVP request is discussed in its respective alphabetical position among the other message descriptions in this appendix.

- A process has terminated under the indicated conditions.
- Assess the process termination conditions; take any indicated corrective measures.

Note: For information on the *HISTORY*, *HISTORYREASON*, *HISTORYCAUSE*, and *HISTORYTYPE* task attributes, see Section 3, "Programming Considerations," as well as the A Series Task Attributes Programming Reference Manual.

**<unit name> <serial number> INVALID DENSITY SPECIFIED
(MSRDPC96)**

- While executing a PG (Purge) or SN (Serial Number) system command that involved the specified unit, the system encountered a density specification that is invalid for that unit. The system purges the tape.
- Reenter the command with one of the following modifications:
 - Include the correct density specification.
 - Omit the density specification entirely. In this case, the density remains unchanged by the command.

**MT<unit number> INVALID LIBRARY/MAINT TAPE BLOCKSIZE,
DEFAULT ASSUMED (MSRLM96)**

- While reading the header labels for the first file from an input tape, the library maintenance procedure determined that the block size recorded in the file header 2 (HDR2) label is not a valid value for a library maintenance tape. The library maintenance procedure assumes the default block size for the type of tape involved and attempts to proceed with copying files from the tape.
- Review any I/O error messages that preceded this message. If possible, examine the log to determine if the tape was originally written correctly by the library maintenance or archive process that created it. Take any indicated corrective measures.

**MT <unit number> IO ERROR DURING COMPARE <file name>
(MSRLM15)**

- An I/O error occurred during the COMPARE procedure while the library maintenance procedure was reading from a tape unit.
- If the procedure follows this message with a “RECOPY REQD” RSVP request, see the description of that RSVP request later in this appendix for a discussion of your options; enter the response of your choice.

If appropriate, resubmit the WFL *COPY* or *ARCHIVE* statement with a different tape unit.

**MT <unit number> IO ERROR DURING COPY <file name>
(MSRLM4)**

- An error occurred while the library maintenance procedures was reading or writing data to or from the tape unit for the specified file.
- If the procedure follows this message with a “RECOPY REQD” RSVP request, see the description of that request later in this appendix for a discussion of your options; enter the response of your choice.

If appropriate, resubmit the WFL *COPY* or *ARCHIVE* statement with a different I/O unit.

**MT<unit number> I/O ERROR ON CHECKSUM RECORD
<file name> (MSRLM84)**

- An I/O error occurred when the library maintenance procedure read the checksum of a file from a tape or wrote the checksum of a file to a tape during the initiation of a WFL *COPY*, *ADD*, or *ARCHIVE* statement with the & VERIFY clause.
- Note the name of the file and the preceding I/O logical result descriptor error message. Decide whether to copy the file again using a different unit or a different tape volume.

**MT<unit number> I/O ERROR READING HDR1 LABEL
<file name> (MSRLM30)**

- An I/O error occurred while the library maintenance procedure was reading a tape label from an input tape. The affected file is not copied.
- If the input tape is damaged or was created incorrectly, the system is unable to copy the affected file. Otherwise, resubmit the WFL *COPY* or *ARCHIVE* statement with a different tape drive.

**I/O ERROR WHILE ACCESSING GUARDFILE <file name> <rcw>
(MSRMISC201)**

- An I/O error occurred while the system was attempting to access the indicated guard file.
- Determine the cause of the I/O error, correct that problem, and resubmit the request.

**<FIB error prefix> LIBRARY CANNOT OPEN THIS FILE <rcw>
(MSRFIB270)**

- The TAPEMANAGER library has attempted to open a file for which an OPEN request is already underway.
- Contact your site's support personnel or the developer of your TAPEMANAGER library to correct the problem. See Section 5, "TAPEMANAGER Library Implementation," for more information on TAPEMANAGER libraries.

NEED AN OUTPUT TAPE FOR OUMT (MSRFOUT11)

- The system rejected your response to an RSVP request. You entered an OU (Output Unit) system command in the form

OU MT

However, no suitable tape is available.

- Load a scratch tape and ready it with an RY (Ready) system command. Then reissue the OU command.

**MT<unit number> NEW SOURCE UNIT OR VOLUME IS
RESTRICTED (MSRLM42)**

- During a file copy operation from tape, a reel switch occurred on the source file. The new file has been named in a RESTRICT (Restrict Unit or Volume) system command to indicate that it is a "hazardous" file. A hazardous file is one that has the potential to compromise the security of the system.

After this message, the following RSVP request appears:

UNRESTRICT SOURCE AND REPLY OK TO COPY <file name>

System Messages

This RSVP request is discussed in its respective alphabetical position among the other message descriptions in this appendix.

The system waits for your response.

- Review the options and take the action of your choice.

NO CATALOG ACTION (LIBRARY TAPE) (MT) <file name> (MSRCAT16)

- You attempted to execute a *WFL CATALOG DELETE* statement on a tape file that is located on a library maintenance tape. Library maintenance tape files are never cataloged.
- No action is required.

<unit identifier> NO ENTRY FOR VOLUME IN 'VOLUME DIRECTORY': <file name> (MSRFOUT15)

- The system is unable to find an entry in the volume directory for the indicated volume.
- Determine whether the request specified the correct volume name. Resubmit the request with a corrected name if appropriate.

MT<unit number> NO RECOPY WHEN USING THIS KIND OF TAPE <file name> (MSRLM69)

- Either the source or the destination volume does not support the *RECOPY* option. The library maintenance procedure continues to copy that file to other destinations as required. The procedure then copies the remaining files as required.
- Note the original error message and the name of the file that was copied. Decide whether to resubmit the copy request to retry the operation that failed.

**NOT A SCRATCH VOLUME <volume name> (<kind>)
[<serial number>] (MSRVL16)**

- You entered one of the WFL statements VOLUME DELETE SCRATCH or VOLUME DESTROY SCRATCH, but the volume you specified is not listed as SCRATCH in the volume library. The system rejects the request, and the job continues.

Note: When you ready a tape volume containing an expired tape file and enter the PER MT form of the PER (Peripheral Status) system command, the system displays the volume as a scratch tape. However, the volume library entry for expired tape files still uses the original volume name rather than the name SCRATCH.

- Display the status of the volume by entering a PV (Print Volume) system command. Then resubmit the VOLUME DELETE statement with the correct information.

<FIB error prefix> NOT CLOSED <rcw> (MSRFIB74)

- A program attempted to open the indicated file or a subfile of that file, but one of the following situations exists:
 - The file or subfile is a port file that has a FILESTATE attribute value other than CLOSED.
 - The file or subfile is a nonport file that is already open.
- Make appropriate corrections to the program.

**<unit name> MT <unit number> PURGE REQUEST REJECTED
BY TAPEMANAGER (MSRDPC177)**

- The TAPEMANAGER library rejected a request to purge the indicated tape. The message includes the name and number of the unit if those values are available; the message lists a null value for any unavailable value.
- Contact your site's support personnel or the developer of your TAPEMANAGER library to determine why the request was denied. See Section 5, "TAPEMANAGER Library Implementation," for more information on TAPEMANAGER libraries.

MT<unit number> RECOPY REQD <file name> (MSRLM5)

- This RSVP request is preceded by a library maintenance error message. Only some error conditions allow recopying, such as those indicated by the following messages:

IO ERROR DURING COPY
COMPARE ERROR
I/O ERROR DURING COMPARE

System Messages

Descriptions of these messages appear in their respective alphabetical positions among the other system message descriptions in this manual.

- The variable <unit number> refers to a specific destination because files can be copied to several destinations at the same time. Enter one of the following system commands:
 - <mix number> DS
Terminates the library maintenance operation. Any files already copied remain, but the file in question and any other files that were to be copied are not copied.
 - <mix number> OF
Prevents the file from being recopied. The library maintenance procedure continues with the next file.
 - <mix number> FR
Prevents any more files from being copied to this destination. If there are no other destinations, the library maintenance program terminates; otherwise, it continues copying to the other specified destinations.
 - <mix number> OK
Permits the file to be recopied. If the error persists, the library maintenance operation repeats the error message and asks again whether the file should be recopied.

MT<unit number> RECORD SEQUENCE ERROR <file name> (MSRLM7)

- During the execution of a WFL *COPY* statement, the sequence indicator read from the tape was either incorrect or out of sequence. If this error occurs while the library maintenance procedure is reading the source tape directory, the procedure discontinues copying files. If this error occurs while the procedure is reading the disk file header record from the source tape, the procedure does not copy that file. If a sequence error occurs while the procedure is reading data records from the source or destination tape, the procedure issues a "RECOPY REQD" RSVP request.
- If "RECOPY REQD" appears, see the description of that RSVP request earlier in this appendix for a discussion of your options and enter the response of your choice. Or, copy the file from another tape drive. If the problem persists for a particular input tape, the tape was probably created incorrectly or was permanently damaged.

**<FIB error prefix> REOPEN STOPPED BY TAPEMANAGER
<rcw> (MSRFIB268)**

- A process attempted to open a tape file that was previously opened and closed with retention. However, the TAPEMANAGER system library specified that this open request should be terminated with an open error.
- Contact your site's support personnel or the developer of your TAPEMANAGER library to determine why the request was denied. See Section 5, "TAPEMANAGER Library Implementation," for more information on TAPEMANAGER libraries.

<file name> REQUIRES <unit identifier> (MSRFOUT30)

- The MCP is unable to perform a requested file assignment because it cannot locate a unit that matches the selection attributes of the named output file. The system waits for your response.
- Enter one of the following system commands:
 - <mix number> OK
Attempts again to locate a unit for file assignment.
 - <mix number> FA <file attribute> = <value>
Updates the specified attribute value for the file and then attempts again to locate a unit for file assignment.
 - <mix number> OU <unit number>
Directs output from the task to the specified unit.
 - <mix number> DS
Terminates the process.

MT<unit number> REQUIRES VOLUME DELETE (MSRDPC93)

- You issued an SN (Serial Number) system command in the following form:

SN MT<unit number> [<serial number>]

The system rejected this command because changing the serial numbers of volumed tapes – that is, tapes listed in the volume library – with the SN command is disallowed. The tape volume that is to be purged is listed in the volume library under its current serial number. The tape is not purged.
- Determine if the serial number of the selected volume must be changed. If so, enter a WFL *VOLUME DELETE* statement for the selected volume. Then reissue the SN command. After the SN command completes, enter a WFL *VOLUME ADD* statement for the new scratch tape family so that it is a volumed tape.

RESTART ABORTED: TAPE LABELKIND CONFLICTS WITH USAGE (MSRRS26)

- This error occurs during a RERUN operation from a checkpoint. The assigned tape volume has a LABELKIND value that indicates that it is structured for use only with A Series system procedures, such as library maintenance. The system prevents user programs from altering the data on such system tapes. (User programs can erase and write ordinary data files on system tapes, but the programs must open the file as a new file starting at the beginning of the tape volume.)

The system closes the tape and discontinues the process. As a result of this action, that tape volume is likely to be unusable as input to system programs.

- Locate the tape volume that was used by the program when the checkpoint was originally taken, place it online, and rerun the job.

MT<unit number> SCRATCHPOOL REJECTED BY TAPEMANAGER (MSRDPC183)

- During the execution of a PG (Purge) or SN (Serial Number) system command, the TAPEMANAGER library rejected the scratchpool value for the indicated unit. The message includes the unit number if that value is available. The tape is not purged.
- Contact your site's support personnel or the developer of your TAPEMANAGER library to determine why the request was denied. See Section 5, "TAPEMANAGER Library Implementation," for more information on TAPEMANAGER libraries.

MT<unit number> SECURITYLABELS REQUIRE ANSI87 LABELS BUT OPTION ISNT SET (MSRSOFT42)

- A new tape volume is being created on the tape that is mounted on the indicated unit. The SECURITYLABELS volume attribute is set for the tape. However, neither of the following specifications has been used to specify that ANSI87 labels are to be written:
 - LABELFORMAT system option
 - LABELFORMAT task attribute

The system creates the volume with ANSI87 labels.

- If ANSI87 labels are acceptable, no action is required. If ANSI87 labels are not acceptable, reset the SECURITYLABELS volume attribute and re-create the tape.

**MT<unit number> SERIAL NUMBER REJECTED BY
TAPEMANAGER (MSRDPC181)**

- During the execution of a PG (Purge) or SN (Serial Number) system command, the TAPEMANAGER library rejected the serial number value for the indicated unit. The message includes the unit number if that value is available. The tape is not purged.
- Contact your site's support personnel or the developer of your TAPEMANAGER library to determine why the request was denied. See Section 5, "TAPEMANAGER Library Implementation," for more information on TAPEMANAGER libraries.

**MT<unit number> SERIALNO MATCHES BUT TAPE
IS A LIBRARY/MAINTENANCE TAPE <file name>
<volume description> (MSRCAT26)**

- This warning message can appear while the system is processing a WFL *CATALOG ADD* statement. The system located an online tape with a serial number that matches the serial number specified in the statement, but the online tape is a library maintenance tape. Only tape files on tapes that are other than library maintenance tapes can be marked as cataloged. The system continues looking for another online tape that properly matches the request.
- If the system is unable to locate another online tape that matches the request, verify that the statement was issued with the correct serial number, tape name, cycle, and version specifications. If you specified them incorrectly, reissue the request with the correct values. If you specified them correctly, verify that the correct tape volume is online and resubmit the *CATALOG ADD* request.

**MT<unit number> SERIALNO MATCHES BUT TAPE IS NOT
FIRST REEL <file name> <volume description> (MSRCAT27)**

- This message can appear while the system processes a WFL *CATALOG ADD* statement. The system located an online tape with a serial number that matches the serial number specified in the statement, but the online tape belongs to a multivolume tape set and the online tape is other than the first tape of that set. The system continues looking for another online tape that properly matches the request.
- If the system is unable to locate another online tape that matches the request, verify that you issued the statement with the correct serial number, tape name, cycle, and version specifications. If you specified them incorrectly, resubmit the request with the correct values. If you specified them correctly, verify that the correct tape volume is online and resubmit the *CATALOG ADD* request.

**<unit number> SERIALNO MATCHES BUT TAPE IS NOT LISTED
IN VOLUME LIBRARY <file name> <volume description>
(MSRCAT9)**

- The message can appear while the system processes a *WFL CATALOG ADD* statement. The system has located an online tape with a serial number that matches the serial number specified in the request, but that tape volume has no entry in the system's volume library. The system continues looking for another online tape that matches the name and serial number in the *CATALOG ADD* statement and that has an entry in the volume library.
- If the system is unable to locate another online tape that matches the request, verify that you issued the *CATALOG ADD* statement with the correct serial number, tape name, cycle, and version specifications. If you specified them incorrectly, resubmit the request with the correct values. If you specified them correctly, verify that the correct tape volume is online and, if necessary, submit a *WFL VOLUME ADD* statement for that tape volume. Then resubmit the *CATALOG ADD* request.

**MT<unit number> SERIALNO MISMATCH, 'COPY & VERIFY'
TERMINATED (MSRLM85)**

- The system was unable to identify the tape volume that the library maintenance procedure was verifying as a result of the *VERIFY* option. After copying files to the tape, the library maintenance procedure rewound the tape to read and verify the copied files. However, the serial number on the tape volume label does not match the serial number of the volume to which the files were written. As a result, the system is unable to verify the files for the specified destination.
- Determine whether one of the following situations caused the mismatch:
 - An I/O error occurred during the rewind operation.
 - An I/O error occurred when the label was read.
 - The tape volume was manually unloaded and replaced between the copy and verify phases.

Decide if the copy and verify process should be repeated.

SOURCE TAPE <tape name> MISSING (MSRLM90)

- The library maintenance procedure failed to open the named source tape for input because one of the following situations exists:
 - You replied to the "NO FILE" RSVP request with the *OF* (Optional File) system command.
 - An error occurred while the tape directory file on the tape was being opened. The utility proceeds to copy from other sources, if any.

- Determine why the utility was unable to correctly open the source tape. If necessary, resubmit the copy request with the correct input tapes.

**TAPE ABANDON FILEHANDLER, TYPE= <action> <file name>
<file name> (MSRDIR1)**

The variable <action> is one of the following:

AUDIT RECOVER
ENTER
ENTER BACKUP
PDSEARCH
REMOVE
SEARCH
TIMESTAMP
UPDATE

- An error occurred during an attempt to perform the specified operation on the catalog record for a cataloged tape file on a cataloging system. The system discontinues the operation and returns an error indication to the calling program.
- Note the error and correlate it with preceding directory error messages. If a particular program repeatedly encounters directory errors, consider discontinuing the program with the DS (Discontinue) system command.

**MT<unit number> TAPE DIRECTORY COMPARE ERROR
(MSRLM17)**

- During the execution of a WFL *COPY* or *ARCHIVE* statement, the system found that the tape directory read from the specified destination differs from the information that was written to the tape. The library maintenance procedure discontinues copying files to that destination tape.
- Resubmit the COPY or ARCHIVE statement with a different tape drive or a different tape volume.

'<host name>' TAPE DIRECTORY READ ERROR (MSRLM0)

- An error occurred while the library maintenance procedure was sending the list of file names to be copied from one host to another. The system copies no files for this request.
- Resubmit the request.

MT<unit number> TAPE DIRECTORY READ ERROR (MSRLM0)

- An I/O error occurred while the library maintenance procedure was reading the list of file names from the tape volume. If the tape is an input tape, the procedure discontinues copying files. If the tape is an output tape and the read error occurred while the system was comparing directories, the procedure discontinues copying files to that tape.

In most cases, the procedure discontinues copying files to the tape, but if the failing tape was started after a reel switch, the beginning of the tape contains the last part of the file that spanned two reels.

- Either resubmit the *WFL COPY* or *ARCHIVE* statement, or move the tape to another tape drive and then resubmit the statement.

'<host name>' TAPE DIRECTORY WRITE ERROR (MSRLM1)

- The library maintenance procedure was unable to copy files to another host system because an error occurred when the procedure was writing the tape directory. The procedure discontinues copying files for this request.
- Resubmit the *WFL COPY* statement.

MT<unit number> TAPE DIRECTORY WRITE ERROR (MSRLM1)

- The library maintenance procedure was unable to copy a file to tape because an I/O error occurred when the procedure was sending the list of file names to the tape volume. The procedure discontinues copying files to that tape.
- Resubmit the *WFL COPY* or *ARCHIVE* statement with a different tape volume or tape drive.

'<host name>' TAPE DISKHEADER READ ERROR <file name> (MSRLM2)

- The library maintenance procedure was unable to copy a file from another host system because an error occurred while the procedure was receiving the disk file header for a file.
- Note the file name and resubmit the *WFL COPY* statement.

**MT<unit number> TAPE DISKHEADER READ ERROR
<file name> (MSRLM2)**

- The library maintenance procedure was unable to copy a file to or from the tape because an I/O error occurred while the procedure was reading the disk file header from the tape.
- Move the tape to another tape drive and resubmit the *WFL COPY* or *ARCHIVE* statement.

**MT<unit number> TAPE DISKHEADER WRITE ERROR
<file name> (MSRLM3)**

- The library maintenance procedure was unable to copy a file to or from tape because an I/O error occurred while the procedure was writing the disk file header to tape.
- Move the tape to another tape drive and resubmit the *WFL COPY* or *ARCHIVE* statement.

MT<unit number> TAPE FILE MISSING <file name> (MSRLM34)

- The library maintenance procedure was unable to locate the specified file on the input tape. The procedure discontinues copying files.
- If the input tape was created incorrectly or is damaged, the system is unable to copy the affected file. Otherwise, resubmit the request on a different tape drive.

**MT<unit number> TAPE POSITIONING ERROR <file name>
(MSRLM6)**

- An I/O error occurred in one of the following situations:
 - The library maintenance procedure was spacing an input tape forward to locate the next file to be copied.
In this case, the procedure continues to attempt copying files.
 - The library maintenance procedure has backspaced after backspacing a tape for a compare or recopy operation.
In this case, the procedure discontinues copying files to or from that tape.
- Resubmit the *WFL COPY* or *ARCHIVE* statement on a different tape drive or tape volume.

System Messages

**(<mix number>) TAPE READ <record type> <error code> ERROR
REC=<number> (MSREH0)**

The variable <record type> is one of the following:

CAT
FAST
FLAT
PAST
VAST
VOLLIB

The variable <error code> is one of the following:

CHECKSUM
HDRBLOCKLENGTH
HDRLOCATION
HDRMARKER
NOT READY
TITLE
UNKNOWN

- A hardware, software, or data error occurred while the system was reading a catalog for a cataloged tape file. The variable <number> indicates the record number of the directory record that encountered the error.
- Note the error so you can correlate it with any other problems.

MT<unit number> TAPE WILL BE PURGED (MSRLM81)

- The system displayed the RSVP request

MT<unit number> VERIFY ERROR; REPLY 'OF' TO CONTINUE WITH
NEXT FILE, 'FR' TO PURGE TAPE <file name>

In response, you entered a FR (Final Reel) system command in the following form:

<mix number>FR

- Resubmit the original WFL *COPY & VERIFY* statement with different tape units or tape volumes.

**(<mix number>) TAPE WRITE <record type> <error code>
ERROR REC=<number> (MSREH0)**

The variable <record type> is one of the following:

CAT
FAST
FLAT
PAST
VAST
VOLLIB
UNKNOWN

The variable <error code> is one of the following:

CHECKSUM
HDRBLOCKLENGTH
HDRLOCATION
HDRMARKER
NOT READY
TITLE
UNKNOWN

- A hardware, software, or data error occurred while the system was updating a catalog for a cataloged tape file. The variable <number> indicates the record number of the directory record that encountered the error.
- Note the error so you can correlate it with any other problems.

**TAPEMANAGER FAILED IN INITIALIZATION PROCEDURE
(MSRMISC214)**

- A SEND TM START request has been entered. However, the TAPEMANAGER library procedure CONTROL_INTERFACE encountered an error.
- Contact your site's support personnel or the developer of your TAPEMANAGER library to determine why the request was denied. See Section 5, "TAPEMANAGER Library Implementation," for more information on TAPEMANAGER libraries.

TAPEMANAGER FAILED IN TERMINATION PROCEDURE (MSRMISC215)

- A SEND TM QUIT system command has been entered. However, the TAPEMANAGER library procedure CONTROL_INTERFACE encountered an error.
- Contact your site's support personnel or the developer of your TAPEMANAGER library to determine why the request was denied. See Section 5, "TAPEMANAGER Library Implementation," for more information on TAPEMANAGER libraries.

TAPEMANAGER IS ACTIVE (MSRMISC207)

- The system has activated the TAPEMANAGER in response to a SEND TM START system command.
- No action is required.

TAPEMANAGER IS INACTIVE (MSRMISC211)

- The system has deactivated the TAPEMANAGER in response to a SEND TM QUIT system command.

After displaying this message, the system displays the following message:

TAPEMANAGER LIBRARY WILL NOT BE USED

This message is described in its respective alphabetical position among the other message descriptions in this appendix.

- No action is required.

TAPEMANAGER LIBRARY HAS LINKED (MSRMISC206)

- The system has successfully linked to the TAPEMANAGER library.
- No action is required.

TAPEMANAGER LIBRARY IS ALREADY ACTIVE (MSRMISC222)

- The system has rejected a SEND TM START system command because the current TAPEMANAGER library is already active. Previously, however, the status of the library had been INACTIVE.
- Contact your site's support personnel or the developer of your TAPEMANAGER library to correct the problem. See Section 5, "TAPEMANAGER Library Implementation," for more information on TAPEMANAGER libraries.

TAPEMANAGER LIBRARY IS MISSING CONTROL_INTERFACE PROCEDURE (MSRMISC210)

- This message is preceded by the following message:

WARNING: TAPEMANAGER LIBRARY IS MISSING ONE OR MORE OBJECTS

The third message to appear in the sequence is the following:

TAPEMANAGER LIBRARY WILL NOT BE USED

Descriptions of the additional messages appear in their respective alphabetical positions among the other descriptions in this appendix.

A SEND TM START system command has been entered. However, the MCP has determined that the CONTROL_INTERFACE procedure and perhaps other standard TAPEMANAGER procedures have not been exported from the TAPEMANAGER library.

Startup of the TAPEMANAGER library is stopped.

- Determine whether the proper code file was named in the SL (Support Library) system command that was used to install the TAPEMANAGER library. If necessary, reissue the SL command with the correct code file name and resubmit the SEND TM START request. Check with the developer of the TAPEMANAGER library to verify that the proper entry points have been exported. See Section 5, "TAPEMANAGER Library Implementation," for more information on TAPEMANAGER libraries.

TAPEMANAGER LIBRARY WILL NOT BE USED (MSRMISC208)

- This message is preceded by the following messages:

WARNING: TAPEMANAGER LIBRARY IS MISSING ONE OR MORE OBJECTS
TAPEMANAGER IS MISSING CONTROL_INTERFACE PROCEDURE

Descriptions of these messages appear in their respective alphabetical positions among the other system message descriptions in this manual.

The `CONTROL_INTERFACE` procedure and perhaps other standard `TAPEMANAGER` procedures have not been exported from the `TAPEMANAGER` library. When `CONTROL_INTERFACE` is missing, the MCP makes no use of the `TAPEMANAGER` library.

- Determine whether the proper code file was named in the SL (Support Library) system command that was used to install the `TAPEMANAGER` library. If necessary, reissue the SL command with the correct code file name and resubmit the `SEND TM START` request. Check with the developer of the `TAPEMANAGER` library to verify that the proper entry points have been exported. See Section 5, "TAPEMANAGER Library Implementation," for more information on `TAPEMANAGER` libraries.

TAPEMANAGER REQUIRED (MSRMISC220)

- A task that requires interaction with the `TAPEMANAGER` library cannot continue because the `TAPEMANAGER` library is not currently active. The system waits for your response.
- Determine why the `TAPEMANAGER` library is not active. Then choose one of the following courses of action:
 - Initiate the library by entering a `SEND TM START` system command. Once the library is active, the waiting task continues automatically.
 - Enter one of the following system commands:
 - `<mix number> OK`
Retries the call to the `TAPEMANAGER` library. This option is likely to produce the same result.
 - `<mix number> OF`
Bypasses the call to the `TAPEMANAGER` library. Unisys does not recommend this option because it can also bypass security measures defined by your site.
 - `<mix number> DS`
Terminates the task.

MT<unit number> TITLE FOUND WAS: <file name> (MSRLM24)

- The disk file title that the library maintenance procedure located on the tape differs from the expected file title.
- If the tape was created correctly, resubmit the `WFL COPY` or `ARCHIVE` statement with a different tape drive.

**<unit identifier> TYPE DISAGREES WITH KIND, REQUIRES
<unit kind> (MSRFOUT23)**

- The system rejected a request to open a file on the indicated unit because the type of unit does not match the KIND specified for the task. The system waits for your response.
- Enter one of the following system commands:
 - <mix number> OK
Attempts again to open the file.
 - <mix number> FA <file attribute> = <value>
Assigns the specified attribute value to the file and then attempts to open the file.
 - <mix number> OU <unit number>
Directs output from the task to a unit of the required kind.
 - <mix number> DS
Terminates the process.

UNABLE TO LINK TO TAPEMANAGER LIBRARY (MSRMISC205)

- A SEND TM START system command has been entered, but the system is unable to link to the TAPEMANAGER library. Library linkage can fail if the TAPEMANAGER function is not defined, if the library code file is missing or is not a library, or for other reasons. The system stops the attempt to link to the TAPEMANAGER library.
- Determine whether the proper code file was named in the SL (Support Library) system command that was used to install the TAPEMANAGER library. If necessary, reissue the SL command with the correct code file name and resubmit the SEND TM START request. Check with the developer of the TAPEMANAGER library to verify that the proper entry points have been exported. See Section 5, "TAPEMANAGER Library Implementation," for more information on TAPEMANAGER libraries.

UNABLE TO LINK TO TAPEMANAGER LIBRARY. ENTER OK TO RETRY (MSRMISC204)

- A SEND TM START system command has been entered, but the system is unable to link to the TAPEMANAGER library. Library linkage can fail if the TAPEMANAGER function is not defined, if the library code file is missing or is not a library, or for other reasons. The system stops the attempt to link to the TAPEMANAGER library and waits for your response.
- Determine the cause of the error and decide whether corrective measures are possible. For example, you might be able to copy a corrected version of the library to its intended location.

Then enter one of the following system commands:

- <mix number> OK
Retries the linkage operation.
- <mix number> DS
Terminates the operation.

MT<unit number> UNEXPECTED TAPE MARK - NOT COPIED <file name> (MSRLM22)

- Library maintenance detected an end-of-file (EOF) marker in the middle of the data portion of the input tape file.
- If the tape was created correctly, resubmit the WFL COPY or ARCHIVE statement on a different tape drive.

UNIT NOT VOLUMED ON MT<unit number> TAPE<number> (MSRCAT11)

- A process attempted to mark a tape file as cataloged, but the volume library contains no listing for the volume on which the file is located.
- If the tape should be volumed, use the WFL VOLUME ADD statement to add the volume to the volume library.

MT<unit number> UNRESTRICT SOURCE AND REPLY OK TO COPY <file name> (MSRLM43)

- This request is preceded by the following message:

NEW SOURCE UNIT OR VOLUME IS RESTRICTED

This message is discussed in its respective alphabetical position among the other message descriptions in this appendix.

During a file copy operation from a tape, a reel switch occurred on the source file. The new file has been named in a RESTRICT (Restrict Unit or Volume) system command to indicate that it is a "hazardous" file. A hazardous file is one that has the potential to compromise the security of the system.

The system waits for your response.

- Take one of the following courses of action:
 - Discontinue the job by entering a DS (Discontinue) system command.
 - Remove the restriction on the source unit or volume. Then respond to the RSVP request with an OK (Reactivate) system command to continue the operation.

USER PROCESS FAULTED OR DSED WHILE IN TAPEMANAGER LIBRARY CODE (MSRMISC221)

- While a task was performing a tape operation – opening a tape file, purging a tape, or calling the programmatic interface to the TAPEMANAGER library, for example – one of the following situations arose:
 - A fault occurred.
 - The system discontinued the process in response to a DS (Discontinue) system command. The job remains in the mix display only if the TAPEMANAGER library was discontinued for one of reasons associated with the following history task attribute values:

HISTORYTYPE	HISTORYCAUSE	HISTORYCAUSE
4 (DSEDV)	1 (OPERATORCAUSEV)	2 (JUSTDSEDV)
4 (DSEDV)	1 (PROGRAMCAUSEV)	18 (DEATHINFAMILY)
4 (DSEDV)	1 (PROGRAMCAUSEV)	125 (DATABASEDIEDV)
4 (DSEDV)	1 (PROGRAMCAUSEV)	126 (LIBRARYDIEDV)

If the TAPEMANAGER library was discontinued for any of these reasons, the system then displays one of the following messages:

FAULT REASON = <fault number> <rcw>

HISTORY REASON = <reason>, CAUSE = <cause>, TYPE = <type> <rcw>

These messages are described in their respective alphabetic positions among the other message descriptions in this appendix.

The system waits for your response to the RSVP request.

- If the TAPEMANAGER library was discontinued for one of reasons just listed, contact your site's support personnel or the developer of your TAPEMANAGER library to correct the problem. See Section 5, "TAPEMANAGER Library Implementation," for more information on TAPEMANAGER libraries.

If the library was discontinued for another reason, enter one of the following system commands:

- <mix number> OK
Retries the operation.
- <mix number> OF
Bypasses the operation.
- <mix number> DS
Terminates the task.

**MT<unit number> VERIFY ERROR; REPLY 'OF' TO CONTINUE
WITH NEXT FILE, 'FR' TO PURGE TAPE <file name> (MSRLM79)**

- An error occurred during a library maintenance file copy and verify operation. A message describing the specific error preceded this message.

The system waits for your response.

Note: In many cases, an error that occurs during the verify phase is the result of an error in the copy phase.

If your installation performs cataloging, be aware that if the COPY statement included the BACKUP option, the catalog already contains entries for all the cataloged files that were copied. Even though an error occurred during the verify phase, and even if you purge the tape, the catalog still contains backup references to the tape for all the files that were copied.

- Enter one of the following system commands:
 - <mix number> FR
Terminates the copy and verify process to the failing tape destination. Copying to other destinations, if any, continues. The failed tape is automatically rewound and purged. Retry the original copy and verify process with different media or units.
 - <mix number> OF
Causes the library maintenance procedure to skip to the next file. Although the file reported in error remains on the tape, it was copied incorrectly to the tape.
 - <mix number> DS

Terminates the copy process immediately, leaving the invalid output tape intact. Subsequently, purge the tape with the PG (Purge) system command. Then reissue the COPY or VERIFY command with the & VERIFY clause with different media or units. You can also attempt to copy files from the tape even though some of the files might have been copied incorrectly to the tape.

**VOLUME DIRECTORY <vol dir extension> <family name>
[<serial number list>] <error condition>**

The variable <error condition> is one of the following:

DUPLICATE SERIAL NUMBER (MSRVS7)
 FAMILY AND NAME MISMATCH (MSRVS11)
 FAMILYOWNER CONFLICTS WITH TASK USERCODE (MSRVS21)
 NOT DONE--NOT PERMANENTLYOWNED WITH SECURITYLABELS (MSRVS23)
 NOT DONE--ONLY 1 SERIAL NUMBER WITH MATCHONLYSERIALNUMBER (MSRVS22)
 NOT DONE--SECURITYGUARD HAS TO BE A GUARDFILE (MSRVS20)
 REPEATED SERIAL NUMBER (MSRVS6)
 SERIAL NUMBERS NOT IN SAME FAMILY (MSRVS10)
 SERIALNO IS MISSING FROM DATA RECORD (MSRVS19)
 VOLUME IS OWNED BY ANOTHER USER (MSRVS18)
 VOLUME NAME CONFLICTS WITH VOLUME DIRECTORY ENTRY (MSRVS12)
 WON'T FIT IN VOLUME DIRECTORY (MSRVS9)

- The system encountered the indicated problem with the volume directory of the indicated family while attempting to perform the indicated action.
- Correct the indicated error condition and resubmit the request.

**VOLUME DIRECTORY MT<unit number> <vol dir extension>
<family name> [<serial number list>] <error condition>**

The variable <error condition> is one of the following:

INVALID SERIAL NUMBER (MSRVS17)
 LABELS CONFLICT WITH VOLUME DIRECTORY ENTRY (MSRVS13)
 MISSING SERIAL NUMBER (MSRVS8)
 SERIALNO IS MISSING FROM DATA RECORD (MSRVS19)
 VOLUME NAME CONFLICTS WITH VOLUME DIRECTORY ENTRY (MSRVS12)
 WARNING VOLUME NAME CONFLICTS WITH ONLINE VOLUME (MSRVS15)

- The system encountered the indicated problem with the volume directory of the indicated unit.
- Correct the indicated error condition and resubmit the request.

**VOLUME DIRECTORY MT<unit number> <family name>
[<serial number list>] DISK ERROR <operation>**

The variable <operation> is one of the following:

DISK ERROR READING DATA BLOCK (MSRVS4)
DISK ERROR READING KEY BLOCK (MSRVS2)
DISK ERROR WRITING DATA BLOCK (MSRVS5)
DISK ERROR WRITING KEY BLOCK (MSRVS3)

- A disk error occurred while the system was attempting to perform the indicated operation on the indicated unit.
- Correct the indicated error condition and resubmit the request.

**VOLUME DIRECTORY MT<unit number> <vol dir extension>
<family name> [<serial number list>] FAULT IN VSHANDLER
<rcw> (MSRVS1)**

- The operating system volume directory procedure VSHANDLER encountered a fault interrupt at the location in the MCP code indicated by the variable <rcw>.
- Note the error. If the problem persists, isolate the program or programs or the tape volume or volumes that encountered the problem. Notify your Unisys field engineer. Make a copy of the *SYSTEM/CATALOG/nnn* or *SYSTEM/ACCESS/nnn* file so that any problems caused by corrupted volume directory data structures can be investigated by the field engineer.

**VOLUME DIRECTORY MT<unit number> <vol dir extension>
<family name> [<serial number list>] RECORD HAS INVALID
DATA IN WORD <rcw> (MSRVS14)**

- While processing a record, the system encountered invalid data in the word at the location indicated by the variable <rcw>.
- Correct the invalid data at the indicated location.

**VOLUME DIRECTORY MT<unit number> <family name>
[<serial number list>] NOT SNED, REQUIRES VOLUME DELETE
(MSRVS16)**

- The system is unable to perform a requested SN (Serial Number) operation on the indicated tape volume directory.
- Verify that the unit number and the serial number are correct. Then specify the unit number in the following sequence of commands:
 - VOLUME DELETE
 - SN
 - VOLUME ADD

**VOLUME DIRECTORY VOLUME ADD <family name>
[<serial number or serial number list>] NOT DONE - NOT
PERMANENTLYOWNED WITH SECURITYLABELS (MSRVS23)**

- The system is unable to add a volume entry to the indicated volume directory as requested because the operation attempted to set the values of both the SECURITYLABELS and PERMANENTLYOWNED attributes to TRUE.
- Enter a request that specifies that either SECURITYLABELS or PERMANENTLYOWNED – but not both – is to be set to TRUE.

**VOLUME FAMILY <volume family> NOT DELETED (TAPE IN
USE) (MSRVL12)**

- A WFL process attempted to execute a VOLUME DELETE or VOLUME DESTROY statement for a tape volume that is in use. The system ignores the command and continues to execute the process issuing the request.
- Wait for the job using the tape to close the tape file. Then resubmit the request.

**MT<unit number> WARNING, TAPE WAS NOT WRITTEN WITH
CHECKSUMS <tape name> (MSRLM82)**

- A library maintenance process initiated in response to your use of the VERIFY option is copying files from a tape volume that was originally written without the VERIFY option selected. The process continues without checksum comparison tests of the input files.
- No action is required.

WARNING: TAPEMANAGER LIBRARY IS MISSING ONE OR MORE OBJECTS (MSRMISC209)

- A SEND TM START MCP command has been entered. However, the MCP has determined that one or more of the standard TAPEMANAGER procedures has not been exported from the TAPEMANAGER library.

If the CONTROL_INTERFACE procedure is missing, startup of the TAPEMANAGER library is stopped. In this case, the MCP follows this message with the following messages:

```
TAPEMANAGER IS MISSING CONTROL_INTERFACE PROCEDURE
TAPEMANAGER LIBRARY WILL NOT BE USED
```

Descriptions of the additional messages appear in their respective alphabetical positions among the other descriptions in this appendix.

If CONTROL_INTERFACE is missing, the MCP makes no use of the TAPEMANAGER library. If CONTROL_INTERFACE is present but one or more other procedures are missing, the MCP uses only those procedures that are present.

- Determine whether the proper code file was named in the SL (Support Library) system command that was used to install the TAPEMANAGER library. If necessary, reissue the SL command with the correct code file name and resubmit the SEND TM START request. Check with the developer of the TAPEMANAGER library to verify that the proper entry points have been exported. See Section 5, "TAPEMANAGER Library Implementation," for more information on TAPEMANAGER libraries.

WARNING 56: 'TAPE7', 'TAPE9', AND 'TAPEPE' ARE BEING DEIMPLEMENTED ON MARK 4.1. USE 'TAPE'. <rcw> (MSRWARNG56)

- As of the Mark 4.1 release of the A series software, any reference to TAPE7, TAPE9, or TAPEPE will produce a syntax error.
- No immediate action is required, but before you use the Mark 4.1 release of the software, Unisys recommends that you modify your programs to ensure that they specify TAPE rather than TAPE7, TAPE9, or TAPEPE, and that they use the DENSITY file attribute to specify the density of a tape when a tape density specification is required.

**WARNING 57: SPECIFYING TAPE DENSITY WILL LIMIT OUTPUT
UNIT SELECTION TO QUALIFYING DEVICES ON MARK 4.1.
<rcw> (MSRWARNG57)**

- As of the Mark 4.1 release of the A series software, the system will reject any attempt to select nonqualifying devices if the tape density is specified.
- No immediate action is required, but before you use the Mark 4.1 release of the software, Unisys recommends that you modify your programs to ensure that when a tape density is specified, an appropriate unit is available.

**WARNING 58: ON MARK 4.1, INTERROGATING KIND FOR A TAPE
DEVICE WILL ALWAYS RETURN 'TAPE'. <rcw> (MSRWARNG58)**

- As of the Mark 4.1 release of the A series software, the system will respond to any interrogation of the KIND value of a tape device with TAPE.
- No action is required.

**WARNING 59: ON MARK 4.1, THE 'RESOURCE' TASK
ATTRIB. WILL GOVERN TOTAL TAPE COUNT ONLY. <rcw>
(MSRWARNG59)**

- As of the Mark 4.1 release of the A series software, the system will reject any attempt to use the RESOURCE task attribute for any purpose other than to govern the total tape count.
- No immediate action is required, but before you use the Mark 4.1 release of the software, Unisys recommends that you modify your programs to ensure that they do not attempt to use the RESOURCE task attribute for any purpose other than to govern the total tape count.

Appendix B

Standard Tape Formats

This appendix illustrates the standard tape formats used on A Series systems.

Note: *The information in this appendix is based on Appendix G of the A Series I/O Subsystem Programming Guide.*

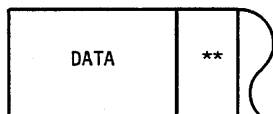
The figures in this appendix use the following notations:

Notation	Meaning
*	Tape mark
**	Double tape mark separated by an interrecord gap
EOF	End-of-file code
EOV	End-of-volume code
HDR	File header
RFE	Reserved for expansion
RFS	Reserved for standard
VOL	Volume header

Unlabeled Tapes

Figure B-1 illustrates the formats of an unlabeled single-file tape and an unlabeled multifile tape.

SINGLE-FILE VOLUME



MULTIFILE VOLUME

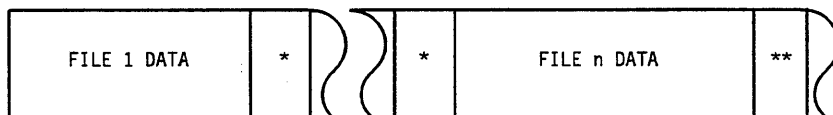


Figure B-1. Unlabeled Single-File-Volume and Unlabeled Multifile-Volume Formats

ANSI X3.27-1969 (ANSI69) Tapes

Figures B-2 through B-8 illustrate the formats of the following types of ANSI X3.27-1969 (ANSI69) tape labels:

- Single-file, single-volume tape
- Multivolume-file and multifile-volume tape
- Multifile, multivolume tape
- Volume header
- File header 1
- File header 2
- User header and trailer labels

Figure B-9 illustrates the format of an ANSI69 scratch tape.

ANSI69 tape labels are written in EBCDIC code.

SINGLE FILE, SINGLE VOLUME

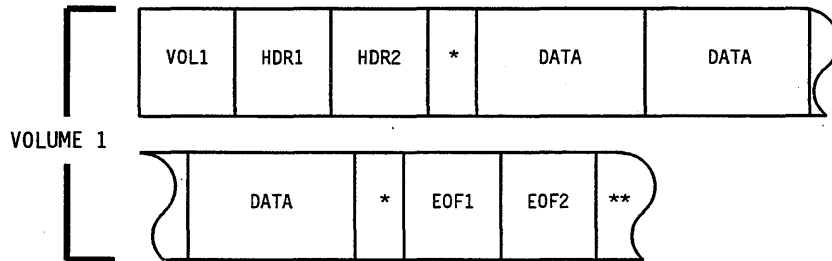
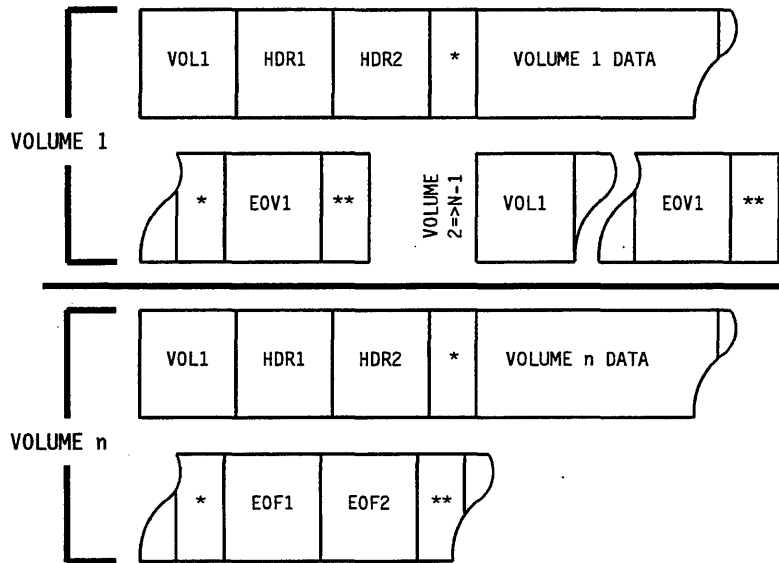


Figure B-2. ANSI69 Single-File, Single-Volume Format

MULTIVOLUME FILE



MULTIFILE VOLUME

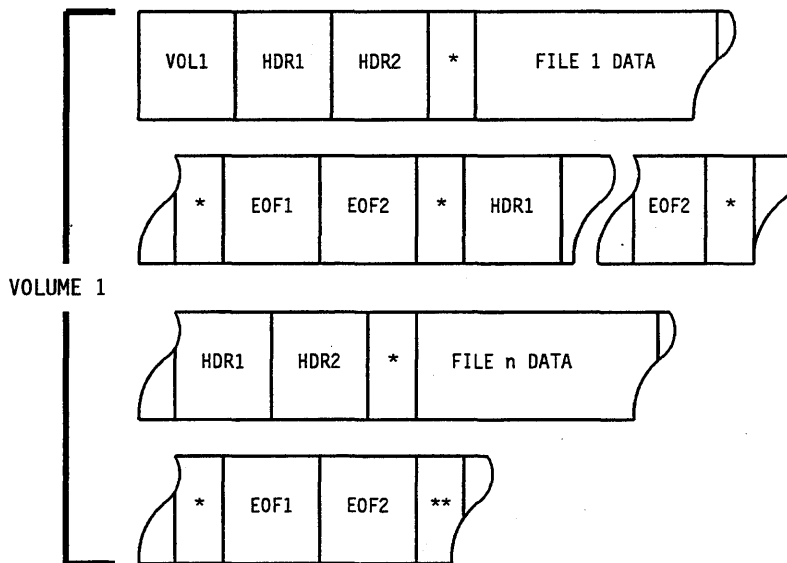


Figure B-3. ANSI69 Multivolume-File and Multifile-Volume Formats

Standard Tape Formats

MULTIFILE, MULTIVOLUME

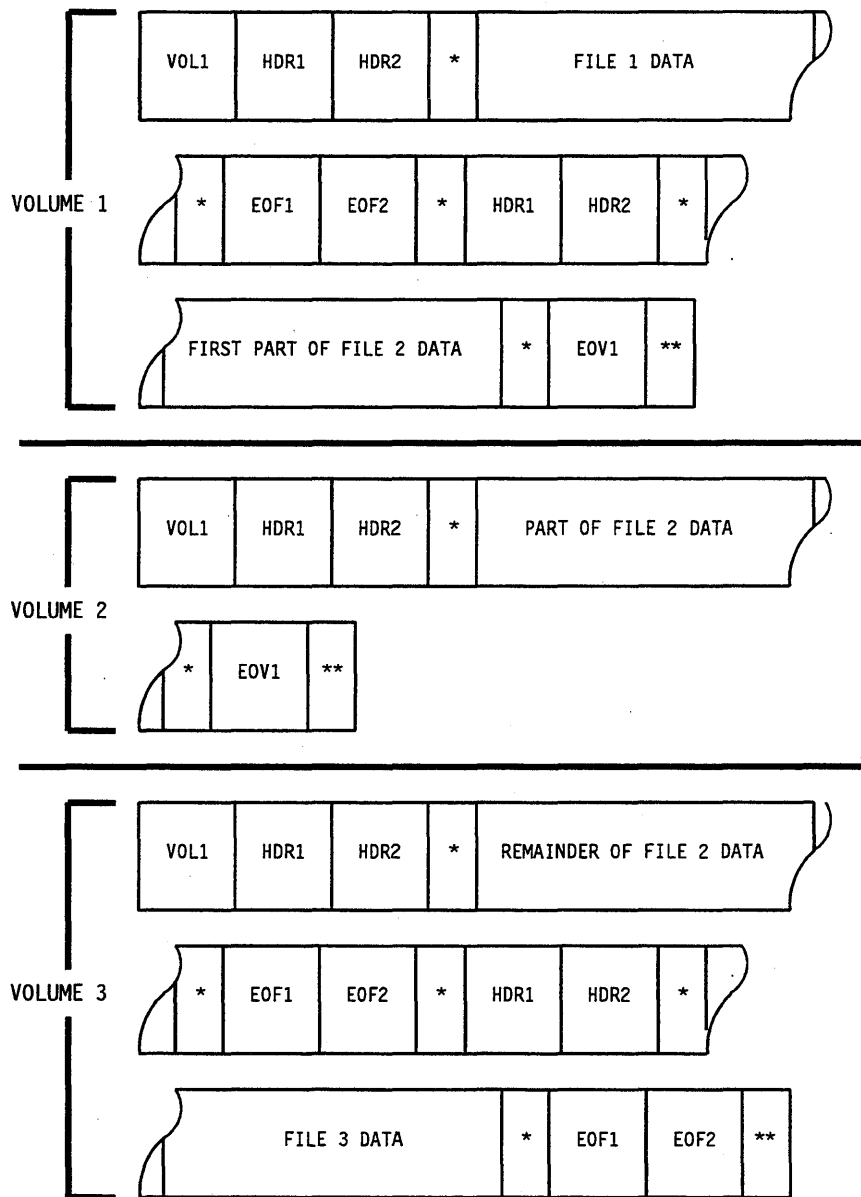


Figure B-4. ANSI69 Multifile, Multivolume Format

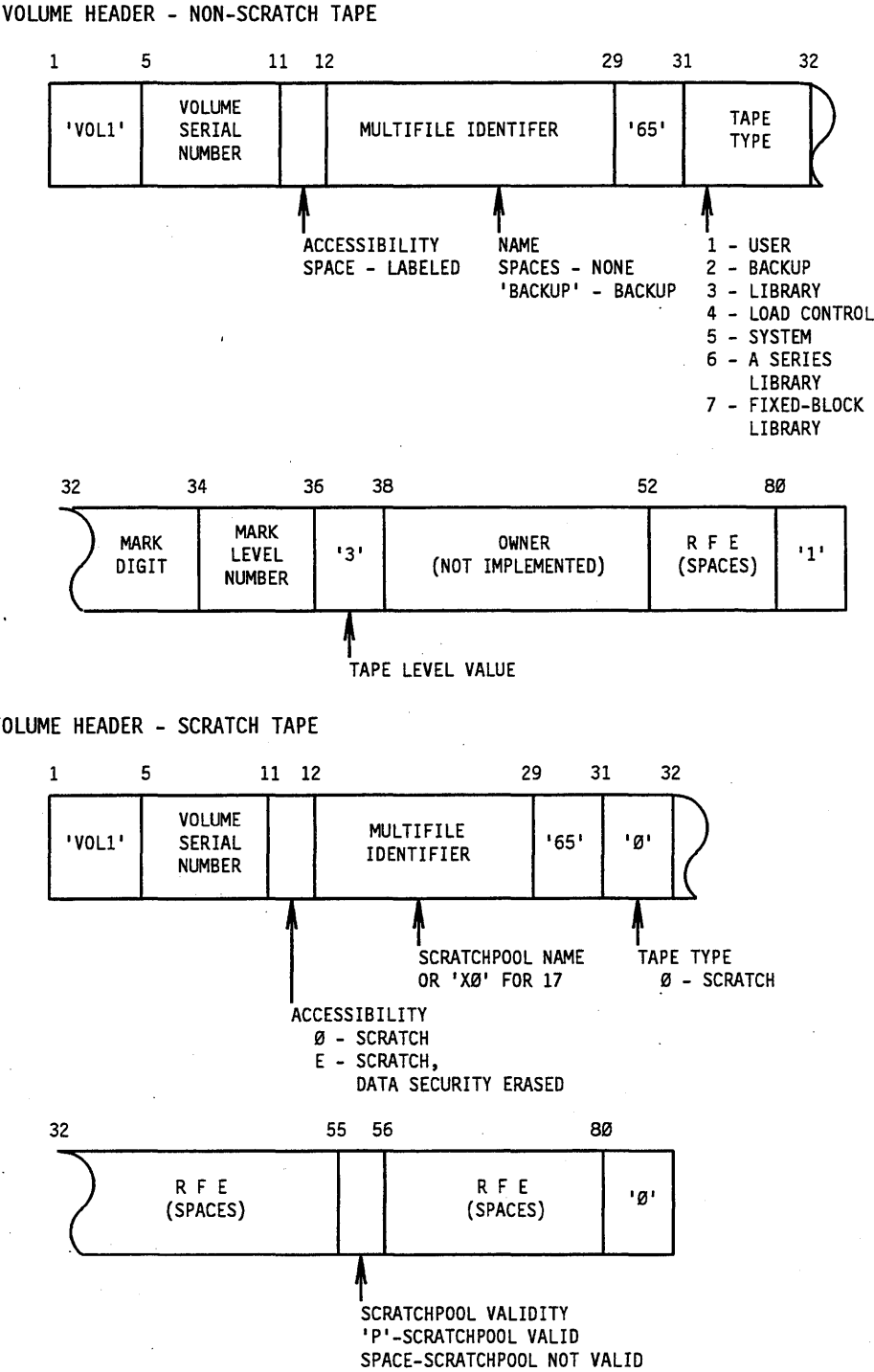


Figure B-5. ANSI69 Volume Header Formats

Standard Tape Formats

Notes:

- The MCP determines whether a tape is a scratch tape by examining characters 11 and 79 of the volume header: if character 11 is 0 or E and character 79 is 0, the MCP treats the tape as a scratch tape.
- The four characters of the mark digit and Mark level number fields combine to identify the Mark release level of the MCP that was active when the tape was created. For example, a tape created on a system running a 3.9 MCP stores the following values in characters 32 through 35 of its volume header:

Ø 3 Ø 9

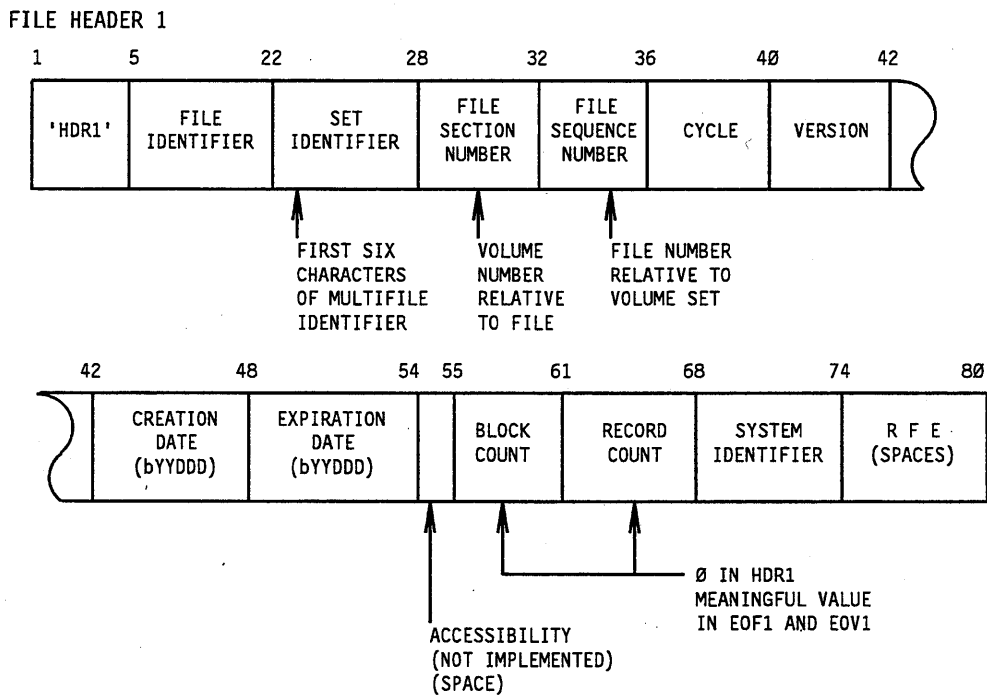


Figure B-6. ANSI69 File Header 1 Format

A first end-of-file label is the same as a first file header, except that the first four characters contain the letters *EOF1* and the block and record counts contain meaningful information.

An end-of-volume label is the same as the first end-of-file label, except that the first four characters contain the letters *EOV1*.

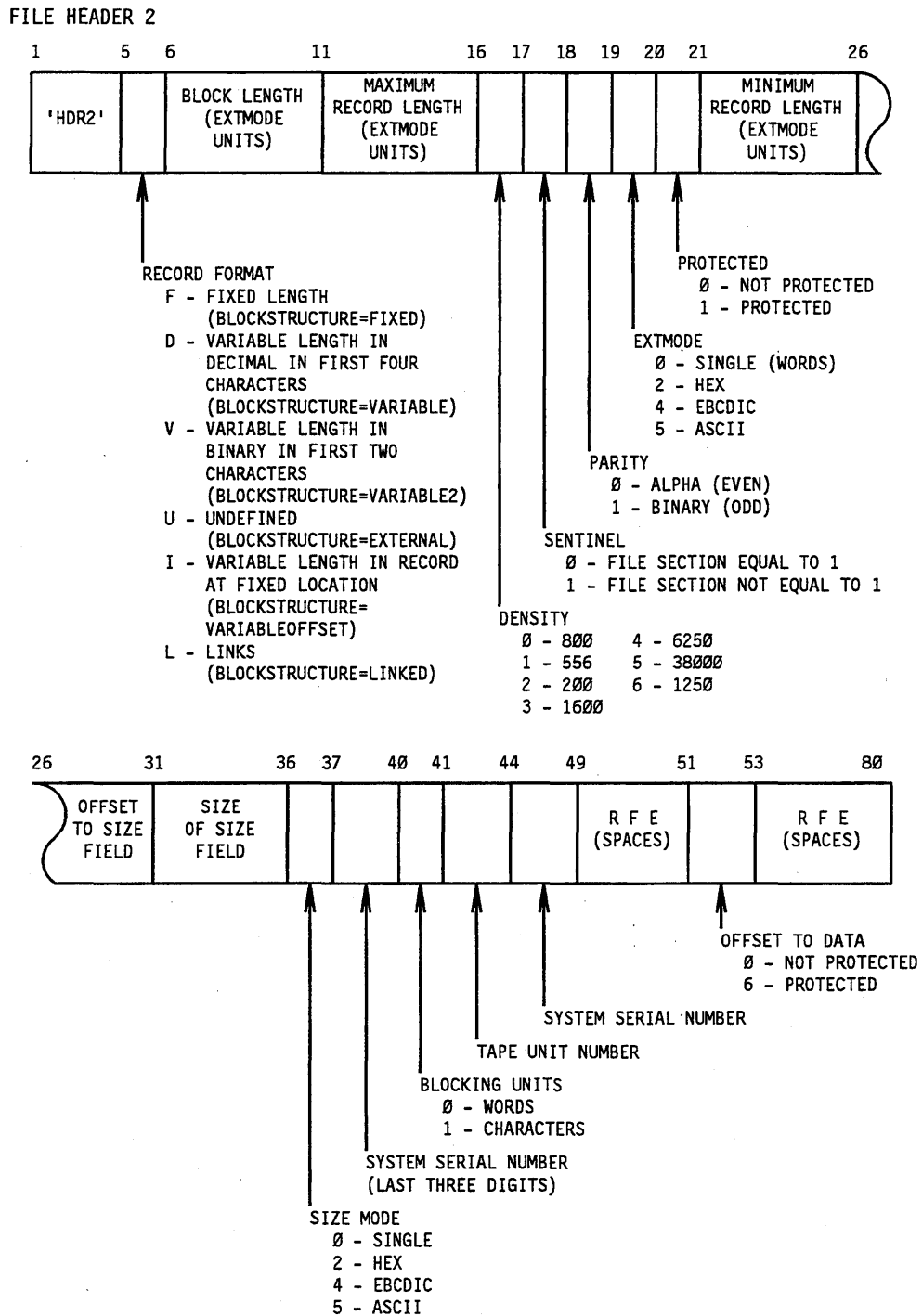


Figure B-7. ANSI69 File Header 2 Format

Standard Tape Formats

A second end-of-file label is the same as the second file header, except that the first four characters contain the letters EOF2.

Figure B-8 illustrates the formats of user header and trailer labels on ANSI69 tapes. User header labels can appear immediately after the characters HDR2 and before the tape mark, and user trailer labels can appear after either EOF2 or EOVI.

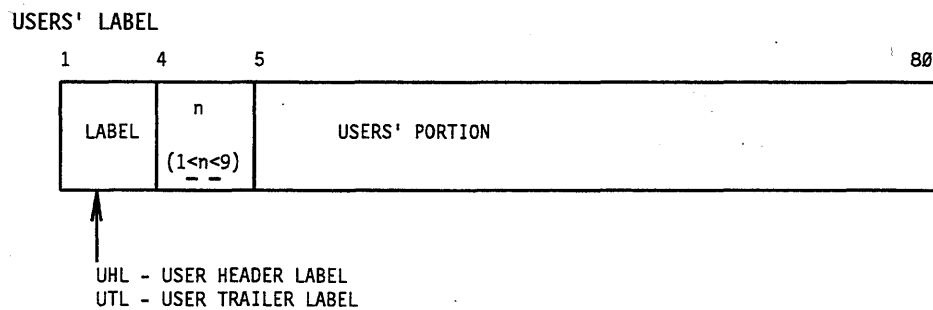


Figure B-8. ANSI69 User Header and Trailer Label Formats

Figure B-9 illustrates the format of an ANSI69 scratch tape.

ANSI69 SCRATCH TAPE FORMAT

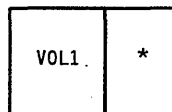


Figure B-9. ANSI69 Scratch Tape Format

ANSI X3.27-1987 (ANSI87) Tapes

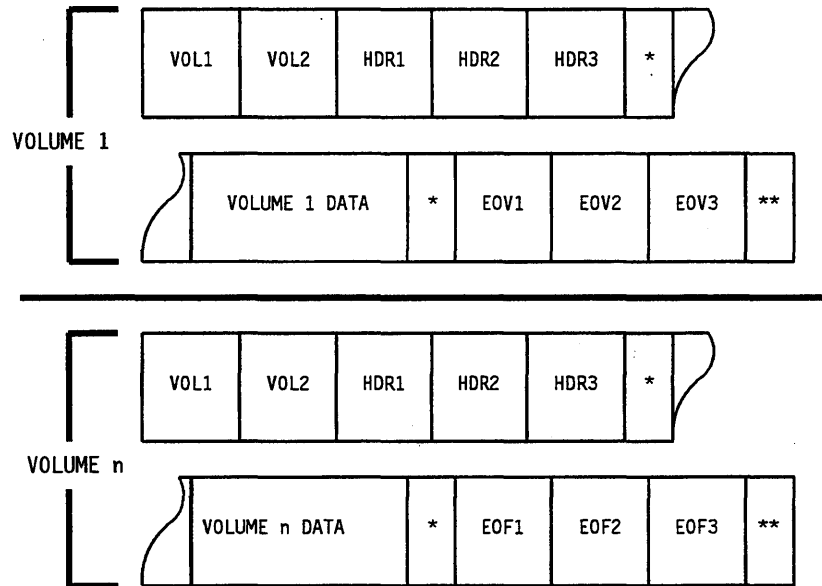
Figures B-10 through B-16 illustrate the formats of the following types of ANSI87 tape labels:

- Multivolume file
- Multifile volume
- Volume 1 header
- Volume 2 header
- Volume 3 header
- File 1 header
- File 2 header
- File 3 header

Figure B-17 illustrates the format of an ANSI87 scratch tape.

ANSI87 tape labels are written in ASCII code.

MULTIVOLUME FILE



MULTIFILE VOLUME

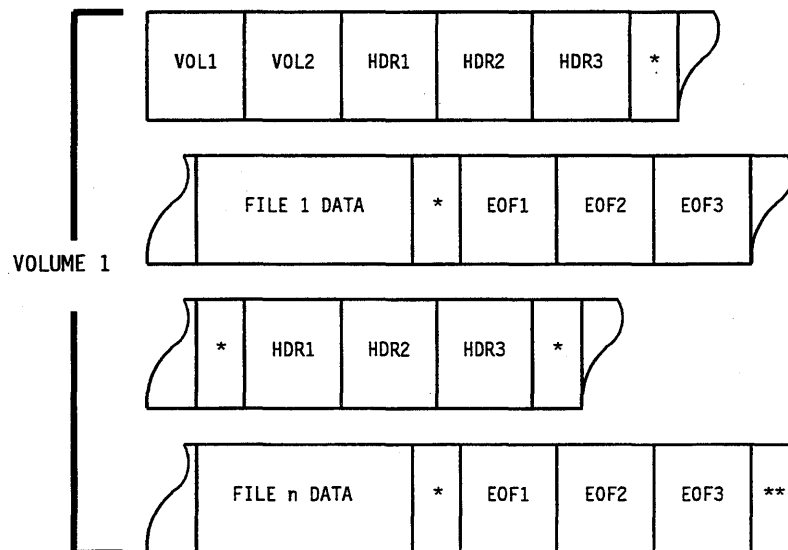


Figure B-10. ANSI87 Multivolume-File and Multifile-Volume Formats

Standard Tape Formats

MULTIFILE, MULTIVOLUME

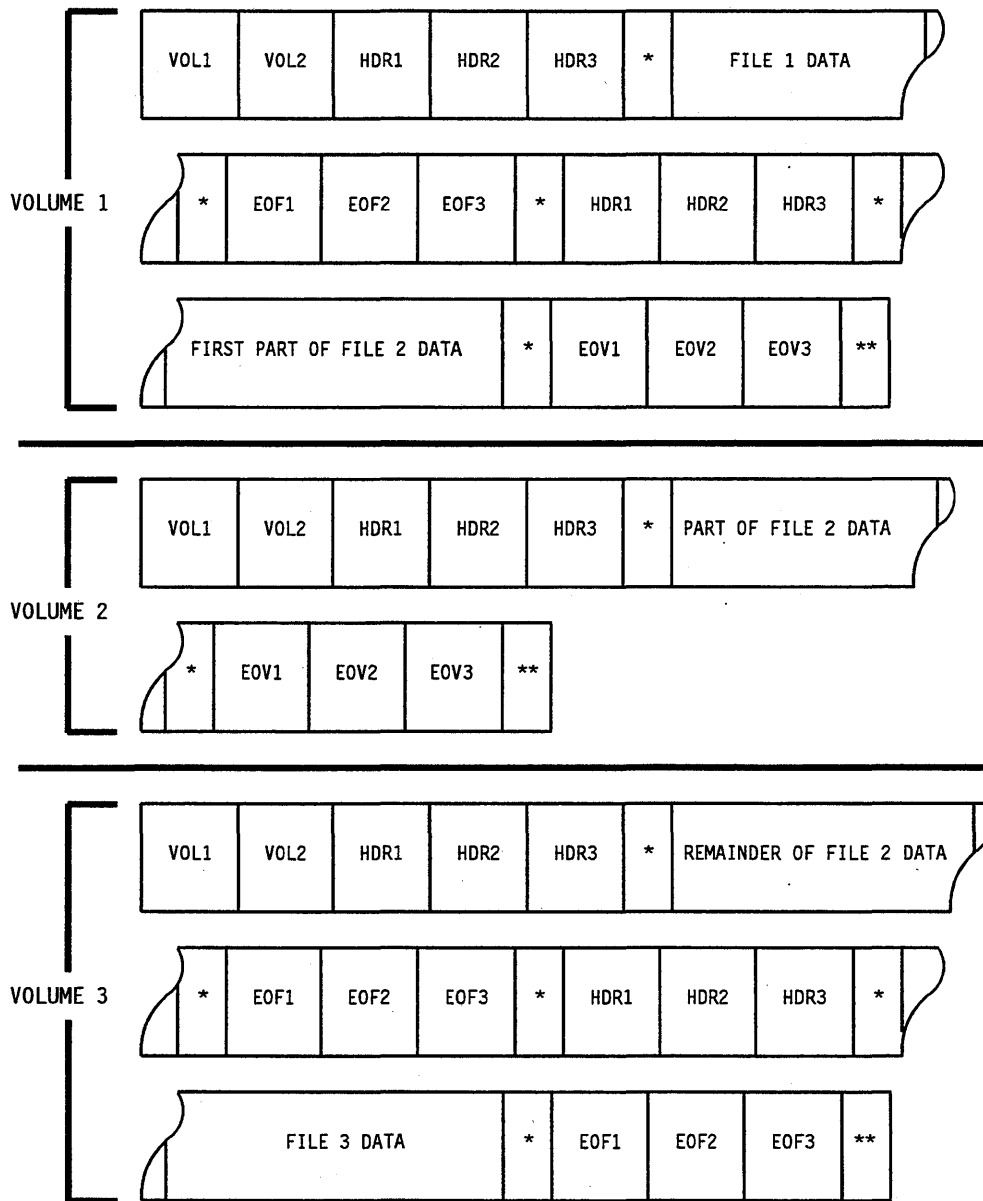


Figure B-11. ANSI87 Multifile, Multivolume Format

VOLUME 1 HEADER

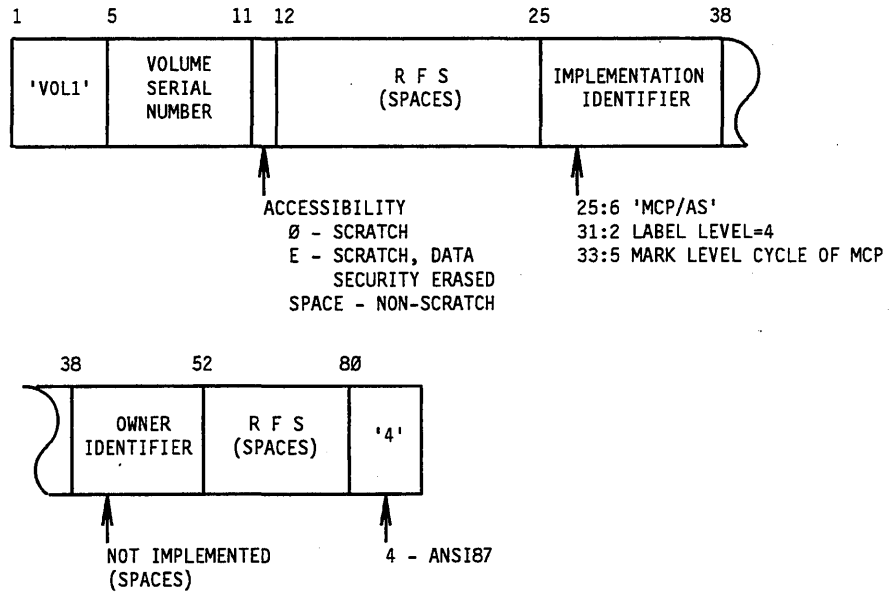


Figure B-12. ANSI87 Volume Header 1 Format

Notes:

- *If the FAMILY OWNER LENGTH field (characters 32 and 33) of the volume 2 header contains spaces or a zero, the MCP ignores the FAMILY OWNER field (characters 34–50).*
- *A tape includes a volume 3 header label if a nonzero value appears in the SECURITYGUARD LENGTH field (characters 51–53) of the volume 2 header. A volume 3 header label can vary in length from a minimum of 80 characters to a variable maximum number of characters that depends on the length of the SECURITYGUARD field. The length of the SECURITYGUARD field varies to accommodate the number of characters in the guard file title specified with the SECURITYGUARD file attribute.*

Standard Tape Formats

FILE HEADER 1

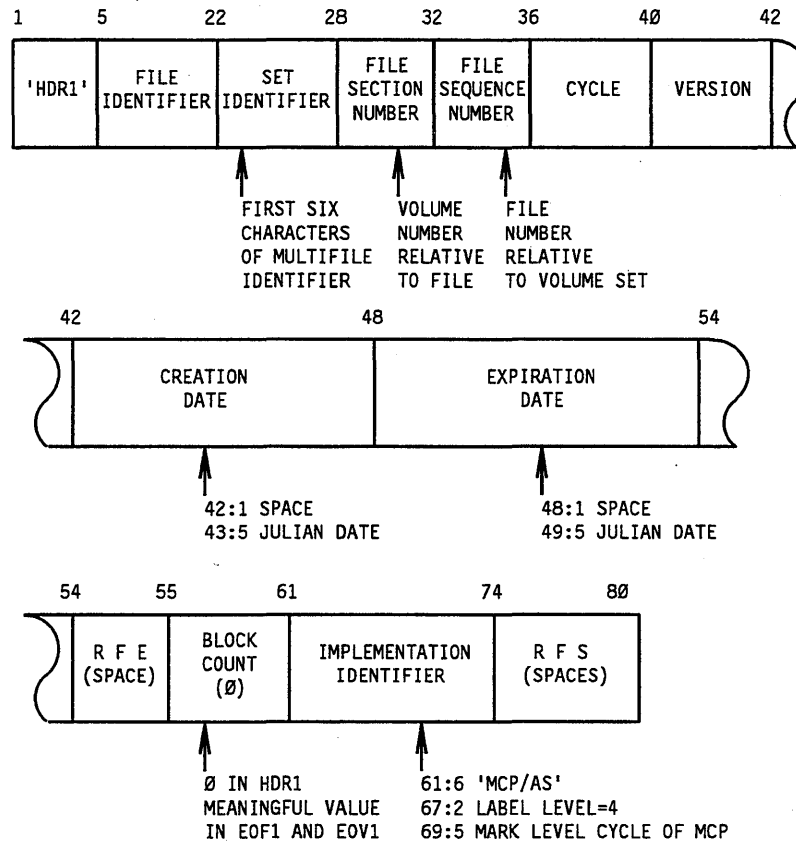
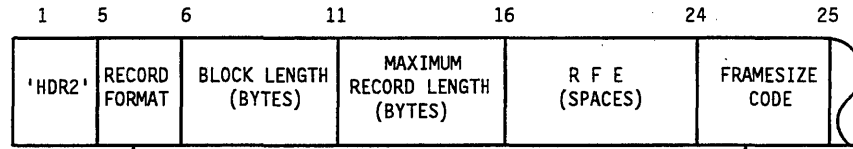


Figure B-14. ANSI87 File Header 1 Format

Notes:

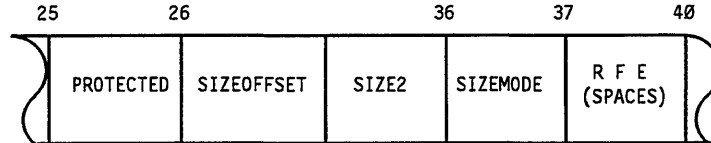
- A first end-of-file label is the same as a first file header, except that the first four characters contain the letters EOF1 and the block count contains meaningful information.
- A first end-of-volume label is the same as the first end-of-file label, except that the first four characters contain the letters EOVI.

FILE HEADER 2



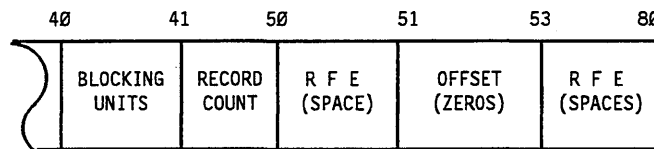
- F - FIXED LENGTH
(BLOCKSTRUCTURE=FIXED)
- D - VARIABLE LENGTH
INTMODE=ASCII and
UNITS=CHARACTERS
(BLOCKSTRUCTURE=VARIABLE)
- S - NOT IMPLEMENTED
- I - VARIABLE LENGTH
INTMODE=NOT ASCII or
UNITS=WORDS
(BLOCKSTRUCTURE=VARIABLE)
- V - VARIABLE LENGTH
(BLOCKSTRUCTURE=VARIABLE2)
- U - UNDEFINED
(BLOCKSTRUCTURE=EXTERNAL)
- I - VARIABLE LENGTH
(BLOCKSTRUCTURE=VARIABLEOFFSET)
- Z - VARIABLE LENGTH
(BLOCKSTRUCTURE=LINKED)

- 0 - 48
- 1 - 4
- 2 - 8



- 0 - NOT PROTECTED
- 1 - PROTECTED

MEANINGFUL ONLY IF
RECORD FORMAT = I
(BLOCKSTRUCTURE =
VARIABLEOFFSET)



- 0 - WORDS
 - 1 - CHARACTERS
- 0 IN HDR2
MEANINGFUL VALUE
IN EOF2 AND EOVS
- NOT IMPLEMENTED

Figure B-15. ANSI87 File Header 2 Format

Standard Tape Formats

Notes:

- A second end-of-file label is the same as a second file header, except that the first four characters contain the letters EOF2 and the record count contains meaningful information.
- A second end-of-volume label is the same as the second end-of-file label, except that the first four characters contain the letters EOV2.

FILE HEADER 3

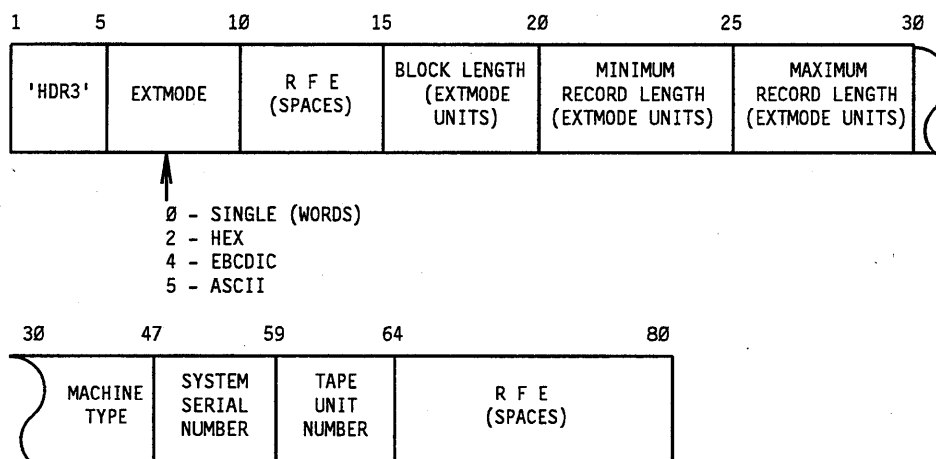


Figure B-16. ANSI87 File Header 3 Format

Note:

- A third end-of-file label is the same as a third file header, except that the first four characters contain the letters EOF3.
- A third end-of-volume label is the same as the third end-of-file label, except that the first four characters contain the letters EOV3.

Figure B-17 illustrates the format of an ANSI87 scratch tape.

ANSI87 SCRATCH TAPE FORMAT

VOL1	VOL2	**
------	------	----

Figure B-17. ANSI87 Scratch Tape Format

B 3500 USASI Tapes

Figures B-18 and B-19 illustrate the volume header format and the file header 1 format for B 3500 USASI tape labels.

VOLUME HEADER

1	5	11		80
'VOL1'	VOLUME SERIAL NUMBER	SPACES		

Figure B-18. B 3500 Volume Header Format

Standard Tape Formats

FILE HEADER 1

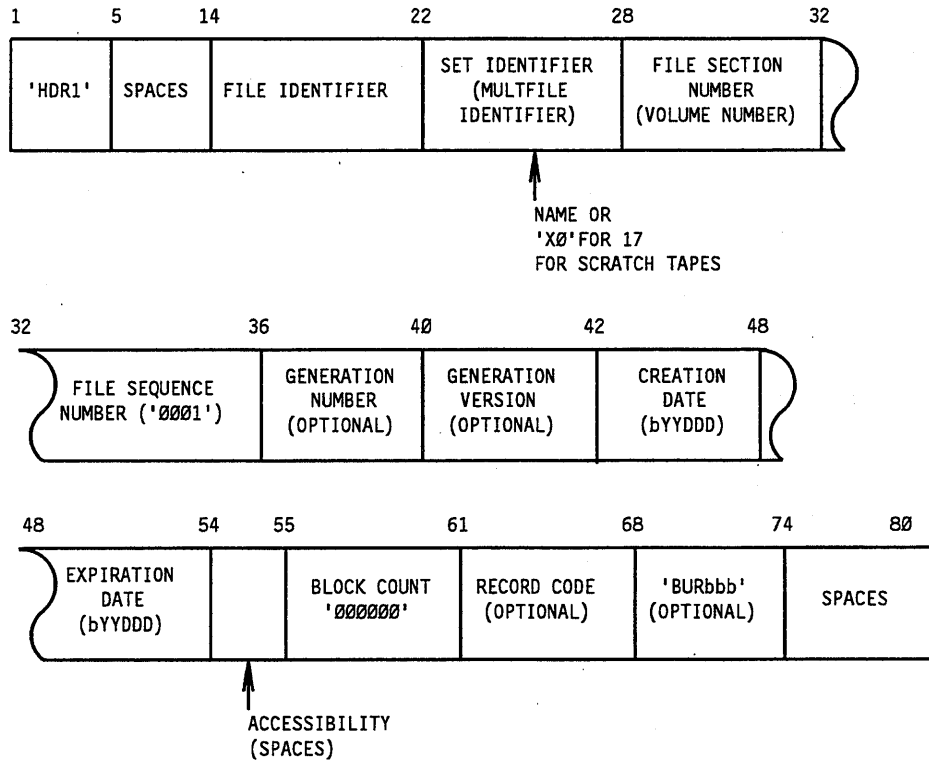


Figure B-19. B 3500 File Header 1 Format

B 5500 Tapes

Table B-1 describes the BCL characters that are common to all B 5500 tape labels.

Table B-1. BCL Characters for B 5500 Tape Labels

Word	Character (Word)	Character (Record)	Field (Description)
1	1-8	1-8	Must contain the word LABEL in uppercase letters
2	1	9	Must be 0
2	2-8	10-16	Multifile ID
3	1	17	Must be 0
3	2-8	18-24	File ID
4	1-3	25-27	Volume number (within file)
4	4-8	28-32	Date written (creation date)
5	1-2	33-34	Cycle number (to distinguish among identical runs on the same day)
5	3-7	35-39	Purge date (date this file can be destroyed)
5	8	40	Sentinel 1 = end-of-volume 0 = end-of-file
6	1-5	41-45	Block count
6-7	6-8 1-4	46-52	Record count
7	5	53	Memory dump key. If value is 1, the memory dump follows the label.
7-8	6-8 1-2	54-58	Physical tape number

ALGOL Files

Table B-2 describes the remaining information that is contained in a B 5500 tape label when the file is an ALGOL file.

Table B-2. B 5500 Tape Label for ALGOL Files

Word	Character (Word)	Character (Record)	Field (Description)
8	3	59	Blocking indicator 3 = blocked 0 = not blocked
8	4-8	60-64	Buffer size (number of words)
9	1-5	65-69	Maximum record size (number of words)
9	6-8	70-72	Zeros

COBOL Files

Table B-3 describes the remaining information that can be contained in a B 5500 tape label for a COBOL file.

Table B-3. B 5500 Tape Label for COBOL Files

Word	Character (Word)	Character (Record)	Field (Description)
8	3-8	59-64	Reserved for file-control routine. Not currently used.
9-end	1-end	65-end	Users' portion. Can be of any format desired by user and can be up to 8120 characters in length for tape files, up to 16 characters in length for card files, and up to 56 characters in length for printer files.

Understanding Railroad Diagrams

all consist of paths that represent the allowable structure, constants, and variables for commands and statements.

By following railroad diagrams, you can easily understand the correct syntax for commands and statements. Once you become proficient in the use of railroad notation, the diagrams serve as quick references to the commands and statements.

Constants and Variables

A constant is an item that cannot be altered. You must enter the constant as it appears in the diagram, either in full or as an allowable abbreviation. If a constant is partially underlined, you can abbreviate the constant by entering only the underlined letters. In addition to the underlined letters, any of the remaining letters can be entered. If no part of the constant is underlined, the constant cannot be abbreviated. Constants can be recognized by the fact that they are never enclosed in angle brackets (< >) and are in uppercase letters.

A variable is an item that represents data. You can replace the variable with data that meets the requirements of the particular command or statement. When replacing a variable with data, you must follow the rules defined for the particular command or statement. Variables appear in railroad diagrams enclosed in angle brackets.

In the following example, BEGIN and END are constants while <statement list> is a variable. The constant BEGIN can be abbreviated since it is partially underlined. Valid abbreviations for BEGIN are BE, BEG, and BEGI.

— BEGIN —<statement list>— END —————|

Constraints

Constraints are used in a railroad diagram to control progression through the diagram. Constraints consist of symbols and unique railroad diagram line paths. They include

- Vertical bars
- Percent signs
- Right arrows
- Required items
- User-selected items
- Loops
- Bridges

A description of each item follows.

Vertical Bar

The vertical bar symbol (|) represents the end of a railroad diagram and indicates the command or statement can be followed by another command or statement.

— SECONDWORD — (—<arithmetic expression>—) —————|

Percent Sign

The percent sign (%) represents the end of a railroad diagram and indicates the command or statement must be on a line by itself.

— STOP —————%

Right Arrow

The right arrow symbol (>) is used when the railroad diagram is too long to fit on one line and must continue on the next. A right arrow appears at the end of the first line, and another right arrow appears at the beginning of the next line.

— SCALERIGHT — (—<arithmetic expression>— , —————→
→<arithmetic expression>—) —————|

Required Items

A required item can be either a constant, a variable, or punctuation. A required item appears as a single entry, by itself or with other items, on a horizontal line. Required items can also exist on horizontal lines within alternate paths or nested (lower-level) diagrams. If the path you are following contains a required item, you must enter the item in the command or statement; the required item cannot be omitted.

In the following example, the word EVENT is a required constant and <identifier> is a required variable:

— EVENT —<identifier>—————|

User-Selected Items

User-selected items appear one below the other in a vertical list. You can choose any one of the items from the list. If the list also contains an empty path (solid line), none of the choices are required. A user-selected item can be either a constant, a variable, or punctuation. In the following railroad diagram, either the plus sign (+) or the minus sign (-) can be entered before the required variable <arithmetic expression>, or the symbols can be disregarded because the diagram also contains an empty path.

+
-

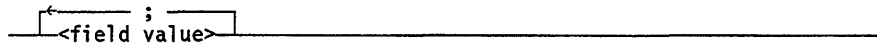
 <arithmetic expression>—————|

Understanding Railroad Diagrams

Loop

A loop represents an item or group of items that you can repeat. A loop can span all or part of a railroad diagram. It always consists of at least two horizontal lines, one below the other, connected on both sides by vertical lines. The top line is a right-to-left path that contains information about repeating the loop.

Some loops include a return character. A return character is a character – often a comma (,) or semicolon (;) – required before each repetition of a loop. If there is no return character, the items must be separated by one or more blank spaces.

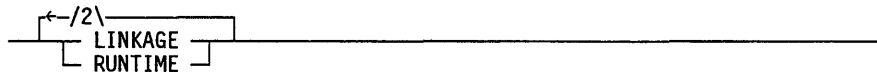
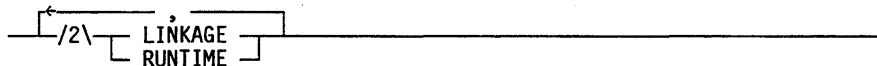


Bridge

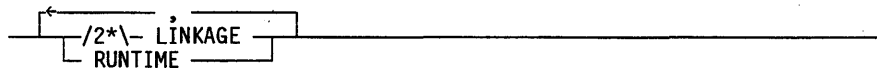
Sometimes a loop also includes a bridge, which is used to show the maximum number of times the loop can be repeated. The bridge can precede the contents of the loop, or it can precede the return character (if any) on the upper line of the loop.

The bridge determines the number of times you can cross that point in the diagram. The bridge is an integer enclosed in sloping lines (/ \). Not all loops have bridges. Those that do not can be repeated any number of times until all valid entries have been used.

In the first bridge example, you can enter LINKAGE or RUNTIME no more than two times. In the second bridge example, you can enter LINKAGE or RUNTIME no more than three times.



In some bridges an asterisk (*) follows the number. The asterisk means that you must cross that point in the diagram at least once. The maximum number of times that you can cross that point is indicated by the number in the bridge.



In the previous bridge example, you must enter LINKAGE at least once but no more than twice, and you can enter RUNTIME any number of times.

The following figure shows the types of constraints used in railroad diagrams.

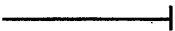

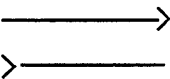
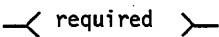
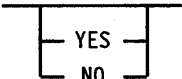
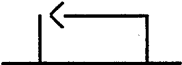
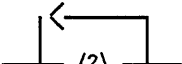
SYMBOL/PATH	EXPLANATION
	Vertical bar. Indicates that the command or statement can be followed by another command or statement.
	Percent sign. Indicates that the command or statement must be on a line by itself.
	Right arrow. Indicates that the diagram occupies more than one line.
	Required items. Indicates the constants, variables, and punctuation that must be entered in a command or statement.
	User-selected items. Indicates the items that appear one below the other in a vertical list. You select which item or items to include.
	A loop. Indicates an item or group of items that can be repeated.
	A bridge. Indicates the maximum number of times a loop can be repeated.

Figure C-1. Railroad Constraints

Following the Paths of a Railroad Diagram

The paths of a railroad diagram lead you through the command or statement from beginning to end. Some railroad diagrams have only one path, while others have several alternate paths. The following railroad diagram indicates there is only one path that requires the constant LINKAGE and the variable <linkage mnemonic>:

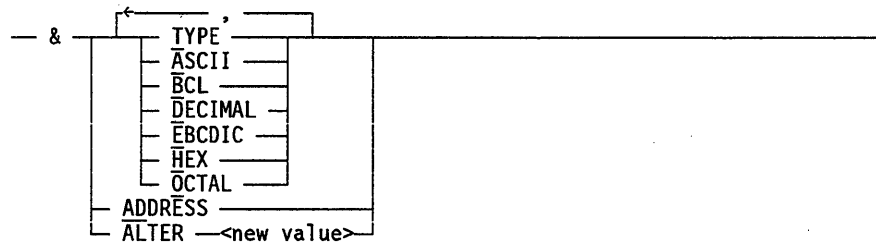
— LINKAGE —<linkage mnemonic>—————|

Alternate paths provide choices in the construction of commands and statements. Alternate paths are provided by loops, user-selected items, or a combination of both. More complex railroad diagrams can consist of many alternate paths, or nested (lower-level) diagrams, that show a further level of detail.

For example, the following railroad diagram consists of a top path and two alternate paths. The top path includes an ampersand (&) and the constants (that are

Understanding Railroad Diagrams

user-selected items) in the vertical list. These constants are within a loop that can be repeated any number of times until all options have been selected. The first alternate path requires the ampersand and the required constant ADDRESS. The second alternate path requires the ampersand followed by the required constant ALTER and the required variable <new value>.



Railroad Diagram Examples with Sample Input

The following examples show five railroad diagrams and possible command and statement constructions based on the paths of these diagrams.

Example 1

<lock statement>



Sample Input

LOCK (FILE4)

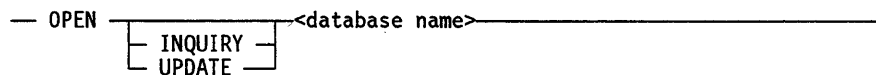
Explanation

LOCK is a constant and cannot be altered. Because no part of the word is underlined, the entire word must be entered.

The parentheses are required punctuation, and FILE4 is a sample file identifier.

Example 2

<open statement>



Sample Input

OPEN DATABASE1

Explanation

The constant OPEN is followed by the variable DATABASE1, which is a database name.

The railroad diagram shows two user-selected items, INQUIRY and UPDATE. However, because there is an empty path (solid line), these entries are not required.

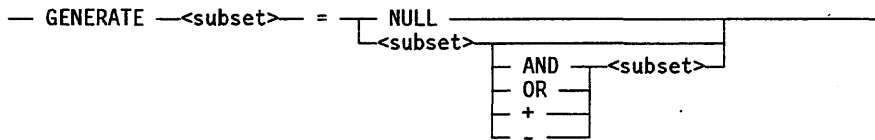
continued

continued

Sample Input	Explanation
OPEN INQUIRY DATABASE1	The constant OPEN is followed by the user-selected constant INQUIRY and the variable DATABASE1.
OPEN UPDATE DATABASE1	The constant OPEN is followed by the user-selected constant UPDATE and the variable DATABASE1.

Example 3

<generate statement>

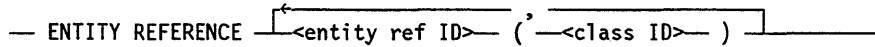


Sample Input	Explanation
GENERATE Z = NULL	The GENERATE constant is followed by the variable Z, an equal sign (=), and the user-selected constant NULL.
GENERATE Z = X	The GENERATE constant is followed by the variable Z, an equal sign, and the user-selected variable X.
GENERATE Z = X AND B	The GENERATE constant is followed by the variable Z, an equal sign, the user-selected variable X, the AND command (from the list of user-selected items in the nested path), and a third variable, B.
GENERATE Z = X + B	The GENERATE constant is followed by the variable Z, an equal sign, the user-selected variable X, the plus sign (from the list of user-selected items in the nested path), and a third variable, B.

Understanding Railroad Diagrams

Example 4

<entity reference declaration>



Sample Input

ENTITY REFERENCE ADVISOR1 (INSTRUCTOR)

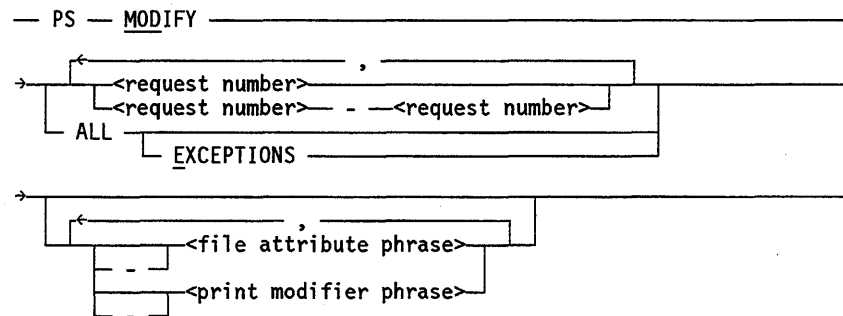
Explanation

The required item ENTITY REFERENCE is followed by the variable ADVISOR1 and the variable INSTRUCTOR. The parentheses are required.

ENTITY REFERENCE ADVISOR1 (INSTRUCTOR),
ADVISOR2 (ASST_INSTRUCTOR)

Because the diagram contains a loop, the pair of variables can be repeated any number of times.

Example 5



Sample Input

PS MODIFY 11159

Explanation

The constants PS and MODIFY are followed by the variable 11159, which is a request number.

PS MODIFY 11159,11160,11163

Because the diagram contains a loop, the variable 11159 can be followed by a comma, the variable 11160, another comma, and the final variable 11163.

PS MOD 11159-11161 DESTINATION =
"LP7"

The constants PS and MODIFY are followed by the user-selected variables 11159-11161, which are request numbers, and the user-selected variable DESTINATION = "LP7", which is a file attribute phrase. Note that the constant MODIFY has been abbreviated to its minimum allowable form.

PS MOD ALL EXCEPTIONS

The constants PS and MODIFY are followed by the user-selected constants ALL and EXCEPTIONS.

Glossary

A

ALGOL

Algorithmic language. A structured, high-level programming language that provides the basis for the stack architecture of the Unisys A Series systems. ALGOL was the first block-structured language developed in the 1960s and served as a basis for such languages as Pascal and Ada. It is still used extensively on A Series systems, primarily for systems programming.

ANSI

American National Standards Institute. A nongovernmental, nonprofit organization that is the central body for coordinating voluntary standards in the United States. ANSI also serves as the United States member of the International Standards Organization (ISO).

APL

A Programming Language. A procedure-oriented language that can produce very short but powerful programs.

automatic-unload mode

A mode of tape drive operation in which the system, after releasing the drive from use or performing a reel switch from the drive, rewinds and then automatically unloads the tape.

B

BCL

See Burroughs Common Language.

Burroughs Common Language (BCL)

An obsolete code using 6-bit character representation.

BNA

The network architecture used on A Series, B 1000, and V Series systems as well as CP9500 and CP 2000 communications processors to connect multiple, independent, compatible computer systems into a network for distributed processing and resource sharing.

Boolean

Pertaining to variables, data items, and attributes having a value of TRUE or FALSE.

C

CANDE

See Command and Edit.

Command and Edit (CANDE)

A time-sharing message control system (MCS) that enables a user to create and edit files, and to develop, test, and execute programs, interactively.

COBOL

Common Business-Oriented Language. A widely used, procedure-oriented language intended for use in solving problems in business data processing. The main characteristics of COBOL are the easy readability of programs and a considerable degree of machine independence. COBOL is the most widely used procedure-oriented language.

COBOL74

A version of the COBOL language that is compatible with the American National Standard X3.23-1974.

control file

(1) In Data Management System II (DMSII), a file containing data file coordination information, audit control information, and dynamic database parameter values. (2) In the Generalized Message Control System (GEMCOS), a file named CQUS, and known externally as DATACOM/QUEUE/CONTROL, that is created by the transaction control language (TCL) compiler and contains all tables and values for all parameters specified in the TCL.

D

DASDL

See Data and Structure Definition Language.

Data and Structure Definition Language (DASDL)

In Data Management System II (DMSII), the language used to describe a database logically and physically, and to specify criteria to ensure the integrity of data stored in the database. DASDL is the source language that is input to the DASDL compiler, which creates or updates the database description file from the input.

Data Management System II (DMSII)

A specialized system software package used to describe a database and maintain the relationships among the data elements in the database.

distributed systems service (DSS)

One of a collection of services provided on Unisys hosts to support communications across multihost networks. DSSs can be services such as file handling, station transfer, and mail transfer.

DMSII

See Data Management System II.

dump

(1) A copy of data in memory or a copy of a program on a tape. (2) A process that writes the contents of a specific area of memory to record the contents of a particular stage of processing. This process is used to determine the cause of a failure. (3) To copy a file or program on a tape. (4) In data or file management, a copy of a database on a tape.

DSS

See distributed systems service.

E

EBCDIC

Extended Binary Coded Decimal Interchange Code. An 8-bit code representing 256 graphic and control characters that are the native character set of most mainframe

EMS

See entry and medium systems.

end of file (EOF)

A code at the end of a data file that signals that the last record in the file has been processed.

end of task (EOT)

The termination of processing of a task.

entry and medium systems (EMS)

A designation referring to the Micro A and A 1 through A 10 systems.

EOF

See end of file.

EOT

See end of task.

F

file

A named group of related records.

file equation

A mechanism for specifying the values of file attributes when a program is compiled or executed. A file equation implicitly assigns a value to the FILECARDS task attribute.

File Transfer Protocol (FTP)

A protocol used to copy disk files between hosts connected across a Transmission Control Protocol/Internet Protocol (TCP/IP) network.

FTP

See File Transfer Protocol.

G

GCR

See group-coded recording.

group-coded recording (GCR)

A scheme for recording data on a magnetic tape (MT).

H

HDU

See host data unit.

host data unit (HDU)

The A 12 and A 15 system host interface to the I/O subsystem. An HDU is configured with up to three host-dependent ports (HDPs), each of which supports two message level interface (MLI) cables.

I

I/O

Input/output. An operation in which the system reads data from or writes data to a file on a peripheral device such as a disk drive.

L

LAN

See local area network.

local area network (LAN)

A network that enables high-speed communications among various devices within a relatively small area.

locked unit

Indicates that the unit is offline either because of an error that occurred during label processing or because certain operator commands such as PGL and SNL were entered.

M

MARC

See Menu-Assisted Resource Control.

master control program (MCP)

The central program of the A Series operating system. The term applies to any master control program that Unisys may release for A Series systems.

MCP

See master control program.

MCS

See message control system.

Menu-Assisted Resource Control (MARC)

A menu-driven interface to A Series systems that also enables direct entry of commands.

message control system (MCS)

A program that controls the flow of messages between terminals, application programs, and the operating system. MCS functions can include message routing, access control, audit and recovery, system management, and message formatting.

O**ODT**

See operator display terminal.

operator display terminal (ODT)

A terminal or other device that is connected to the system in such a way that it can communicate directly with the operating system. The ODT allows operations personnel to accomplish system operations functions through either of two operating modes: system command mode or data comm mode.

P**Pascal**

A high-level programming language developed by Niklaus Wirth, based on the block structuring nature of ALGOL 60 and the data structuring innovations of C.A.R. Hoare. Pascal is a general-purpose language.

purge

For magnetic tapes, an operation that writes a special label that contains the serial number of the tape and sets a flag that says no valid records are on the tape. The tape is not erased.

R**reel switch**

During the processing of a multivolume file, the traversal from one volume to the next.

reorganization

In Data Management System II (DMSII), the process of reordering or reformatting data sets, sets, or subsets. Reorganization can restore space in files, reorder data sets for more efficient retrieval, and reformat data set records when items are added, deleted, or changed.

resource management module (RMM)

A hardware module that interfaces with the I/O subsystem and schedules tasks on the E-mode processor (EMP) by way of a message protocol.

RMM

See resource management module.

RPG

Report Program Generator. A high-level, commercially oriented programming language used most frequently to produce reports based on information derived from data files.

S

scratch tape

A labeled magnetic tape (MT) whose label indicates that there are no files on the tape. Old data might remain on the tape, but this old data cannot be read unless the tape is read as an unlabeled tape. The old data present on a scratch tape is written over when new data is written to the tape.

Semantic Information Manager (SIM)

(1) The basis of the InfoExec™ environment. SIM is a database management system used to describe and maintain associations among data by means of subclass-superclass relationships and linking attributes. (2) In the Data Transfer System (DTS), a Unisys software product (SIM kernel) that DTS uses to extract data from host files.

SIM

See Semantic Information Manager.

SQLDB

See Structured Query Language Database.

Structured Query Language Database (SQLDB)

The A Series implementation of the American National Standards Institute (ANSI) Structured Query Language (SQL). SQLDB includes a data definition language (DDL) for describing a relational database, and a data manipulation language (DML) and a module language for querying a relational database.

substandard form name

A 1-byte binary count followed by the EBCDIC characters of the name.

T

tape file

A discrete collection of information on one or more tape volumes.

tape management system (TMS)

Any application that is designed to record or control the usage of tapes on a system.

tape volume

A physical reel of magnetic tape (MT) which can be labeled or unlabeled, and can contain one or more files, if it is not a scratch tape. A scratch tape volume or uninitialized new tape volume contains no accessible files.

TCP/IP

See Transmission Control Protocol/Internet Protocol.

TMS

See tape management system (TMS).

Transmission Control Protocol/Internet Protocol (TCP/IP)

(1) A family of protocols that were originally developed for use in a Department of Defense network, and which have been widely adopted as standard protocols for multivendor networks. The applications protocols typically supported by TCP/IP are File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), and Telnet.

(2) A Unisys A Series product that provides telecommunications capabilities between A Series, 1100 Series, U Series, and other computing systems. The A Series TCP/IP product is designed to interoperate with other TCP/IP systems in a multivendor network using industry and military standard protocols. The A Series TCP/IP product provides file transfer, station transfer, and (with the Mail System) mail transfer.

U**USASI**

United States of America Standards Institute. The former name of the American National Standards Institute (ANSI).

V**volume**

The medium of a mass storage device such as a disk, disk pack, or tape reel.

W**WFL**

See Work Flow Language (WFL).

WFL command

A Work Flow Language (WFL) statement entered as a command. The statements ADD, CHANGE, COPY, REMOVE, RUN, SECURITY, and START can be entered directly in the Action field of a MARC screen; any other WFL statement must be preceded by the characters *WFL* to be accepted as a WFL command.

Work Flow Language (WFL)

A Unisys language used for constructing jobs that compile or run programs on A Series systems. WFL includes variables, expressions, and flow-of-control statements that offer the programmer a wide range of capabilities with regard to task control.

Bibliography

- A Series ALGOL Programming Reference Manual, Volume 1: Basic Implementation* (8600 0098) Unisys Corporation.
- A Series Binder Programming Reference Manual* (8600 0304) Unisys Corporation.
- A Series CANDE Operations Reference Manual* (8600 1500). Unisys Corporation.
- A Series Disk Subsystem Administration and Operations Guide* (8600 0668). Unisys Corporation.
- A Series DMSII Data and Structure Definition Language (DASDL) Programming Reference Manual* (8600 0213). Unisys Corporation.
- A Series DMSII Technical Overview* (5044191). Unisys Corporation.
- A Series DMSII Utilities Operations Guide* (8600 0759). Unisys Corporation.
- A Series File Attributes Programming Reference Manual* (8600 0064). Unisys Corporation.
- A Series GETSTATUS/SETSTATUS Programming Reference Manual* (8600 0346). Unisys Corporation.
- A Series I/O Subsystem Programming Guide* (8600 0056). Unisys Corporation.
- A Series Report Program Generator (RPG) Programming Reference Manual, Volume 1: Basic Implementation* (8600 0544). Unisys Corporation
- A Series RoboHost A Operations, Installation, and Administration Guide* (8600 1740). Unisys Corporation.
- A Series RoboHost A Programming Guide* (8600 1609). Unisys Corporation.
- A Series Security Administration Guide* (8600 0973). Unisys Corporation.
- A Series Security Features Operations and Programming Guide* (8600 0528). Unisys Corporation.
- A Series System Commands Operations Reference Manual* (8600 0395). Unisys Corporation.
- A Series System Messages Support Reference Manual* (8600 0429). Unisys Corporation.
- A Series System Software Support Reference Manual* (8600 0478). Unisys Corporation.

Bibliography

A Series SYSTEMSTATUS Programming Reference Manual (8600 0452). Unisys Corporation.

A Series Task Attributes Programming Reference Manual (8600 0502). Unisys Corporation.

A Series Task Management Programming Guide (8600 0494). Unisys Corporation.

A Series Work Flow Language (WFL) Programming Reference Manual (8600 1047). Unisys Corporation.

Index

A

A Series

- security enhancement software (InfoGuard), 1-1, 1-8
 - support for user-created TMSs, 1-12, 5-1
 - tape formats, 1-1, 1-9, 3-7, B-1
 - tape management enhancements, 1-1
- account number, defining scratch pools by, 1-2
- ACTIVE status, TAPEMANAGER library, 5-14, 5-56

ADD

- command, CANDE, 3-47
- statement, Work Flow Language (WFL), 1-2, 1-3, 1-7, 3-47

ALGOL

- examples
 - file close operations and automatic tape unloading, 3-4
 - writing program output to scratchpools, 1-4
- files
 - B 5500 tape labels and, B-20
 - procedures exported by TAPEMANAGER library written in, 5-7, 5-9

- ALTERNATE option, VISIBLE DBS utility, 4-7

- ANALYZE option, SEND (Send Message) system command, 2-47

ANSI standards

- X3.27-1969 (ANSI69), B-2
- X3.27-1987 (ANSI87), 3-43, B-8

ANSI69 format

- file header 1, B-6
- file header 2, B-6
- multifile, multivolume tapes, B-3
- multivolume file tapes, B-2
- scratch tapes, B-6, B-8
- single-file, single-volume tapes, B-2
- tape labels, 1-9, 1-10, 3-8
- user header, B-8
- user trailer, B-8

- volume header, B-4

ANSI87 format

- file header 1, B-14
- file header 2, B-15
- file header 3, B-16
- multifile volume tapes, B-9
- multifile, multivolume tapes, B-10
- multivolume files, B-9
- scratch tapes, B-17
- tape labels, 1-9, 1-10, 3-8, 3-43, B-8
- volume header 1, B-11
- volume header 2, B-12, B-14
- volume header 3, B-12, B-14

- ARCHIVE BACKUP ENTRY DELETED system message, A-3

archive log

- creating, 2-27

- ARCHIVE statement, Work Flow Language (WFL), 1-2, 1-3, 3-45, 3-47

- archive tapes, defining scratch pools for, 1-2

- ARCHIVING system option, 2-27

- ASCII code, ANSI87 tape labels in, B-9

assigning

- serial numbers, examples of, 1-3

tapes

- to scratch pools, 1-3
- to tasks, 1-7

ASSIGNMENT STOPPED BY

- TAPEMANAGER LIBRARY system message, 5-29

- CHECK_TAPE_ASSIGNMENT procedure and, 5-29, A-4

- asynchronous purges of tapes, 1-8

- ATTERR177, 5-36

attributes

file

- AUTOUNLOAD, 1-6, 3-2, 3-47, 3-51
- correspondence to I/O subsystem types, 3-44

NEWFILE, 1-12
 recording changes made by
 TAPEMANAGER library, 1-11
 SCRATCHPOOL, 1-3, 1-5, 3-5, 3-47,
 4-6
 security, 1-8, 3-42
 task
 LABELFORMAT, 1-9, 1-10, 3-7, 3-43
 volume
 FAMILYOWNER, 3-42
 SAVEFACTOR, 1-10
 SECURITYGUARD, 3-42
 SECURITYLABELS, 1-8, 3-8, 3-42,
 3-43, 3-51, 3-53
 SECURITYTYPE, 3-42
 SECURITYUSE, 3-42
 volume security
 FAMILYOWNER, 3-53
 SECURITYGUARD, 3-53
 SECURITYTYPE, 3-53
 SECURITYUSE, 3-53
 audit
 control information, 4-2
 files, copying, 4-4
 AUDIT option
 system, 2-28
 VISIBLE DBS utility, 4-7
 AUDIT SCRATCHPOOL command
 VISIBLE DBS utility, 1-5, 4-6
 autobackup number, 2-23
 AUTODC system option, 2-24
 AUTOLOAD option, SEND (Send Message)
 system command, 2-47
 automatic tape unloading, 1-5
 controlling
 ADD statements (WFL), 1-7
 COPY statements (WFL), 1-7
 through programs, 1-7
 with system operations, 1-6
 automatic-unload mode, 1-6
 AUTORECOVERY system option, 2-23
 AUTORM system option, 2-23
 AUTORUNNING system option, 2-24
 AUTOUNLOAD
 file attribute, 1-6, 3-2, 3-47, 3-51
 status, 2-4

B

B 3500 tape label format, B-17
 B 5500 tape label format, B-19, B-20

backing up DMSII database to tape, 1-5
 BACKUPBYJOBNR system option, 2-26
 BAD DISK HEADER ON TAPE system
 message, A-4
 BCL characters, B 5500 tape labels, B-19
 BLASTED status, peripheral units, 2-37
 blocked messages, network processor, 2-47
 BLOCKING option, SEND (Send Message)
 system command, 2-47
 BLOCKTIMEOUT option, SEND (Send
 Message) system command, 2-47
 BNA
 port debug information, 2-30
 recovery, 2-29
 buffers, NP, 2-50
 BUFFERSIZE option, SEND (Send Message)
 system command, 2-47

C

CALLER parameter
 PROGRAM_INTERFACE procedure,
 5-39
 CALL_TAPEMANAGER procedure, 5-53,
 5-54
 role in tape management systems, 5-7
 CANDE commands, COPY and ADD, 3-47
 CANDIDATE_FILE parameter
 CHECK_TAPE_ASSIGNMENT
 procedure, 5-26
 CHECK_TAPE_UNIT_ASSIGNMENT
 procedure, 5-36
 CANDIDATE_TASK parameter
 CHECK_TAPE_ASSIGNMENT
 procedure, 5-27
 CHECK_TAPE_UNIT_ASSIGNMENT
 procedure, 5-36
 CANDIDATE_UNIT parameter
 CHECK_TAPE_UNIT_ASSIGNMENT
 procedure, 5-36
 CANNOT QUIT TAPEMANAGER system
 message, A-5
 CANNOT START TAPEMANAGER system
 message, A-5
 card input, 2-23
 card punch
 files, assigning to backup disk, 2-24
 card reader, 2-23
 Cartridge Library Unit (CLU)
 devices, dump recommendations, 4-1, 4-4
 hardware

- activities, reporting, 1-12
- cartridge tapes, 1-10
 - file protection mechanisms, 1-10
 - Unisys HD600XTD, HD6150, 1-10
- catalog family, missing, 2-35
- cataloging, 1-8, 2-30
- CATALOGING system option, 2-26
- CATTEST system option, 2-30
- CCSVERSION system option, 2-60
 - displaying current, 2-59
 - setting, 2-59
- CDONLY system option, 2-23
- changing
 - SCRATCHPOOL file attribute
 - for audit tape operations, 1-5, 4-6
 - serial numbers, examples of, 1-3
 - write-enable status
 - disks, 2-4
 - tapes, 2-4
 - units, 2-4
- CHECK_TAPE_ASSIGNMENT procedure, 5-25, 5-29
 - invoked for nontape file opens, 5-17
- CHECK_TAPE_PURGE procedure, 5-30, 5-31
- CHECK_TAPE_REOPEN procedure, 5-32, 5-33
- CHECK_TAPE_UNIT_ASSIGNMENT procedure, 5-34, 5-36
- close operations, file, 1-1
- CLU Hardware Report (CLUHWREPORT)
 - system log entry, 1-12
- CLUHWREPORT (CLU Hardware Report)
 - system log entry, 1-12
- COBOL
 - files
 - B 5500 tape labels and, B-20
- commands
 - CANDE
 - ADD, 3-47
 - COPY, 3-47
 - COPYAUDIT utility
 - COPY, 4-4
 - DUMPALL utility
 - COPY, 1-4
 - system
 - affected by scratch pools, 1-2
 - DS (Discontinue), 5-57, 5-58
 - FA (File Attributes), 5-26
 - FR (Final Reel), 5-29
 - FREE, 2-2
 - HOLD (Hold Unit for Operator Use), 1-7, 1-11, 1-12, 2-2, 2-3, 2-33
 - IL (Ignore Label), 5-34, 5-36
 - media library, 2-53
 - MODE (Unit Mode), 1-6, 2-4
 - network processor, 2-48
 - OL (Display Label and Paths), 2-8
 - OU (Output Unit), 1-3, 2-33, 5-34, 5-35, A-13
 - PER (Peripheral Status), 2-2, 2-37
 - PG (Purge), 1-3, 2-43
 - PGL (Purge and Lock), 2-43
 - pointing to scratch pools, 1-2
 - RESTRICT (Restrict Unit or Volume), A-14, A-30
 - RY (Ready), 2-2, 2-43, 2-48, 2-57, A-13
 - SECOPT (Security Options), 1-8
 - SEND (Send Message), 2-47, 5-7, 5-57
 - SEND TM QUIT, 2-54, 5-14, 5-57, 5-60
 - SEND TM START, 2-54, 5-13, 5-14, A-27
 - SEND TM STATUS, 2-55
 - SN (Serial Number), 1-3, 2-56
 - SNL (Serial Number Locked), 2-56
 - SYSOPS (System Options), 1-9, 1-11, 2-59, 3-7, 3-43
 - TDIR (Tape Directory), 2-3
 - UL (Unlabeled), 5-34, 5-36
 - VISIBLE DBS utility
 - AUDIT SCRATCHPOOL, 1-5, 4-6
- continuation tapes
 - TAPEMANAGER library and, 5-17
- control files
 - creating, 4-2
 - table of contents, 4-2
- control, inboard and outboard information, 2-9
- controlling
 - effects of file close actions, 1-1
 - tape usage with TMS, 1-13
- CONTROL_INTERFACE procedure, 5-37, 5-38
- CONVENTION system option, 2-60
 - displaying current, 2-59
 - setting, 2-59
- COPY
 - command
 - CANDE, 3-47
 - COPYAUDIT utility, 4-4, 4-6
 - DUMPALL utility, 1-4

Index

- option, VISIBLE DBS utility, 4-7
- statement, Work Flow Language (WFL),
1-2, 1-3, 1-5, 1-7, 3-47
- COPYAUDIT utility
 - COPY command, 4-4, 4-6
- copying audit files, 4-4
- CPBDONLY system option, 2-24
- creating
 - ANSI87 tapes, 3-43
 - groups of tapes under common identifier,
1-2
 - scratch pools as part of tape purges, 1-2
 - tape management systems (TMSs), 1-12
- CREATIONDATE file attribute
 - tape expiration and, 5-23
- CRUNCH system option, 2-25

D

- DASNAME option, SEND (Send Message)
system command, 2-47
- data comm, automatic initiation of, 2-24
- Data Management System II (DMSII), 1-2,
4-1
 - backing up to tape, example of, 1-5
 - copying audit files, 4-4
 - database, backing up to tape, 1-5
 - DMUTILITY DUMP statement, 4-3
 - dump media, 4-3
 - dump tape directory maintenance, 4-1
 - option, SERIALNO, 1-5
 - SCRATCHPOOL file attribute in, 1-5, 4-2
 - table of contents, 4-2
 - TAPEMANAGER library and, 5-8
 - technical overview, 4-1
 - utilities, 4-2
 - pointing to scratch pools, 1-2
- data, reading from expired tapes, 1-10
- databases
 - DMSII, backing up to tape, 1-5
 - TAPEMANAGER library and, 5-8
- date, operator verification of, 2-26
- DCKEYIN statement
 - role in tape management systems, 5-6
- default procedure, tape drives
 - upon performing reel switches, 1-5
 - when released from use, 1-5
- defining scratch pools
 - by account number, 1-2
 - by media type, 1-2
 - for archive tapes, 1-2
 - for dump tapes, 1-2
- deimplementation warning
 - NOSUMMARY system option, 2-26
- density, tape
 - as criterion for purging, 2-44
 - assigning, 2-56
- DESTROYED VOLUME NOT ADDED IN
SCRATCH POOL system message,
A-5
- device application supervisor name, NP, 2-50
- DIAGNOSTICDUMP system option, 2-28
- DIAGNOSTICS option, SEND (Send
Message) system command, 2-47
- DIAGNOSTICS system option, 2-23
- digit, mark, B-6
- DIRDEBUG system option, 2-26
- DISK FILE HEADER VERSION ERROR
system message, A-6
- disks, 2-43
 - address, verifying, 2-27
 - mirroring, 2-28
 - write-enable status, 2-4
- DMCONTROL
 - UPDATE statement, 4-3
- DMDUMPDIR utility, 4-1, 4-4
- DMSII, (See Data Management System II
(DMSII))
- DMUTILITY DUMP statement, 4-3
- double tape mark notation, tape label
illustrations, B-1
- drives, tape, 1-1
- DS (Discontinue) system command, 5-58
 - no effect on TAPEMANAGER library,
5-57
 - TAPEMANAGER protected from, 5-57
- dump
 - media, 4-3
 - tapes
 - defining scratch pools for, 1-2
 - directory maintenance, 4-1
- DUMP option, SEND (Send Message)
system command, 2-47
- DUMPALL utility
 - SCRATCHPOOL file attribute in, 1-4
- DUPINTRINSICS system option, 2-24
- DUPLICATE SERIAL NUMBER system
message, A-33
- DUPSUPERVISOR system option, 2-23

E

EBCDIC

- characters, scratch pool name stored in, 1-12
- code, ANSI69 tape labels in, B-2

end-of-file

- label, B-6, B-8, B-14, B-16
- notation, tape label illustrations, B-1

END-OF-FILE READING HDR1 LABEL

- system message, A-6

end-of-volume

- label, B-6, B-14, B-16
- notation, tape label illustrations, B-1

entries, system log

- CLU Hardware Report (CLUHWREPORT), 1-12

CLUHWREPORT (CLU Hardware Report), 1-12

File Close (LOGCLOSETYPE), 1-11

File Open (LOGOPENTYPE), 1-11

LOGCLOSETYPE (File Close), 1-11

LOGMAJLIBS (Tape Library Entry), 3-74

LOGMAJ_MEDIALIB (Tape Library Entry), 1-12

LOGMAJVOL (Volume Status), 1-11

LOGOPENTYPE (File Open), 1-11

LOGSETSTAT (SETSTATUS call), 1-11

Minor Type 12 (Tape Volume Used), 1-11, 3-70, 3-71, 3-72, 3-73, 3-74

Minor Type 13 (Tape Volume Hold), 1-11, 3-70, 3-71, 3-72, 3-73, 3-74

Minor Type 26 (Tape Library Entry), 1-12, 3-74

Minor Type 3 (Tape Volume Purged), 1-12

Minor Type 5 (Tape Volume NEWFILE), 1-12

SETSTATUS call (LOGSETSTAT), 1-11

Tape Library Entry (Minor Type 26), 1-12, 3-74

Tape Volume Hold (Minor Type 13), 1-11, 3-70, 3-71, 3-72, 3-73, 3-74

Tape Volume New File (Minor Type 5), 1-12

Tape Volume Purged (Minor Type 3), 1-12

Tape Volume Used File (Minor Type 12), 1-11, 3-70, 3-71, 3-72, 3-73, 3-74

Volume Status (LOGMAJVOL), 1-11

EOF (end-of-file) notation, tape label illustrations, B-1

EOTSTATISTICS system option, 2-29

EOV (end-of-volume) notation, tape label illustrations, B-1

ERROR CLOSING TAPE FILE system message, A-6

ERROR OPENING TAPE FILE system message, A-6

ERROR READING TAPE LABEL system message, A-7

ERROR WRITING TAPE LABEL OR TAPE MARK system message, A-7

ERROR: ATTRIBUTE MODIFICATION PROHIBITED system message, 5-36

errors

messages

- inappropriate OU response, 2-35
- PG response mismatch, 2-45
- SETSTATUS returns, 3-32, 3-33, 3-36, 3-40

event information retrieval

- role in TMS design, 5-4

examining tapes before MCP assigns to tasks, 1-7

examples

ALGOL

- file close operations and automatic tape unloading, 3-4
- writing program output to scratchpools, 1-4

ARCHIVE statement, 3-47

assigning

- serial numbers, 1-3
- tapes to scratch pools, 1-3

changing serial numbers, 1-3

controlling automatic tape unloading in programs, 1-7

COPY statement, 3-51

examining tapes before MCP assigns to tasks, 1-7

HOLD (Hold Unit for Operator Use) system command, 1-7, 2-2

MODE (Unit Mode) system command, 1-6, 2-6

OL (Display Label and Paths) system command, 2-14, 2-17, 2-18

OU (Output Unit) system command, 1-3

overriding request for scratch pool, 1-3

PER (Peripheral Status) system command, 2-38, 2-39, 2-40, 2-41, 2-42

PG (Purge) system command, 1-3, 2-46

purging tapes, 1-3

railroad diagrams, C-6, C-7, C-8

Index

- SEND (Send Message) system command, 2-53
 - serial numbers
 - assigning, changing, 1-3
 - SETSTATUS calls, 3-32, 3-34, 3-37, 3-41
 - SN (Serial Number) system command, 1-3, 2-58
 - SYSOPS (System Options) system command, 2-61, 2-62, 2-63, 2-64
 - TAPESERVER library
 - initiating, 2-55
 - requesting status of, 2-55
 - terminating, 2-55
 - using SCRATCHPOOL file attribute
 - in DMSII, 1-5
 - in Work Flow Language (WFL) COPY statement, 1-3
 - VOLUME statement, 3-53
 - Work Flow Language (WFL) COPY statement, 1-3
 - EXPECTED RECORD ID system message, A-7
 - expiration dates
 - reading tape data past, 1-10
 - tape labels, 1-10
 - expired tapes and tape management systems, 5-23
 - exported procedures, TAPEMANAGER library, 1-12, 1-13
- F**
- FA (File Attributes) system command effect simulated by CHECK_TAPE_ASSIGNMENT procedure, 5-26
 - FAMILYOWNER MISMATCH OR SECURITY ERROR, A-7
 - FAMILYOWNER volume security attribute, 3-42, 3-53
 - FAULT REASON = system message, A-8
 - fault-handling
 - within TAPEMANAGER library, 5-9
 - FETCH statement, Work Flow Language (WFL), 2-26
 - file assignment statements
 - SCRATCHPOOL file attribute in, 1-3
 - file attributes
 - AUTOUNLOAD, 1-6, 3-2, 3-47, 3-51
 - correspondence to I/O subsystem types, 3-44
 - FLEXIBLE, 2-29
 - NEWFILE, 1-12
 - recording changes made by TAPEMANAGER library, 1-11
 - SCRATCHPOOL, 1-3, 1-5, 3-5, 3-47, 3-51, 4-6
 - SERIALNO, 2-27
 - USECATALOG, 2-30
 - File Close (LOGCLOSETYPE) system log entry, 1-11
 - file close operations
 - ALGOL
 - automatic tape unloading and, example, 3-4
 - effects on tape drives, 1-1
 - programmatic, 1-5
 - file header format
 - header 1
 - ANSI69 tapes, B-6
 - ANSI87 tapes, B-14
 - USASI B 3500 tapes, B-17
 - header 2
 - ANSI69 tapes, B-6
 - ANSI87 tapes, B-15
 - header 3, ANSI87 tapes, B-16
 - FILE NOT VERIFIED system message, A-9
 - File Open (LOGOPENTYPE) system log entry, 1-11
 - file protection mechanisms, cartridge tapes, 1-10
 - FILE TITLE ERROR system message, A-10
 - FILE VERIFIED system message, A-10
 - <file name> REQUIRES <unit identifier> system message, 1-4, 2-35, A-17
 - FILECOPY utility, 1-4, 1-7
 - AUTOUNLOAD file attribute in, 1-7
 - SCRATCHPOOL file attribute in, 1-7
 - files
 - ALGOL
 - B 5500 tape labels and, B-20
 - audit
 - copying, 4-4
 - COBOL
 - B 5500 tape labels and, B-20
 - crunching, 2-25
 - data corruption, 2-24
 - disk space exhausted, 2-28
 - duplicate, 2-23
 - hazardous, A-13, A-31
 - invalid access, 2-24
 - multivolume, tape
 - ANSI69 format, B-2

FILESATURATION system option, 2-28
 FLEXIBLE file attribute, 2-29
 formats
 ANSI87 tape labels, B-8
 B 5500 tape labels, B-19, B-20
 tape labels
 ANSI69, 1-9, 1-10, 3-8, B-2
 ANSI87, 1-9, 1-10, 3-43, B-8
 B 5500, B-19, B-20
 standard, B-1
 USASI B 3500, B-17
 unlabeled tapes
 multifile, B-1
 single-file, B-1
 USASI B 3500 tape labels, B-17
 FR (Final Reel) system command
 simulated by CHECK_TAPE_ASSIGNMENT result, 5-18, 5-29
 FREE (Free Resource) system command, 2-2
 FREEINPUTS option, SEND (Send Message) system command, 2-47
 functions
 GETSTATUS
 role in tape management systems, 5-6
 SYSTEMSTATUS (DCALGOL), 5-6
 TAPEMANAGER
 as part of tape management system, 1-12

G

GETSTATUS function
 role in tape management systems, 5-6
 GOINGAWAY status of TAPEMANAGER library, 5-15, 5-57, 5-58
 grouping tapes under common identifier, 1-2

H

halt/loads
 automatic-unload mode preserved across, 1-6
 values retained across
 automatic-unload mode, 1-6
 AUTOUNLOAD option settings, 2-7
 options set with SYSOPS command, 1-9
 hardware activities, CLU, reporting, 1-12

hazardous files, A-13, A-31
 HDR (file header) notation, tape label illustrations, B-1
 HISTORY REASON =, CAUSE =, TYPE = system message, A-10
 HISTORYCAUSE task attribute, 3-9
 HISTORYREASON task attribute, 3-11
 HOLD (Hold Unit for Operator Use)
 SETSTATUS call, 3-33
 system command, 1-7, 1-11, 1-12, 2-2, 2-3, 2-33
 examples of, 1-7
 HOLD of unit
 status, 2-2, 2-33
 removing, 1-7
 host control (HC) unit
 changing write-enable status of, 2-4
 locking, 2-43
 purging, 2-43
 READPARTNER specification, 2-4
 WRITEPARTNER specification, 2-4
 hubindex, read and write operations, 2-4

I

I/O ERROR ON CHECKSUM RECORD
 system message, A-12
 I/O ERROR READING HDR1 LABEL
 system message, A-12
 I/O ERROR WHILE ACCESSING
 GUARDFILE system message, A-13
 ID option, SEND (Send Message) system command, 2-47
 ignoring tape expiration dates, 1-10
 IL (Ignore Label) system command, 5-36
 causing CHECK_TAPE_UNIT_ASSIGNMENT call, 5-34
 detecting as source of UNITNO assignment, 5-20
 IL, OU, or UL, in CHECK_TAPE_UNIT_ASSIGNMENT figure, 5-19
 ILLEGAL OPTIONAL FILE system message, 5-29
 CHECK_TAPE_REOPEN procedure and, 5-29
 INACTIVE status, TAPEMANAGER library, 5-13, 5-56
 inboard information control, 2-9

Index

- port, 2-9
- InfoGuard security enhancement software, 1-1, 1-8
- INITIALIZING status, TAPEMANAGER library, 5-14, 5-56
- INITIALREADS option, SEND (Send Message) system command, 2-47
- interrecord gap notation, tape label illustrations, B-1
- intrinsic, SETSTATUS, calls to, 1-8
- INVALID DENSITY SPECIFIED system message, A-11
- INVALID LIBRARY/MAINT TAPE BLOCKSIZE system message, A-11
- IO ERROR DURING COMPARE system message, A-12
- IO ERROR DURING COPY system message, A-12
- IODIAGNOSTICS system option, 2-30
- IORANGECHECK system option, 2-27
- ISCDEBUG system option, 2-30

- J**
- job
 - CDONLY system option and, 2-23
 - discontinuing, 5-10
 - number
 - printing by, 2-26
 - prohibiting from acquiring scratch tapes, 2-44, 2-57, 2-58
 - summary, 2-26, 3-57, 3-61
 - Work Flow Language (WFL) reactivating, 2-26

- K**
- KEYEDIOII files
 - creating, 2-28
- KEYEDIOII system option, 2-28

- L**
- LABELFORMAT
 - system option, 1-9, 1-10, 2-60, 3-7, 3-43
 - task attribute, 1-9, 1-10, 3-7, 3-43
- labels
 - for peripheral devices, 2-8
- tape
 - ANSI69 format, 1-9, 1-10, 3-8, B-2
 - ANSI87 format, 1-9, 1-10, 3-8, 3-43, B-8
 - B 5500 format, B-19
 - expiration date, 1-10
 - USASI B 3500 format, B-17
 - user header, ANSI69, B-2
 - user trailer, ANSI69, B-2
- LANGUAGE system option, 2-60
 - displaying current, 2-59
 - setting, 2-59
- level, Mark, of MCP, B-6
- libraries, support
 - media library, 2-53
 - network processor, 2-48
 - TAPEMANAGER, 1-12
- LIBRARY CANNOT OPEN THIS FILE system message, 5-25, 5-33, 5-34, A-13
- LINKING status, TAPEMANAGER library, 5-13, 5-14, 5-56
- LOAD option, SEND (Send Message) system command, 2-47
- locking tapes, on packs or host control (HC) units, 2-43
- LOCKTRACE system option, 2-27
- log entries, system
 - CLU Hardware Report (CLUHWREPORT), 1-12
 - CLUHWREPORT (CLU Hardware Report), 1-12
 - File Close (LOGCLOSETYPE), 1-11
 - File Open (LOGOPENTYPE), 1-11
 - LOGCLOSETYPE (File Close), 1-11
 - LOGMAJLIBS (Tape Library Entry), 3-74
 - LOGMAJ_MEDIALIB (Tape Library Entry), 1-12
 - LOGMAJVOL (Volume Status), 1-11
 - LOGOPENTYPE (File Open), 1-11
 - LOGSETSTAT (SETSTATUS call), 1-11
 - Minor Type 12 (Tape Volume Used), 1-11, 3-70, 3-71, 3-72, 3-73, 3-74
 - Minor Type 13 (Tape Volume Hold), 1-11, 3-70, 3-71, 3-72, 3-73, 3-74
 - Minor Type 26 (Tape Library Entry), 1-12, 3-74
 - Minor Type 3 (Tape Volume Purged), 1-12
 - Minor Type 5 (Tape Volume NEWFILE), 1-12
 - SETSTATUS call (LOGSETSTAT), 1-11

Tape Library Entry (Minor Type 26),
 1-12, 3-74
 Tape Volume Hold (Minor Type 13), 1-11,
 3-70, 3-71, 3-72, 3-73, 3-74
 Tape Volume New File (Minor Type 5),
 1-12
 Tape Volume Purged (Minor Type 3), 1-12
 Tape Volume Used File (Minor Type 12),
 1-11, 3-70, 3-71, 3-72, 3-73, 3-74
 Volume Status (LOGMAJVOL), 1-11
 log, system (SUMLOG), 1-11
 LOGCLOSETYPE (File Close) system log
 entry, 1-11
 logging new and changed log entries, 3-53
 CLU hardware report entries, 3-67
 file close entries, 3-60
 file open entries, 3-56
 SETSTATUS call entries, 3-68
 tape library entries, 3-74
 volume status entries, 3-70
 LOGIOERRORSNEW system option, 2-29
 LOGMAJLIBS (Tape Library Entry) system
 log entry, 3-74
 LOGMAJ_MEDIALIB (Tape Library Entry)
 system log entry, 1-12
 LOGMAJVOL (Volume Status) system log
 entry, 1-11
 LOGOPENTYPE (File Open) system log
 entry, 1-11
 LOGPOSITIONING system option, 2-27
 LOGSETSTAT (SETSTATUS call) system
 log entry, 1-11
 LP (Lock Program) system command
 TAPEMANAGER library and, 5-57
 LPBDONLY system option, 2-23

M

MAINTENANCE status, unit, 2-12
 mark digit, B-6
 Mark level, B-6
 Mark release level, MCP
 active when tape was created, B-6
 master control program (MCP)
 exported procedures
 CALL_TAPEMANAGER, 5-53
 how it identifies scratch tapes, B-6
 mark digit, B-6
 Mark level, B-6
 options, 2-21
 MAXINPUTS option, SEND (Send Message)
 system command, 2-47
 MCPINIT library attribute
 TAPEMANAGER libraries and, 5-10
 MCPTEST system option, 2-30
 media libraries, system commands, 2-53
 media type, defining scratch pools by, 1-2
 memory disk, 2-40
 displaying information, 2-40
 memory dump
 fatal, 2-23
 inhibiting, 2-23
 suppressing, 2-24
 messages, system
 ARCHIVE BACKUP ENTRY DELETED,
 A-3
 ASSIGNMENT STOPPED BY
 TAPEMANAGER LIBRARY, 5-29,
 A-4
 ATTERR177, 5-36
 BAD DISK HEADER ON TAPE, A-4
 CANNOT QUIT TAPEMANAGER, A-5
 CANNOT START TAPEMANAGER, A-5
 DESTROYED VOLUME NOT ADDED
 IN SCRATCH POOL, A-5
 DISK FILE HEADER VERSION ERROR,
 A-6
 DUPLICATE SERIAL NUMBER, A-33
 END-OF-FILE READING HDR1 LABEL,
 A-6
 ERROR CLOSING TAPE FILE, A-6
 ERROR OPENING TAPE FILE, A-6
 ERROR READING TAPE LABEL, A-7
 ERROR WRITING TAPE LABEL OR
 TAPE MARK, A-7
 ERROR: ATTRIBUTE MODIFICATION
 PROHIBITED, 5-36
 EXPECTED RECORD ID, A-7
 FAMILYOWNER MISMATCH OR
 SECURITY ERROR, A-7
 FAULT REASON =, A-8
 FILE NOT VERIFIED, A-9
 FILE TITLE ERROR, A-10
 FILE VERIFIED, A-10
 <file name> REQUIRES
 <unit identifier>, 2-35, A-17
 example of, 1-4
 HISTORY REASON =, CAUSE =,
 TYPE =, A-10
 I/O ERROR ON CHECKSUM RECORD,
 A-12

Index

- I/O ERROR READING HDR1 LABEL, A-12
- I/O ERROR WHILE ACCESSING GUARDFILE, A-13
- ILLEGAL OPTIONAL FILE, 5-29
- INVALID DENSITY SPECIFIED, A-11
- INVALID LIBRARY/MAINT TAPE BLOCKSIZE, A-11
- IO ERROR DURING COMPARE, A-12
- IO ERROR DURING COPY, A-12
- LIBRARY CANNOT OPEN THIS FILE, 5-25, 5-33, 5-34, A-13
- MSRCAT11, A-30
- MSRCAT16, A-14
- MSRCAT26, A-19
- MSRCAT27, A-19
- MSRCAT9, A-20
- MSRDIR1, A-21
- MSRDPC177, 5-31, A-15
- MSRDPC181, 5-31, A-19
- MSRDPC183, 5-31, A-18
- MSRDPC93, A-17
- MSRDPC96, A-11
- MSREH0 (TAPE READ), A-24
- MSREH0 (TAPE WRITE), A-25
- MSRFIB16, 5-29
- MSRFIB268, 5-33, A-17
- MSRFIB270, 5-25, 5-33, 5-34, A-13
- MSRFIB271, 5-29, A-4
- MSRFIB74, A-15
- MSRFOUT11, A-13
- MSRFOUT15, A-14
- MSRFOUT23, 5-19, A-29
- MSRFOUT30, 2-35, A-17
- MSRFOUT30, example of, 1-4
- MSRFOUT9, A-7
- MSRLM0, A-21, A-22
- MSRLM1, A-22
- MSRLM11, A-10
- MSRLM12, A-4
- MSRLM15, A-12
- MSRLM17, A-21
- MSRLM19, A-5
- MSRLM2, A-22, A-23
- MSRLM21, A-7
- MSRLM22, A-30
- MSRLM24, A-28
- MSRLM26, A-6
- MSRLM27, A-4
- MSRLM29, A-6
- MSRLM3, A-23
- MSRLM30, A-12
- MSRLM31, A-6
- MSRLM34, A-23
- MSRLM37, A-6
- MSRLM39, A-7
- MSRLM4, A-12
- MSRLM42, A-13, A-31
- MSRLM43, A-14, A-30
- MSRLM5, A-15
- MSRLM6, A-23
- MSRLM69, A-14
- MSRLM7, A-16
- MSRLM79, A-32
- MSRLM8, A-7
- MSRLM81, A-24
- MSRLM82, A-35
- MSRLM84, A-12
- MSRLM85, A-20
- MSRLM86, A-10
- MSRLM88, A-9
- MSRLM90, A-20
- MSRLM92, A-3
- MSRLM96, A-11
- MSRMISC117, 2-24
- MSRMISC201, A-13
- MSRMISC204, 5-13, A-30
- MSRMISC205, A-29
- MSRMISC206, A-26
- MSRMISC207, A-26
- MSRMISC208, 5-13, A-27, A-36
- MSRMISC209, 5-13, A-27, A-28, A-36
- MSRMISC210, 5-13, A-27, A-28, A-36
- MSRMISC211, A-26
- MSRMISC212, A-5
- MSRMISC213, A-5
- MSRMISC214, 5-38, A-25
- MSRMISC215, 5-38, A-26
- MSRMISC220, 5-59, A-28
- MSRMISC221, 5-9, 5-59, A-8, A-10, A-31
- MSRMISC222, 5-14, A-27
- MSRMISC223, A-8
- MSRMISC224, A-10
- MSRRS26, A-18
- MSRSOFT42, A-18
- MSRVL12, A-35
- MSRVL16, A-15
- MSRVS1, A-34
- MSRVS10, A-33
- MSRVS11, A-33
- MSRVS12, A-33
- MSRVS13, A-33
- MSRVS14, A-34
- MSRVS15, A-33

- MSRVS16, A-35
MSRVS17, A-33
MSRVS18, A-33
MSRVS19, A-33
MSRVS2, A-34
MSRVS20, A-33
MSRVS21, A-33
MSRVS22, A-33
MSRVS23, A-33, A-35
MSRVS3, A-34
MSRVS4, A-34
MSRVS5, A-34
MSRVS6, A-33
MSRVS7, A-33
MSRVS8, A-33
MSRVS9, A-33
MSRWARNG56, A-36
MSRWARNG57, A-37
MSRWARNG58, A-37
MSRWARNG59, A-37
NEED AN OUTPUT TAPE FOR OUMT,
A-13
NEW SOURCE UNIT OR VOLUME IS
RESTRICTED, A-13, A-31
NO CATALOG ACTION (LIBRARY
TAPE), A-14
NO ENTRY FOR VOLUME IN 'VOLUME
DIRECTORY', A-14
NO RECOPY WHEN USING THIS KIND
OF TAPE, A-14
NOT A SCRATCH VOLUME, A-15
NOT CLOSED, A-15
PURGE REQUEST REJECTED BY
TAPEMANAGER, 5-31, A-15
RECOPY REQD, A-15
RECORD SEQUENCE ERROR, A-16
REOPEN STOPPED BY
TAPEMANAGER, 5-33, A-17
REQUIRES VOLUME DELETE, A-17
RESTART ABORTED, A-18
SCRATCHPOOL REJECTED BY
TAPEMANAGER, 5-31, A-18
SECURITYLABELS REQUIRE ANSI87
LABELS, A-18
SERIAL NUMBER REJECTED BY
TAPEMANAGER, 5-31, A-19
SERIALNO MATCHES BUT TAPE IS A
LIBRARY MAINTENANCE TAPE,
A-19
SERIALNO MATCHES BUT TAPE IS
NOT FIRST REEL, A-19
TAPE DISKHEADER READ ERROR,
A-22, A-23
TAPE DISKHEADER WRITE ERROR,
A-23
TAPE DOESN'T SEEM TO BACKSPACE,
A-4
TAPE FILE MISSING, A-23
TAPE POSITIONING ERROR, A-23
TAPE READ ERROR, A-24
TAPE WILL BE PURGED, A-24
TAPE WRITE ERROR, A-25
TAPEMANAGER FAILED IN
INITIALIZATION PROCEDURE,
5-38, A-25
TAPEMANAGER FAILED IN
TERMINATION PROCEDURE,
5-38, A-26
TAPEMANAGER IS ACTIVE, A-26
TAPEMANAGER IS INACTIVE, A-26
TAPEMANAGER LIBRARY HAS
LINKED, A-26
TAPEMANAGER LIBRARY IS
ALREADY ACTIVE, 5-14, A-27
TAPEMANAGER LIBRARY IS MISSING
CONTROL_INTERFACE, 5-13,
A-27
TAPEMANAGER LIBRARY WILL NOT
BE USED, 5-13, A-27, A-28
TAPEMANAGER REQUIRED, 5-59,
A-28
TITLE FOUND WAS, A-28
TYPE DISAGREES WITH KIND, 5-19,
A-29
UNABLE TO LINK TO
TAPEMANAGER, 5-13, A-30
UNABLE TO LINK TO TAPEMANAGER
LIBRARY, A-29, A-30
UNEXPECTED TAPE MARK, A-30
UNIT NOT VOLUMED, A-30
UNRESTRICT SOURCE AND REPLY
OK TO COPY, A-14, A-30
USER PROCESS FAULTED OR DSED
IN TAPEMANAGER, 5-9, 5-59,
A-8, A-10, A-31
VERIFY ERROR, A-32
VOLUME DIRECTORY, A-33
VOLUME DIRECTORY <action> NOT
DONE, A-35
VOLUME DIRECTORY FAULT IN
VSHANDLER, A-34
VOLUME DIRECTORY NOT SNED,
A-35

Index

- VOLUME DIRECTORY RECORD HAS
INVALID DATA, A-34
- VOLUME FAMILY NOT DELETED,
A-35
- WARNING 56, A-36
- WARNING 57, A-37
- WARNING 58, A-37
- WARNING 59, A-37
- WARNING, TAPE WAS NOT WRITTEN
WITH CHECKSUMS, A-35
- WARNING: TAPEMANAGER LIBRARY
IS MISSING OBJECTS, 5-13, A-27,
A-28, A-36
- 'OP AUTORUNNING' IS SET, REPLY
<reply> ASSUMED, 2-24
- Minor Type 12 (Tape Volume Used) system
log entry, 1-11, 3-70, 3-71, 3-72,
3-73, 3-74
- Minor Type 13 (Tape Volume Hold) system
log entry, 1-11, 3-70, 3-71, 3-72,
3-73, 3-74
- Minor Type 26 (Tape Library Entry) system
log entry, 1-12, 3-74
- Minor Type 3 (Tape Volume Purged) system
log entry, 1-12
- Minor Type 5 (Tape Volume NEWFILE)
system log entry, 1-12
- mirrored sets
 - bringing on line, 2-28
- MIRRORING system option, 2-28
- mix entries, waiting, 2-45
- MODE (Unit Mode) system command, 1-6,
2-4
- MP (Mark Program) system command
 - assigning identity to TAPEMANAGER
library, 5-7, 5-56
- MSRCAT11 system message, A-30
- MSRCAT16 system message, A-14
- MSRCAT26 system message, A-19
- MSRCAT27 system message, A-19
- MSRCAT9 system message, A-20
- MSRDIR1 system message, A-21
- MSRDPC177 system message, 5-31, A-15
- MSRDPC181 system message, 5-31, A-19
- MSRDPC183 system message, 5-31, A-18
- MSRDPC93 system message, A-17
- MSRDPC96 system message, A-11
- MSREH0 (TAPE READ) system message,
A-24
- MSREH0 (TAPE WRITE) system message,
A-25
- MSRFIB16 system message, 5-29
- MSRFIB268 system message, 5-33, A-17
- MSRFIB270 system message, 5-25, 5-33,
5-34, A-13
- MSRFIB271 system message, 5-29, A-4
- MSRFIB74 system message, A-15
- MSRFOUT11 system message, A-13
- MSRFOUT15 system message, A-14
- MSRFOUT23 system message, 5-19, A-29
- MSRFOUT30 system message, 2-35, A-17
- MSRFOUT30 system message, example of,
1-4
- MSRFOUT9 system message, A-7
- MSRLM0 system message, A-21, A-22
- MSRLM1 system message, A-22
- MSRLM11 system message, A-10
- MSRLM12 system message, A-4
- MSRLM15 system message, A-12
- MSRLM17 system message, A-21
- MSRLM2 system message, A-22, A-23
- MSRLM21 system message, A-7
- MSRLM22 system message, A-30
- MSRLM24 system message, A-28
- MSRLM26 system message, A-6
- MSRLM27 system message, A-4
- MSRLM29 system message, A-6
- MSRLM3 system message, A-23
- MSRLM30 system message, A-12
- MSRLM31 system message, A-6
- MSRLM34 system message, A-23
- MSRLM37 system message, A-6
- MSRLM39 system message, A-7
- MSRLM4 system message, A-12
- MSRLM42 system message, A-13, A-31
- MSRLM43 system message, A-14, A-30
- MSRLM5 system message, A-15
- MSRLM6 system message, A-23
- MSRLM69 system message, A-14
- MSRLM7 system message, A-16
- MSRLM79 system message, A-32
- MSRLM8 system message, A-7
- MSRLM81 system message, A-24
- MSRLM82 system message, A-35
- MSRLM84 system message, A-12
- MSRLM85 system message, A-20
- MSRLM86 system message, A-10
- MSRLM88 system message, A-9
- MSRLM90 system message, A-20
- MSRLM92 system message, A-3
- MSRLM96 system message, A-11
- MSRMISC117 system message, 2-24
- MSRMISC201 system message, A-13
- MSRMISC204 system message, 5-13, A-30

MSRMISC205 system message, A-29
 MSRMISC206 system message, A-26
 MSRMISC207 system message, A-26
 MSRMISC208 system message, 5-13, A-27,
 A-36
 MSRMISC209 system message, 5-13, A-27,
 A-28, A-36
 MSRMISC210 system message, 5-13, A-27,
 A-28, A-36
 MSRMISC211 system message, A-26
 MSRMISC212 system message, A-5
 MSRMISC213 system message, A-5
 MSRMISC214 system message, 5-38, A-25
 MSRMISC215 system message, 5-38, A-26
 MSRMISC220 system message, 5-59, A-28
 MSRMISC221 system message, 5-9, 5-59,
 A-8, A-10, A-31
 MSRMISC222 system message, 5-14, A-27
 MSRMISC223 system message, A-8
 MSRMISC224 system message, A-10
 MSRRS26 system message, A-18
 MSRSOFT42 system message, A-18
 MSRVL12 system message, A-35
 MSRVL16 system message, A-15
 MSRVL19 system message, A-5
 MSRVS1 system message, A-34
 MSRVS10 system message, A-33
 MSRVS11 system message, A-33
 MSRVS12 system message, A-33
 MSRVS13 system message, A-33
 MSRVS14 system message, A-34
 MSRVS15 system message, A-33
 MSRVS16 system message, A-35
 MSRVS17 system message, A-33
 MSRVS18 system message, A-33
 MSRVS19 system message, A-33
 MSRVS2 system message, A-34
 MSRVS20 system message, A-33
 MSRVS21 system message, A-33
 MSRVS22 system message, A-33
 MSRVS23 system message, A-33, A-35
 MSRVS3 system message, A-34
 MSRVS4 system message, A-34
 MSRVS5 system message, A-34
 MSRVS6 system message, A-33
 MSRVS7 system message, A-33
 MSRVS8 system message, A-33
 MSRVS9 system message, A-33
 MSRWARNING56 system message, A-36
 MSRWARNING57 system message, A-37
 MSRWARNING58 system message, A-37
 MSRWARNING59 system message, A-37

MT status, 2-37
 multfile volumes
 ANSI69 tape format, B-2
 ANSI87 tape format, B-9
 unlabeled tape format, B-1
 multfile, multivolume tapes
 ANSI69 format, B-3
 ANSI87 format, B-10
 multivolume files
 ANSI69 tape format, B-2
 ANSI87 tape format, B-9
 multivolume tapes
 TAPEMANAGER library and, 5-17

N

NEED AN OUTPUT TAPE FOR OUMT
 system message, A-13
 NETRECOVERY system option, 2-29
 network processor
 communicating with, 2-47
 creating new controller stack for, 2-53
 diagnostic commands, 2-48
 maximizing throughput, 2-48
 nondiagnostic commands, 2-48
 support library, 2-48
 system commands, 2-48
 NEW SOURCE UNIT OR VOLUME IS
 RESTRICTED system message,
 A-13, A-31
 NEWFILE file attribute, 1-12
 NEWP
 procedures exported by TAPEMANAGER
 library written in, 5-7
 TAPEMANAGER library exported
 procedures
 EXCEPTION, 5-8
 NEW_FILENAME parameter
 of CHECK_TAPE_REOPEN procedure,
 5-33
 NF (No File) system command
 simulated by CHECK_TAPE_
 ASSIGNMENT result, 5-18,
 5-29
 NO CATALOG ACTION (LIBRARY TAPE)
 system message, A-14
 NO ENTRY FOR VOLUME 'IN VOLUME
 DIRECTORY' system message, A-14
 NO RECOPY WHEN USING THIS KIND
 OF TAPE system message, A-14
 NOCHECK system option, 2-23

Index

NODUMP system option, 2-24
NOFETCH system option, 2-26
NOSUMMARY system option, 2-26
NOT A SCRATCH VOLUME system message, A-15
NOT CLOSED system message, A-15
NOT READY status, unit, 2-12
notations, tape label illustrations, B-1
NP, (See network processor)

O

OF (Optional File) system command
simulated by CHECK_TAPE_ASSIGNMENT result, 5-18, 5-29
OFFLINE status, unit, 2-12
OK (Reactivate) system command
and TAPEMANAGER_INITIALIZATION independent runner, 5-13
OKTIMEANDDATE system option, 2-26
OL (Display Label and Paths) system command, 2-8
ONEONLY library attribute
TAPEMANAGER libraries and, 5-10
ONLINE status, unit, 2-12
OP (Options) command, 2-21
OP AUTORUNNING IS SET system message, 2-24
open operations, for tape files
handling by TAPEMANAGER, 5-15
open requests, in CHECK_TAPE_ASSIGNMENT figure, 5-17
OPEN system option, 2-22
options
Data Management System II (DMSII)
SERIALNO, 1-5
master control program (MCP), 2-21
security
TAPECHECK, 1-8
SEND (Send Message) system command
ANALYZE, 2-47
AUTOLOAD, 2-47
BLOCKING, 2-47
BLOCKTIMEOUT, 2-47
BUFFERSIZE, 2-47
DIAGNOSTICS, 2-47
DUMP, 2-47
FREEINPUTS, 2-47
ID, 2-47
INITIALREADS, 2-47
LOAD, 2-47
MAXINPUTS, 2-47
PROGRAMDUMP, 2-47
QUIT, 2-47
RELEASE, 2-47
RESET, 2-47
SET, 2-47
TANKING, 2-47
system, 2-59
ARCHIVING, 2-27
AUDIT, 2-28
AUTODC, 2-24
AUTORECOVERY, 2-23
AUTORM, 2-23
AUTORUNNING, 2-24
BACKUPBYJOBNR, 2-26
CATALOGING, 2-26
CATTEST, 2-30
CDONLY, 2-23
CONVENTION, 2-60
CPBDONLY, 2-24
CRUNCH, 2-25
DIAGNOSTICDUMP, 2-28
DIAGNOSTICS, 2-23
DIRDEBUG, 2-26
DUPINTRINSICS, 2-24
DUPSUPERVISOR, 2-23
EOTSTATISTICS, 2-29
FILESATURATION, 2-28
IODIAGNOSTICS, 2-30
IORANGECHECK, 2-27
ISCDEBUG, 2-30
KEYEDIOII, 2-28
LABELFORMAT, 1-9, 1-10, 2-60, 3-7, 3-43
LABELFORMAT = ANSI87DEFAULT, 1-9
LANGUAGE, 2-60
LOCKTRACE, 2-27
LOGIOERRORSNEW, 2-29
LOGPOSITIONING, 2-27
LPBDONLY, 2-23
MCPTEST, 2-30
MIRRORING, 2-28
NETRECOVERY, 2-29
NOCHECK, 2-23
NODUMP, 2-24
NOFETCH, 2-26
NOSUMMARY, 2-26
OKTIMEANDDATE, 2-26
OPEN, 2-22, 2-60

- PATHBALANCING, 2-29
- PDTODISK, 2-26
- PORTDEBUG, 2-30
- RESOURCECHECK, 2-26
- SEND (Send Message), 2-47
- SERIALNUMBER, 2-27
- TAPEEXPIRATION, 1-11, 2-60
- TAPEMANAGER, 2-60
- TAPESERVER, 2-60
- TERMINATE, 2-22
- TRANSWARNINGS, 2-24
- USECATDEFAULT, 2-30
- OU (Output Unit) system command, 2-33, 5-35, A-13
 - causing CHECK_TAPE_UNIT_ASSIGNMENT call, 5-34
 - detecting as source of UNITNO assignment, 5-20
 - example of, 1-3
 - outboard information
 - control, 2-9
 - port, 2-9
 - output files, directing to device, 2-33
 - overriding request for scratch pool
 - example of, 1-3
 - overriding scratch pool assignments, 1-2
- P**
- packs, 2-43, (See also disks)
 - locking, 2-43
 - purging, 2-43
 - sectors required on, 2-35
- parameters
 - calls to SETSTATUS intrinsic, 1-8
- path
 - dynamic balancing, 2-29
 - peripheral devices, 2-8
- PATHBALANCING system option, 2-29
- PDTODISK system option, 2-26
- PER (Peripheral Status) system command, 2-2, 2-37
- peripheral units
 - status, 2-37
 - BLASTED, 2-37
 - U, 2-37
 - UNINITIALIZED, 2-37
 - UNLABELED, 2-37
- PG (Purge)
 - SETSTATUS call, 1-8, 3-35
 - system command, 1-3, 2-43, 5-22
 - causing CHECK_TAPE_PURGE call, 5-30
- PGL (Purge and Lock)
 - SETSTATUS call, 1-8, 3-35
 - system command, 2-43
- pools, scratch, (See scratch pools)
- port
 - inboard and outboard information, 2-9
- PORTDEBUG system option, 2-30
- PRIMARY option, VISIBLE DBS utility, 4-7
- printer
 - backup disk, 2-23
 - output, assigning to backup disk, 2-23
- printing
 - by job number, 2-26
- procedures
 - CALL_TAPEMANAGER, 5-7, 5-53, 5-54
 - CHECK_TAPE_ASSIGNMENT, 5-25, 5-29
 - CHECK_TAPE_PURGE, 5-30, 5-31
 - CHECK_TAPE_REOPEN, 5-32, 5-33
 - CHECK_TAPE_UNIT_ASSIGNMENT, 5-34, 5-36
 - CONTROL_INTERFACE, 5-38
 - PROGRAM_INTERFACE, 5-39, 5-40
- PROGRAMDUMP option, SEND (Send Message) system command, 2-47
- programmatic file close operations, 1-5
- PROGRAM_INTERFACE procedure, 5-39, 5-40
 - effects if missing, 5-39
- programs
 - pointing to scratch pools, 1-2
- purge operations, for tape files
 - handling by TAPEMANAGER, 5-21
- PURGE REQUEST REJECTED BY TAPEMANAGER system message, 5-31, A-15
- purging
 - tapes
 - asynchronously, 1-8
 - examples of, 1-3
 - synchronously, 1-8
 - tapes, packs, or host control (HC) units, 2-43
- Q**
- quarter-inch tapes, 1-5
- queue

Index

resource limit, 2-26
QUIT option, SEND (Send Message) system
command, 2-47

R

railroad diagrams, explanation of, C-1
READPARTNER specification
host control (HC) unit, 2-4
READY status, unit, 2-12
RECOPY REQD system message, A-15
RECORD SEQUENCE ERROR system
message, A-16
recording TAPEMANAGER library
changes to file attributes, 1-11
reel switches, 1-5
during copy of hazardous files, A-31
in CHECK_TAPE_ASSIGNMENT figure,
5-17
reel-to-reel tapes, 1-10
RELEASE option, SEND (Send Message)
system command, 2-47
reliability, degraded, 2-23
removing HOLD restriction, 1-7
reopen accepted, in CHECK_
TAPE_REOPEN figure,
5-21
REOPEN STOPPED BY TAPEMANAGER
system message
CHECK_TAPE_REOPEN procedures
and, 5-33, A-17
reopening a tape, in CHECK_TAPE_OPEN
figure, 5-20
REPORT_LOG_ENTRIES procedure
role in tape management systems, 5-6
used to monitor tape expirations, 5-23
REQUEST parameter
CONTROL_INTERFACE procedure, 5-38
REQUEST_AND_RESPONSE parameter
CALL_TAPEMANAGER procedure, 5-54
CHECK_TAPE_ASSIGNMENT
procedure, 5-27
CHECK_TAPE_PURGE procedure, 5-30
PROGRAM_INTERFACE procedure,
5-40
REQUIRES VOLUME DELETE system
message, A-17
RESERVED status, unit, 2-12
RESET option, SEND (Send Message)
system command, 2-47
RESOURCECHECK system option, 2-26

resources
limit for queue, 2-26
RESTART ABORTED system message, A-18
RESTRICT (Restrict Unit or Volume)
system command, A-14, A-30
restrictions, HOLD, removing, 1-7
RETAINED_FILE parameter
CHECK_TAPE_REOPEN procedure,
5-32
RETAINED_FILE_TASK parameter
CHECK_TAPE_REOPEN procedure,
5-33
rewinding tapes, 1-1, 1-5
RFE (reserved for expansion) notation, B-1
RFS (reserved by standard) notation, B-1
RSVP requests
AUTORUNNING system option and,
2-24
DIAGNOSTICS system option and, 2-23
FOUT30, 1-4
inappropriate OU response, A-13
issued to operator, 1-2
issued to operator display terminal
(ODT), 1-4
MSRFOUT23, 5-19, A-29
MSRFOUT30, 2-35
MSRLM43, A-14, A-30
MSRMISC204, 5-13, A-30
MSRMISC220, 5-59, A-28
MSRMISC221, 5-9, 5-59, A-8, A-10, A-31
NP support library and, 2-48
OKTIMEANDDATE system option and,
2-25
OU (Output Unit) system command and,
A-13
PG (Purge) system command and, 2-43,
2-45
RY (Ready) system command, 2-2, 2-43,
2-48, 2-57, A-13

S

sample program, TAPEMANAGER library,
5-41
SAVEFACTOR
file attribute
tape expiration and, 5-23
volume attribute, 1-10
scratch pools, 1-1, 1-2
assigning tapes to, 1-3
assigning tapes to, examples of, 1-3

- assignments, overriding, 1-2
- creating as part of tape purges, 1-2
- defining
 - by account number, 1-2
 - by media type, 1-2
 - for archive tapes, 1-2
 - for dump tapes, 1-2
- names, 1-2, 4-4, 4-6, 4-7
- overriding request for, example of, 1-3
- pointing to
 - in DMSII utilities, 1-2
 - in programs, 1-2
 - in system commands, 1-2
- writing program output to, ALGOL
 - example, 1-4
- scratch tapes, 1-1, 1-2, 1-10
 - ANSI69 format, B-8
 - ANSI87 format, B-17
 - format
 - ANSI69, B-6
 - ANSI87, B-17
 - how MCP identifies, B-6
 - logical groups of (scratch pools), 1-1
- SCRATCHPOOL file attribute, 1-3, 3-5, 3-51
 - changing
 - VISIBLE DBS utility AUDIT SCRATCHPOOL command, 1-5, 4-6
 - COPY command of COPYAUDIT utility, 4-6
 - COPY statement of Work Flow Language (WFL), 3-51
 - dump media, 4-4
 - Work Flow Language (WFL), 3-47
- SCRATCHPOOL REJECTED BY
 - TAPEMANAGER system message, 5-31, A-18
- SECONDARY option, VISIBLE DBS utility, 4-7
- SECOPT (Security Options) system
 - command, 1-8
- security
 - attributes, 1-8, 3-42
 - attributes, volume
 - FAMILYOWNER, 3-53
 - SECURITYGUARD, 3-53
 - SECURITYTYPE, 3-53
 - SECURITYUSE, 3-53
 - enhancement software (InfoGuard), 1-1, 1-8
 - enhancements, 1-8, 3-42
 - options
 - SECOPT (Security Options) system
 - command, TAPECHECK, 1-8
 - SECURITY statement
 - TAPEMANAGER library, 5-56
 - Work Flow Language (WFL), 5-56
 - SECURITYGUARD
 - volume attribute, 3-42
 - volume security attribute, 3-53
 - SECURITYLABELS REQUIRE ANSI87 LABELS system message, A-18
 - SECURITYLABELS volume attribute, 1-8, 1-9, 3-8, 3-42, 3-43, 3-51, 3-53
 - Work Flow Language (WFL), 1-8, 3-51
 - SECURITYTYPE
 - volume attribute, 3-42
 - volume security attribute, 3-53
 - SECURITYUSE
 - volume attribute, 3-42
 - volume security attribute, 3-53
 - Semantic Information Manager (SIM)
 - TAPEMANAGER library and, 5-8
 - SEND (Send Message) system command, 2-47
 - and freeze type TAPEMANAGER library, 5-7
 - and TAPEMANAGER security features, 5-11
 - and TERMINATING status of TAPEMANAGER, 5-56
 - causing CONTROL_INTERFACE call, 5-37
 - displaying TAPEMANAGER status, 5-56
 - effect of redundant START messages, 5-15
 - effect on TAPEMANAGER library
 - implementation, 5-7
 - implementing TAPEMANAGER response to QUIT message, 5-14
 - initiating TAPEMANAGER library, 5-56
 - options
 - ANALYZE, 2-47
 - AUTOLOAD, 2-47
 - BLOCKING, 2-47
 - BLOCKTIMEOUT, 2-47
 - BUFFERSIZE, 2-47
 - DASNAME, 2-47
 - DIAGNOSTICS, 2-47
 - DUMP, 2-47
 - FREEINPUTS, 2-47
 - ID, 2-47
 - INITIALREADS, 2-47
 - LOAD, 2-47

Index

- MAXINPUTS, 2-47
- PROGRAMDUMP, 2-47
- QUIT, 2-47
- RELEASE, 2-47
- RESET, 2-47
- SET, 2-47
- TANKING, 2-47
- processes suspended because of QUIT message, 5-59
- QUIT message used for updating TAPEMANAGER version, 5-60
- STATUS message responses generated by MCP, 5-37
- terminating TAPEMANAGER library, 5-57
- text passed to CONTROL_INTERFACE REQUEST parameter, 5-38
- SEND TAPEMANAGER, SETSTATUS call, 3-31
- SEND TAPESERVER, SETSTATUS call, 3-31
- SEND TM QUIT system command, 2-54, 5-14, 5-57, 5-60
- SEND TM START system command, 2-54, 5-13, 5-14, A-27
- SEND TM STATUS system command, 2-55
- SERIAL NUMBER REJECTED BY TAPEMANAGER system message, 5-31, A-19
- serial numbers, 1-1
 - assigning to tapes, 2-56
 - examples of assigning, changing, 1-3
- SERIALNO file attribute, 2-27
- SERIALNO MATCHES BUT TAPE IS LIB. MAINT. TAPE system message, A-19
- SERIALNO MATCHES BUT TAPE IS NOT FIRST REEL system message, A-19
- SERIALNO MATCHES BUT TAPE NOT IN VOL. LIB. system message, A-20
- SERIALNO MISMATCH system message, A-20
- SERIALNO option, Data Management System II (DMSII), 1-5
- SERIALNUMBER system option, 2-27
- SET option, SEND (Send Message) system command, 2-47
- SETSTATUS
 - calls
 - HOLD (Hold Unit for Operator Use), 3-33
 - PG (Purge), 3-35
 - PGL (Purge and Lock), 3-35
 - SEND TAPEMANAGER, 3-31
 - SEND TAPESERVER, 3-31
 - SN (Serial Number), 3-39
 - SNL (Serial Number Locked), 3-39
 - intrinsic
 - calls to, 1-8
 - calls to, parameter bit, 1-8
 - role in tape management systems, 5-6
 - SETSTATUS call (LOGSETSTAT) system log entry, 1-11
- SIM, (See Semantic Information Manager (SIM))
- single-file tapes, unlabeled format, B-1
- single-file, single-volume tapes, ANSI69 format, B-2
- SL (Support Library) system command
 - installing TAPEMANAGER library, 5-55
- SN (Serial Number)
 - SETSTATUS call, 1-8, 3-39
 - system command, 1-3, 2-56
 - causing CHECK_TAPE_PURGE call, 5-30
- SNL (Serial Number Locked)
 - SETSTATUS call, 1-8, 3-39
 - system command, 2-56
- SOURCE parameter
 - CHECK_TAPE_ASSIGNMENT procedure, 5-26
 - CHECK_TAPE_PURGE procedure, 5-30
 - CHECK_TAPE_UNIT_ASSIGNMENT procedure, 5-35
 - CONTROL_INTERFACE procedure, 5-37
- SOURCE TAPE MISSING system message, A-20
- SQLDB, (See Structured Query Language Database (SQLDB))
- standard tape label formats, 1-9, 1-10, B-1
- standards
 - ANSI X3.27-1969 (ANSI69), B-2
 - ANSI X3.27-1987 (ANSI87), 3-43, B-8
- statements, Work Flow Language (WFL)
 - ADD, 1-3, 1-7
 - ARCHIVE, 1-3
 - COPY, 1-3, 1-5, 1-7
 - VOLUME, 1-8, 3-42, 3-51
- statistics
 - end-of-task (EOT), 2-29
- status
 - AUTOUNLOAD, 2-4
 - control
 - inboard and outboard, 2-9

- units, 2-14
- disks, write-enable, 2-4
- HOLD of unit, 2-2, 2-33
- MT, 2-37
- paths, returned by OL system command, 2-12
- peripheral units, 2-37
 - BLASTED, 2-37
 - U, 2-37
 - UNINITIALIZED, 2-37
 - UNLABELED, 2-37
- port, inboard and outboard, 2-9
- system, 2-45
- tape manager, 2-47, 2-55
- tape server, 2-47, 2-54
- TAPEMANAGER library, 5-56, 5-58
 - ACTIVE, 5-14, 5-56
 - GOINGAWAY, 5-15, 5-57, 5-58
 - INACTIVE, 5-13, 5-56
 - INITIALIZING, 5-14, 5-56
 - LINKING, 5-13, 5-14, 5-56
 - TERMINATING, 5-14, 5-56
 - WAITING TO DELINK, 5-14, 5-57
- tapes, write-enable, 2-4
- units
 - BLASTED, 2-37
 - MAINTENANCE, 2-12
 - NOT READY, 2-12
 - OFFLINE, 2-12
 - ONLINE, 2-12
 - READY, 2-12
 - RESERVED, 2-12
 - returned by OL (Display Label and Paths) system command, 2-12
 - U, 2-37
 - UNINITIALIZED, 2-37
 - UNLABELED, 2-37
 - write-enable, 2-4
- volume, system log entries, 3-70
- waiting mix entries, 2-45
- status information retrieval
 - role in TMS design, 5-4
- STATUS_CHANGE_REQUEST procedure
 - role in tape management systems, 5-6
- Structured Query Language Database (SQLDB)
 - TAPEMANAGER library and, 5-8
- supervisor program, 2-23
- support libraries
 - customized TMSs as, 1-12
 - network processor, 2-48
 - TAPEMANAGER, 1-12
- synchronous purges of tapes, 1-8
- SYSOPS (System Options) system command, 1-9, 1-11, 2-59, 3-7, 3-43
 - effect on CALL_TAPEMANAGER procedure, 5-53, 5-54
 - installing TAPEMANAGER library, 5-55
 - TAPEEXPIRATION option, 5-23
 - temporarily suspending use of TAPEMANAGER library, 5-58
- system
 - commands
 - affected by scratch pools, 1-2
 - DS (Discontinue), 5-57, 5-58
 - FA (File Attributes), 5-26
 - FR (Final Reel), 5-29
 - FREE, 2-2
 - HOLD (Hold Unit for Operator Use), 1-7, 1-11, 1-12, 2-2, 2-3, 2-33
 - HOLD (Hold Unit for Operator Use), examples of, 1-7
 - IL (Ignore Label), 5-34, 5-36
 - media library, 2-53
 - MODE (Unit Mode), 1-6, 2-4
 - MODE (Unit Mode), examples of, 1-6
 - network processor, 2-48
 - OL (Display Label and Paths), 2-8
 - OU (Output Unit), 1-3, 2-33, 5-34, 5-35, A-13
 - PER (Peripheral Status), 2-2, 2-37
 - PG (Purge), 1-3, 2-43
 - PGL (Purge and Lock), 2-43
 - pointing to scratch pools, 1-2
 - RESTRICT (Restrict Unit or Volume), A-14, A-30
 - RY (Ready), 2-2, 2-43, 2-48, 2-57, A-13
 - SECOPT (Security Options), 1-8
 - SEND (Send Message), 2-47, 5-7, 5-57
 - SEND TM QUIT, 2-54, 5-14, 5-57, 5-60
 - SEND TM START, 2-54, 5-13, 5-14, A-27
 - SEND TM STATUS, 2-55
 - SN (Serial Number), 1-3, 2-56
 - SNL (Serial Number Locked), 2-56
 - SYSOPS (System Options), 1-9, 1-11, 2-59, 3-7, 3-43
 - TDIR (Tape Directory), 2-3
 - UL (Unlabeled), 5-34, 5-36
 - initialization, 2-24
 - intrinsic, 2-24
 - log, 1-11

log entries

CLU Hardware Report
 (CLUHWREPORT), 1-12
 CLUHWREPORT (CLU Hardware
 Report), 1-12
 File Close (LOGCLOSETYPE), 1-11
 File Open (LOGOPENTYPE), 1-11
 LOGCLOSETYPE (File Close), 1-11
 LOGMAJLIBS (Tape Library Entry),
 3-74
 LOGMAJ_MEDIALIB (Tape Library
 Entry), 1-12
 LOGMAJVOL (Volume Status), 1-11
 LOGOPENTYPE (File Open), 1-11
 LOGSETSTAT (SETSTATUS call),
 1-11
 Minor Type 12 (Tape Volume Used),
 1-11, 3-70, 3-71, 3-72, 3-73, 3-74
 Minor Type 13 (Tape Volume Hold),
 1-11, 3-70, 3-71, 3-72, 3-73, 3-74
 Minor Type 26 (Tape Library Entry),
 1-12, 3-74
 Minor Type 3 (Tape Volume Purged),
 1-12
 Minor Type 5 (Tape Volume
 NEWFILE), 1-12
 SETSTATUS call (LOGSETSTAT),
 1-11
 Tape Library Entry (Minor Type 26),
 1-12, 3-74
 Tape Volume Hold (Minor Type 13),
 1-11, 3-70, 3-71, 3-72, 3-73, 3-74
 Tape Volume New File (Minor Type 5),
 1-12
 Tape Volume Purged (Minor Type 3),
 1-12
 Tape Volume Used File (Minor
 Type 12), 1-11, 3-70, 3-71, 3-72,
 3-73, 3-74
 Volume Status (LOGMAJVOL), 1-11

messages

ARCHIVE BACKUP ENTRY
 DELETED, A-3
 ASSIGNMENT STOPPED BY
 TAPEMANAGER LIBRARY, 5-29,
 A-4
 ATTERR177, 5-36
 BAD DISK HEADER ON TAPE, A-4
 CANNOT QUIT TAPEMANAGER,
 A-5
 CANNOT START TAPEMANAGER,
 A-5

DESTROYED VOLUME NOT ADDED
 IN SCRATCH POOL, A-5
 DISK FILE HEADER VERSION
 ERROR, A-6
 DUPLICATE SERIAL NUMBER, A-33
 END-OF-FILE READING HDR1
 LABEL, A-6
 ERROR CLOSING TAPE FILE, A-6
 ERROR OPENING TAPE FILE, A-6
 ERROR READING TAPE LABEL, A-7
 ERROR WRITING TAPE LABEL OR
 TAPE MARK, A-7
 ERROR: ATTRIBUTE
 MODIFICATION PROHIBITED,
 5-36
 EXPECTED RECORD ID, A-7
 FAMILYOWNER MISMATCH OR
 SECURITY ERROR, A-7
 FAULT REASON =, A-8
 FILE NOT VERIFIED, A-9
 FILE TITLE ERROR, A-10
 FILE VERIFIED, A-10
 <file name> REQUIRES
 <unit identifier>, 2-35, A-17
 <file name> REQUIRES
 <unit identifier>, example of, 1-4
 HISTORY REASON =, CAUSE =,
 TYPE =, A-10
 I/O ERROR ON CHECKSUM
 RECORD, A-12
 I/O ERROR READING HDR1 LABEL,
 A-12
 I/O ERROR WHILE ACCESSING
 GUARDFILE, A-13
 ILLEGAL OPTIONAL FILE, 5-29
 INVALID DENSITY SPECIFIED, A-11
 INVALID LIBRARY/MAINT TAPE
 BLOCKSIZE, A-11
 IO ERROR DURING COMPARE, A-12
 IO ERROR DURING COPY, A-12
 LIBRARY CANNOT OPEN THIS
 FILE, 5-25, 5-33, 5-34, A-13
 MSRCAT11, A-30
 MSRCAT16, A-14
 MSRCAT26, A-19
 MSRCAT27, A-19
 MSRCAT9, A-20
 MSRDIR1, A-21
 MSRDPC177, 5-31, A-15
 MSRDPC181, 5-31, A-19
 MSRDPC183, 5-31, A-18
 MSRDPC93, A-17

MSRDPC96, A-11
MSREHO (TAPE READ), A-24
MSREHO (TAPE WRITE), A-25
MSRFIB16, 5-29
MSRFIB268, 5-33, A-17
MSRFIB270, 5-25, 5-33, 5-34, A-13
MSRFIB271, 5-29, A-4
MSRFIB74, A-15
MSRFOUT11, A-13
MSRFOUT15, A-14
MSRFOUT23, 5-19, A-29
MSRFOUT30, 2-35, A-17
MSRFOUT30, example of, 1-4
MSRFOUT9, A-7
MSRLM0, A-21, A-22
MSRLM1, A-22
MSRLM11, A-10
MSRLM12, A-4
MSRLM15, A-12
MSRLM17, A-21
MSRLM19, A-5
MSRLM2, A-22, A-23
MSRLM21, A-7
MSRLM22, A-30
MSRLM24, A-28
MSRLM26, A-6
MSRLM27, A-4
MSRLM29, A-6
MSRLM3, A-23
MSRLM30, A-12
MSRLM31, A-6
MSRLM34, A-23
MSRLM37, A-6
MSRLM39, A-7
MSRLM4, A-12
MSRLM42, A-13, A-31
MSRLM43, A-14, A-30
MSRLM5, A-15
MSRLM6, A-23
MSRLM69, A-14
MSRLM7, A-16
MSRLM79, A-32
MSRLM8, A-7
MSRLM81, A-24
MSRLM82, A-35
MSRLM84, A-12
MSRLM85, A-20
MSRLM86, A-10
MSRLM88, A-9
MSRLM90, A-20
MSRLM92, A-3
MSRLM96, A-11
MSRMISC117, 2-24
MSRMISC201, A-13
MSRMISC204, 5-13, A-30
MSRMISC205, A-29
MSRMISC206, A-26
MSRMISC207, A-26
MSRMISC208, 5-13, A-27, A-36
MSRMISC209, 5-13, A-27, A-28, A-36
MSRMISC210, 5-13, A-27, A-28, A-36
MSRMISC211, A-26
MSRMISC212, A-5
MSRMISC213, A-5
MSRMISC214, 5-38, A-25
MSRMISC215, 5-38, A-26
MSRMISC220, 5-59, A-28
MSRMISC221, 5-9, 5-59, A-8, A-10,
A-31
MSRMISC222, 5-14, A-27
MSRMISC223, A-8
MSRMISC224, A-10
MSRRS26, A-18
MSRSOFT42, A-18
MSRVL12, A-35
MSRVL16, A-15
MSRVS1, A-34
MSRVS10, A-33
MSRVS11, A-33
MSRVS12, A-33
MSRVS13, A-33
MSRVS14, A-34
MSRVS15, A-33
MSRVS16, A-35
MSRVS17, A-33
MSRVS18, A-33
MSRVS19, A-33
MSRVS2, A-34
MSRVS20, A-33
MSRVS21, A-33
MSRVS22, A-33
MSRVS23, A-33, A-35
MSRVS3, A-34
MSRVS4, A-34
MSRVS5, A-34
MSRVS6, A-33
MSRVS7, A-33
MSRVS8, A-33
MSRVS9, A-33
MSRWRNG56, A-36
MSRWRNG57, A-37
MSRWRNG58, A-37
MSRWRNG59, A-37

- NEED AN OUTPUT TAPE FOR OUMT, A-13
- NEW SOURCE UNIT OR VOLUME IS RESTRICTED, A-13, A-31
- NO CATALOG ACTION (LIBRARY TAPE), A-14
- NO ENTRY FOR VOLUME IN 'VOLUME DIRECTORY', A-14
- NO RECOPY WHEN USING THIS KIND OF TAPE, A-14
- NOT A SCRATCH VOLUME, A-15
- NOT CLOSED, A-15
- PURGE REQUEST REJECTED BY TAPEMANAGER, 5-31, A-15
- RECOPY REQD, A-15
- RECORD SEQUENCE ERROR, A-16
- REOPEN STOPPED BY TAPEMANAGER, 5-33, A-17
- REQUIRES VOLUME DELETE, A-17
- RESTART ABORTED, A-18
- SCRATCHPOOL REJECTED BY TAPEMANAGER, 5-31, A-18
- SECURITYLABELS REQUIRE ANS187 LABELS, A-18
- SERIAL NUMBER REJECTED BY TAPEMANAGER, 5-31, A-19
- SERIALNO MATCHES BUT TAPE IS A LIBRARY MAINTENANCE TAPE, A-19
- SERIALNO MATCHES BUT TAPE IS NOT FIRST REEL, A-19
- TAPE DISKHEADER READ ERROR, A-22, A-23
- TAPE DISKHEADER WRITE ERROR, A-23
- TAPE DOESN'T SEEM TO BACKSPACE, A-4
- TAPE FILE MISSING, A-23
- TAPE POSITIONING ERROR, A-23
- TAPE READ ERROR, A-24
- TAPE WILL BE PURGED, A-24
- TAPE WRITE ERROR, A-25
- TAPEMANAGER FAILED IN INITIALIZATION PROCEDURE, 5-38, A-25
- TAPEMANAGER FAILED IN TERMINATION PROCEDURE, 5-38, A-26
- TAPEMANAGER IS ACTIVE, A-26
- TAPEMANAGER IS INACTIVE, A-26
- TAPEMANAGER LIBRARY HAS LINKED, A-26
- TAPEMANAGER LIBRARY IS ALREADY ACTIVE, 5-14, A-27
- TAPEMANAGER LIBRARY IS MISSING CONTROL_INTERFACE, 5-13, A-27
- TAPEMANAGER LIBRARY WILL NOT BE USED, 5-13, A-27, A-28
- TAPEMANAGER REQUIRED, 5-59, A-28
- TITLE FOUND WAS, A-28
- TYPE DISAGREES WITH KIND, 5-19, A-29
- UNABLE TO LINK TO TAPEMANAGER, 5-13, A-30
- UNABLE TO LINK TO TAPEMANAGER LIBRARY, A-29, A-30
- UNEXPECTED TAPE MARK, A-30
- UNIT NOT VOLUMED, A-30
- UNRESTRICT SOURCE AND REPLY OK TO COPY, A-14, A-30
- USER PROCESS FAULTED OR DSED IN TAPEMANAGER, 5-9, 5-59, A-8, A-10, A-31
- VERIFY ERROR, A-32
- VOLUME DIRECTORY, A-33
- VOLUME DIRECTORY <action> NOT DONE, A-35
- VOLUME DIRECTORY FAULT IN VSHANDLER, A-34
- VOLUME DIRECTORY NOT SNED, A-35
- VOLUME DIRECTORY RECORD HAS INVALID DATA, A-34
- VOLUME FAMILY NOT DELETED, A-35
- WARNING 56, A-36
- WARNING 57, A-37
- WARNING 58, A-37
- WARNING 59, A-37
- WARNING, TAPE WAS NOT WRITTEN WITH CHECKSUMS, A-35
- WARNING: TAPEMANAGER LIBRARY IS MISSING OBJECTS, 5-13, A-27, A-28, A-36
- 'OP AUTORUNNING' IS SET, REPLY <reply> ASSUMED, 2-24
- operations, automatic tape unloading and, 1-6
- options, 2-21, 2-59
- ARCHIVING, 2-27

AUDIT, 2-28
 AUTODC, 2-24
 AUTORECOVERY, 2-23
 AUTORM, 2-23
 AUTORUNNING, 2-24
 BACKUPBYJOBNR, 2-26
 CATALOGING, 2-26
 CATTEST, 2-30
 CDONLY, 2-23
 CONVENTION, 2-60
 CPBDONLY, 2-24
 CRUNCH, 2-25
 DIAGNOSTICDUMP, 2-28
 DIAGNOSTICS, 2-23
 DIRDEBUG, 2-26
 DUPINTRINSICS, 2-24
 DUPSUPERVISOR, 2-23
 EOTSTATISTICS, 2-29
 FILESATURATION, 2-28
 IODIAGNOSTICS, 2-30
 IORANGECHECK, 2-27
 ISCDEBUG, 2-30
 KEYEDIOIL, 2-28
 LABELFORMAT, 1-9, 1-10, 2-60, 3-7,
 3-43
 LABELFORMAT = ANSI87DEFAULT,
 1-9
 LANGUAGE, 2-60
 LOCKTRACE, 2-27
 LOGIOERRORSNEW, 2-29
 LOGPOSITIONING, 2-27
 LPBDONLY, 2-23
 MCPTEST, 2-30
 MIRRORING, 2-28
 NETRECOVERY, 2-29
 NOCHECK, 2-23
 NODUMP, 2-24
 NOFETCH, 2-26
 NOSUMMARY, 2-26
 OKTIMEANDDATE, 2-26
 OPEN, 2-22, 2-60
 PATHBALANCING, 2-29
 PDTODISK, 2-26
 PORTDEBUG, 2-30
 RESOURCECHECK, 2-26
 SERIALNUMBER, 2-27
 TAPEEXPIRATION, 1-11, 2-60
 TAPEEXPIRATION, changing setting
 of, 1-11
 TAPEMANAGER, 2-60
 TAPESERVER, 2-60
 TERMINATE, 2-22

TRANSWARNINGS, 2-24
 USECATDEFAULT, 2-30
 status, 2-45
 SYSTEMSTATUS function
 role in tape management systems, 5-6

T

table of contents, control file, 4-2
 TANKING option, SEND (Send Message)
 system command, 2-47
 tape
 assigning density, 2-56
 assigning serial number to, 2-56
 creation
 MCP Mark release active at, B-6
 density as criterion for purging, 2-44
 drives, automatic-unload mode, 1-6
 locking, 2-57
 purging, 2-56
 retrying errors, 2-27
 scratch, assigning, 2-27
 TAPE ABANDON FILEHANDLER system
 message, A-21
 tape assigned, in CHECK_TAPE_REOPEN
 figure, 5-21
 tape data, reading from expired tapes, 1-10
 TAPE DIRECTORY COMPARE ERROR
 system message, A-21
 TAPE DIRECTORY READ ERROR system
 message, A-21, A-22
 TAPE DIRECTORY WRITE ERROR system
 message, A-22
 TAPE DISKHEADER READ ERROR system
 message, A-22, A-23
 TAPE DISKHEADER WRITE ERROR
 system message, A-23
 TAPE DOESN'T SEEM TO BACKSPACE
 system message, A-4
 tape drives, 1-1
 performing reel switches, default
 procedure, 1-5
 upon performing reel switches, 1-5
 when released from use, 1-5
 when released from use, default
 procedure, 1-5
 TAPE FILE MISSING system message,
 A-23
 tape labels
 examining before tapes assigned to tasks,
 1-7

- formats
 - ANSI69, 1-9, 1-10, 3-8, B-2
 - ANSI87, 1-9, 1-10, 3-8, 3-43, B-8
 - B 3500, B-17
 - B 5500, B-19, B-20
 - USASI B 3500, B-17
- specifying expiration date, 1-10
- Tape Library Entry (Minor Type 26) system
 - log entry, 1-12, 3-74
- tape management
 - operations, unattended, 1-5
- tape management system
 - sample program, 5-41
- tape management system (TMS), 5-1
 - creating, 1-12
 - definition of, 1-12
 - event and status information, 5-4
- tape manager status, 2-47, 2-55
- tape mark notation, tape label illustrations,
 - B-1
- TAPE POSITIONING ERROR system
 - message, A-23
- TAPE READ ERROR system message, A-24
- tape server status, 2-47, 2-54
- tape usage
 - controlling with TMS, 1-13
- Tape Volume Hold (Minor Type 13) system
 - log entry, 1-11, 3-70, 3-71, 3-72, 3-73, 3-74
- Tape Volume New File (Minor Type 5)
 - system log entry, 1-12
- Tape Volume Purged (Minor Type 3) system
 - log entry, 1-12
- Tape Volume Used File (Minor Type 12)
 - system log entry, 1-11, 3-70, 3-71, 3-72, 3-73, 3-74
- TAPE WILL BE PURGED system message,
 - A-24
- TAPE WRITE ERROR system message,
 - A-25
- TAPECHECK security option, 1-8
- TAPEEXPIRATION (SYSOPS option)
 - tape management systems and, 5-23
- TAPEEXPIRATION system option, 1-11, 2-60
- TAPEMANAGER FAILED IN TERMINATION PROCEDURE
 - system message, 5-38, A-26
- TAPEMANAGER FAILED INITIALIZATION PROCEDURE
 - system message, 5-38, A-25
- TAPEMANAGER function name, 1-12
- TAPEMANAGER IS ACTIVE system
 - message, A-26
- TAPEMANAGER IS INACTIVE system
 - message, A-26
- TAPEMANAGER library
 - binding procedures into library, 5-7
 - changes to file attributes, recording, 1-11
 - exported procedures, 1-12, 1-13, 5-23
 - CHECK_TAPE_ASSIGNMENT, 5-25
 - CHECK_TAPE_PURGE, 5-30
 - CHECK_TAPE_REOPEN, 5-32
 - CHECK_TAPE_UNIT_ASSIGNMENT, 5-34
 - CONTROL_INTERFACE, 5-37
 - effect of missing procedures, 5-13, 5-24
 - PROGRAM_INTERFACE, 5-39
- fault handling, 5-8
 - by TAPEMANAGER itself, 5-9
 - by the MCP, 5-9
- freeze type, 5-7
- general considerations, 5-7
- general functions
 - handling expired tapes, 5-23
 - handling programmatic calls, 5-23
 - opening files, 5-15
 - purging tape volumes, 5-21
 - shutting down, 5-14
 - starting up, 5-12
- implementation of, 5-1
- initiated only by MCP, 5-10
- interaction with MCP, 5-11
- invoked for nontape file opens, 5-17
- marking with an identity, 5-7, 5-56
- only one version allowed, 5-10
- opening tapes
 - on a specific unit, 5-18
 - reopening tapes, 5-20
- operations interface, 5-55
 - initiating the library, 5-56
 - inspecting library status, 5-56
 - installing the library, 5-55
 - replacing library with a new version, 5-60
 - responding to waiting processes, 5-59
 - suspending use of the library, 5-58
 - terminating the library, 5-57
- protected from DS commands, 5-57
- restrictions on database usage, 5-8
- sample program, 5-41
- security considerations, 5-10
 - making code file private, 5-56
- status, 5-56, 5-58

- GOINGAWAY, 5-58
- TERMINATING, 5-58
- WAITING TO DELINK, 5-58
 - status changes, 5-11
 - written in ALGOL or NEWP only, 5-7
- TAPEMANAGER LIBRARY HAS LINKED
 - system message, A-26
- TAPEMANAGER LIBRARY IS ALREADY
 - ACTIVE system message, 5-14, A-27
- TAPEMANAGER LIBRARY IS MISSING ...
 - system message, 5-13, A-27
- TAPEMANAGER LIBRARY WILL NOT BE
 - USED system message, 5-13, A-27, A-28
- TAPEMANAGER REQUIRED system message, 5-59, A-28
- TAPEMANAGER system option, 2-60
- TAPEMANAGER_INITIALIZATION
 - independent runner
 - and TAPEMANAGER library implementation, 5-13
 - calling CONTROL_INTERFACE procedure, 5-14
 - using RSVP message when linkage fails, 5-13
- TAPEMANAGER_TERMINATION
 - independent runner
 - actions taken to terminate TAPEMANAGER, 5-14
 - invoking CONTROL_INTERFACE procedure, 5-37
- tapes
 - ANSI69 format, 1-9, 1-10, 3-8, B-2
 - ANSI87 format, 1-9, 1-10, 3-8, 3-43, B-8, B-9, B-10, B-11, B-12, B-14, B-15, B-16, B-17
 - creating, 3-43
 - archive, defining scratch pools for, 1-2
 - B 5500 format, B-19, B-20
 - cartridge
 - file protection mechanisms, 1-10
 - Unisys HD600XTD, HD6150, 1-10
 - write-enable status, 1-10
 - dump, defining scratch pools for, 1-2
 - examining before MCP assigns to tasks, 1-7
 - examples of, 1-7
 - formats
 - ANSI69, B-2
 - ANSI87, 3-43, B-8
 - B 5500, B-19, B-20
 - standard, B-1
 - unlabeled, multifile, B-1
 - unlabeled, single-file, B-1
 - USASI B 3500, B-17
 - grouped under common identifier, 1-2
 - identifying
 - by scratch pool name, 1-1
 - by serial number, 1-1
 - label formats, standard, B-1
 - locking, 2-43
 - multifile, multivolume
 - ANSI69 format, B-3
 - ANSI87 format, B-10
 - purging, 1-2, 2-43
 - asynchronously, 1-8
 - examples of, 1-3
 - synchronously, 1-8
 - reel-to-reel
 - write-enable status, 1-10
 - rewinding, 1-1, 1-5
 - scratch, 1-1, 1-2, 1-10
 - ANSI69 format, B-6, B-8
 - serial numbers, examples of changing, assigning, 1-3
 - single-file, single-volume
 - ANSI69 format, B-2
 - standard formats, B-1
 - unlabeled
 - multifile, B-1
 - single-file, B-1
 - USASI B 3500 format, B-17
 - write-enable status, 2-4
 - 5072 cartridge
 - write-enable status, 1-10
- TAPESEVER system option, 2-60
- task
 - abnormal termination, 2-22
 - assignment, 1-7
 - attributes
 - LABELFORMAT, 1-9, 1-10, 3-7, 3-43
 - EOT statistics, 2-29
 - output, 1-10
 - directing to device, 2-33
- TDIR (Tape Directory) system command, 2-3
- TERMINATE system option, 2-22
- TERMINATING status, TAPEMANAGER
 - library, 5-14, 5-56, 5-58
- time, operator verification of, 2-26
- TITLE FOUND WAS system message, A-28
- TMS, (See tape management system (TMS))

transaction state
 TAPEMANAGER library and, 5-8
 TRANSWARNINGS system option, 2-24
 TYPE DISAGREES WITH KIND system
 message, 5-19, A-29

U

U status
 peripheral units, 2-37
 units, 2-37
 UL (Unlabeled) system command, 5-36
 causing CHECK_TAPE_UNIT_
 ASSIGNMENT call,
 5-34
 detecting as source of UNITNO
 assignment, 5-20
 UNABLE TO LINK TO TAPEMANAGER
 LIBRARY system message, 5-13,
 A-29, A-30
 unattended tape management operations,
 1-5
 UNEXPECTED TAPE MARK system
 message, A-30
 UNINITIALIZED status
 peripheral units, 2-37
 units, 2-37
 Unisys HD600XTD, HD6150 tape cartridges
 write-protect plug, 1-10
 UNIT NOT VOLUMED system message,
 A-30
 unit requests, SETSTATUS
 HOLD (Hold Unit for Operator Use), 3-33
 PG (Purge), 3-35
 PGL (Purge and Lock), 3-35
 SEND TAPEMANAGER, 3-31
 SEND TAPESERVER, 3-31
 SN (Serial Number), 3-39
 SNL (Serial Number Locked), 3-39
 UNITNO specified, in CHECK_TAPE_UNIT_
 ASSIGNMENT figure, 5-18
 units
 degraded reliability, 2-23
 status
 BLASTED, 2-37
 MAINTENANCE, 2-12
 NOT READY, 2-12
 OFFLINE, 2-12
 ONLINE, 2-12
 READY, 2-12
 RESERVED, 2-12

U, 2-37
 UNINITIALIZED, 2-37
 UNLABELED, 2-37
 write-enable status, 2-4
 UNLABELED status
 peripheral units, 2-37
 units, 2-37
 unlabeled tapes, formats
 multifile, B-1
 single-file, B-1
 UNLINKED_ACTION parameter
 CALL TAPEMANAGER procedure, 5-54
 UNRESTRICT SOURCE AND REPLY OK
 TO COPY system message, A-14,
 A-30
 UPDATE statement, DMCONTROL, 4-3
 USASI format
 B 3500 file header 1, B-17
 B 3500 tape labels, B-17
 USECATALOG file attribute, 2-30
 USECATDEFAULT system option, 2-30
 user header labels, ANSI69 tapes, B-2, B-8
 USER PROCESS FAULTED OR DSED IN
 TAPEMANAGER system message,
 5-9, 5-59, A-8, A-10, A-31
 user trailer labels, ANSI69 tapes, B-2, B-8
 utilities
 COPYAUDIT, 4-4, 4-6
 Data Management System II (DMSII),
 1-2, 4-2
 DUMPALL, 1-4
 FILECOPY, 1-4, 1-7
 using SCRATCHPOOL file attribute in,
 1-4
 VISIBLE DBS, 1-5, 4-6

V

VERIFY ERROR system message, A-32
 <visible dbs command>, 4-6
 VISIBLE DBS utility, 1-5, 4-6
 VOL (volume header) notation, tape label
 illustrations, B-1
 volume attributes
 FAMILYOWNER, 3-42
 SAVEFACTOR, 1-10
 SECURITYGUARD, 3-42
 SECURITYLABELS, 1-8, 1-9, 3-8, 3-42,
 3-43, 3-51, 3-53
 SECURITYTYPE, 3-42
 SECURITYUSE, 3-42

- VOLUME DIRECTORY <action> NOT DONE system message, A-35
- VOLUME DIRECTORY FAULT IN VSHANDLER system message, A-34
- VOLUME DIRECTORY NOT SNED system message, A-35
- VOLUME DIRECTORY RECORD HAS INVALID DATA system message, A-34
- VOLUME DIRECTORY system message, A-33
- VOLUME FAMILY NOT DELETED system message, A-35
- volume header format
- ANSI69 tapes, B-4
 - USASI B 3500 tapes, B-17
- volume header 1 format
- ANSI87 tapes, B-11
- volume header 2 format
- ANSI87 tapes, B-12, B-14
- volume header 3 format
- ANSI87 tapes, B-12, B-14
- VOLUME statement
- Work Flow Language (WFL), 3-51
 - SECURITYLABELS volume attribute in, 1-8, 3-42
- Volume Status (LOGMAJVOL) system log entry, 1-11
- volumes
- multifile
 - ANSI69 tape format, B-2
 - ANSI87 tape format, B-9, B-10 - status
 - system log entries, 3-70
- WFL, (See Work Flow Language (WFL))
- Work Flow Language (WFL), 1-3, 3-43
- file attributes
 - AUTOUNLOAD, 3-47, 3-51
 - SCRATCHPOOL, 3-47, 3-51 - statements, 1-7
 - ADD, 1-2, 1-3, 1-7, 3-47
 - ARCHIVE, 1-2, 1-3, 3-45, 3-47
 - COPY, 1-2, 1-3, 1-5, 1-7, 3-47
 - FETCH, 2-26
 - VOLUME, 1-8, 3-42, 3-51 - volume attributes
 - FAMILYOWNER, 3-42
 - SECURITYGUARD, 3-42
 - SECURITYLABELS, 1-8, 3-42, 3-51, 3-53
 - SECURITYTYPE, 3-42
 - SECURITYUSE, 3-42 - write rings, 1-10
 - write-enable status
 - disks, 2-4
 - tapes, 1-10
 - units, 2-4 - write-protect plug
 - Unisys HD600XTD, HD6150 tape cartridges, 1-10
- WRITEPARTNER specification
- host control (HC) unit, 2-4
- writing output to tape, 1-9, 1-10
- 5073 cartridge tapes
- write enable status, 1-10
- W**
- waiting mix entries, status of, 2-45
- WAITING TO DELINK status,
- TAPEMANAGER library, 5-14, 5-57, 5-58
- WARNING 56 system message, A-36
- WARNING 57 system message, A-37
- WARNING 58 system message, A-37
- WARNING 59 system message, A-37
- WARNING, TAPE WAS NOT WRITTEN WITH CHECKSUMS system message, A-35
- WARNING: TAPEMANAGER LIBRARY MISSING OBJECTS system message, 5-13, A-27, A-28, A-36



86000940-000