

# HP NonStop TMF Reference Manual

## Abstract

This manual describes how to use the TMFCOM command interface to the HP NonStop™ Transaction Management Facility (TMF). This manual includes command syntax, semantics, and examples, and is intended for system managers and operators.

## Product Version

TMF H01

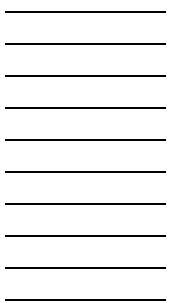
## Supported Releases

This publication supports H06.03 and all subsequent H-series release version updates (RVUs) until otherwise indicated by its replacement publication.

| Part Number | Published |
|-------------|-----------|
| 540138-001  | July 2005 |

## Document History

| Part Number | Product Version | Published |
|-------------|-----------------|-----------|
| 540138-001  | TMF H01         | July 2005 |



# HP NonStop TMF Reference Manual

| Index | Figures | Tables |
|-------|---------|--------|
|-------|---------|--------|

- [What's New in This Manual](#)   vii
  - [Manual Information](#)   vii
  - [New and Changed Information](#)   vii
- [About This Manual](#)   ix
  - [Who Should Read This Manual](#)   ix
  - [How This Manual Is Organized](#)   ix
  - [TMF Documentation](#)   x
  - [Other Documentation](#)   xi
  - [Notation Conventions](#)   xii

## 1. Introduction to TMF

- [About TMF](#)   1-1
- [TMF Objects](#)   1-2
- [TMFCOM Command Interpreter](#)   1-4

## 2. Using TMFCOM

- [Running TMFCOM](#)   2-1
  - [Command Syntax for Starting a Session](#)   2-1
  - [Using TMFCOM Interactively](#)   2-2
  - [Using TMFCOM Noninteractively](#)   2-4
  - [Using TMFCOM From a Command File](#)   2-5
- [Entering TMFCOM Commands](#)   2-6
  - [Including Comments in Command Lines](#)   2-6
  - [Continuing Commands on Multiple Lines](#)   2-7
  - [Entering Multiple Commands on a Line](#)   2-7
- [Directing Output to an Alternate Destination](#)   2-8
- [Using Partially Qualified File Names](#)   2-9
- [Requesting Online Help](#)   2-10

### **3. TMFCOM Commands**

|   |       |
|---|-------|
| <a href="#"><u>Command Description Elements</u></a>       | 3-1   |
| <a href="#"><u>Purpose, Syntax, and Parameters</u></a>    | 3-2   |
| <a href="#"><u>Security Restrictions</u></a>              | 3-2   |
| <a href="#"><u>TMF State Requirement</u></a>              | 3-2   |
| <a href="#"><u>Usage Guidelines</u></a>                   | 3-6   |
| <a href="#"><u>Output Displayed</u></a>                   | 3-6   |
| <a href="#"><u>Examples</u></a>                           | 3-6   |
| <a href="#"><u>File Names and Process Identifiers</u></a> | 3-6   |
| <a href="#"><u>Disk File Names</u></a>                    | 3-7   |
| <a href="#"><u>Nondisk Device Names</u></a>               | 3-8   |
| <a href="#"><u>Process Identifiers</u></a>                | 3-9   |
| <a href="#"><u>ABORT TRANSACTION</u></a>                  | 3-10  |
| <a href="#"><u>ADD AUDITTRAIL</u></a>                     | 3-14  |
| <a href="#"><u>ADD DATAVOLS</u></a>                       | 3-20  |
| <a href="#"><u>ADD DUMPS</u></a>                          | 3-24  |
| <a href="#"><u>ADD RESOURCEMANAGER</u></a>                | 3-36  |
| <a href="#"><u>ADD TAPEMEDIA</u></a>                      | 3-39  |
| <a href="#"><u>ALTER AUDITDUMP</u></a>                    | 3-43  |
| <a href="#"><u>ALTER AUDITTRAIL</u></a>                   | 3-51  |
| <a href="#"><u>ALTER BEGINTRANS</u></a>                   | 3-61  |
| <a href="#"><u>ALTER CATALOG</u></a>                      | 3-67  |
| <a href="#"><u>ALTER DATAVOLS</u></a>                     | 3-71  |
| <a href="#"><u>ALTER DUMPS</u></a>                        | 3-73  |
| <a href="#"><u>ALTER PROCESS</u></a>                      | 3-75  |
| <a href="#"><u>ALTER TAPEMEDIA</u></a>                    | 3-92  |
| <a href="#"><u>ALTER TMF</u></a>                          | 3-94  |
| <a href="#"><u>CANCEL OPERATIONS</u></a>                  | 3-96  |
| <a href="#"><u>CLOSE RESOURCEMANAGER</u></a>              | 3-98  |
| <a href="#"><u>CMDVOL</u></a>                             | 3-100 |
| <a href="#"><u>DELETE CATALOG</u></a>                     | 3-102 |
| <a href="#"><u>DELETE DATAVOLS</u></a>                    | 3-103 |
| <a href="#"><u>DELETE DUMPS</u></a>                       | 3-105 |
| <a href="#"><u>DELETE RESOURCEMANAGER</u></a>             | 3-108 |
| <a href="#"><u>DELETE TAPEMEDIA</u></a>                   | 3-110 |
| <a href="#"><u>DELETE TMF</u></a>                         | 3-111 |
| <a href="#"><u>DELETE TRANSACTION</u></a>                 | 3-113 |
| <a href="#"><u>DISABLE AUDITDUMP</u></a>                  | 3-116 |
| <a href="#"><u>DISABLE BEGINTRANS</u></a>                 | 3-118 |

### **3. TMFCOM Commands (continued)**

|   |       |
|---|-------|
| <a href="#"><u>DISABLE DATAVOLS</u></a>     | 3-119 |
| <a href="#"><u>DISPLAY OPERATIONS</u></a>   | 3-121 |
| <a href="#"><u>DUMP FILES</u></a>           | 3-125 |
| <a href="#"><u>ENABLE AUDITDUMP</u></a>     | 3-135 |
| <a href="#"><u>ENABLE BEGINTRANS</u></a>    | 3-136 |
| <a href="#"><u>ENABLE DATAVOLS</u></a>      | 3-137 |
| <a href="#"><u>ENV</u></a>                  | 3-139 |
| <a href="#"><u>EXIT</u></a>                 | 3-141 |
| <a href="#"><u>FC</u></a>                   | 3-142 |
| <a href="#"><u>HELP</u></a>                 | 3-145 |
| <a href="#"><u>HISTORY</u></a>              | 3-147 |
| <a href="#"><u>INFO AUDITDUMP</u></a>       | 3-150 |
| <a href="#"><u>INFO AUDITTRAIL</u></a>      | 3-152 |
| <a href="#"><u>INFO BEGINTRANS</u></a>      | 3-156 |
| <a href="#"><u>INFO CATALOG</u></a>         | 3-159 |
| <a href="#"><u>INFO DATAVOLS</u></a>        | 3-161 |
| <a href="#"><u>INFO DUMPS</u></a>           | 3-165 |
| <a href="#"><u>INFO PROCESS</u></a>         | 3-172 |
| <a href="#"><u>INFO RESOURCEMANAGER</u></a> | 3-176 |
| <a href="#"><u>INFO TAPEMEDIA</u></a>       | 3-179 |
| <a href="#"><u>INFO TMF</u></a>             | 3-182 |
| <a href="#"><u>NEXT AUDITTRAIL</u></a>      | 3-189 |
| <a href="#"><u>OBEY</u></a>                 | 3-190 |
| <a href="#"><u>OBEYVOL</u></a>              | 3-192 |
| <a href="#"><u>OPEN SERVER</u></a>          | 3-194 |
| <a href="#"><u>RECOVER FILES</u></a>        | 3-201 |
| <a href="#"><u>RELOCATE DISKDUMPS</u></a>   | 3-217 |
| <a href="#"><u>RESET AUDITTRAIL</u></a>     | 3-219 |
| <a href="#"><u>RESET DATAVOLS</u></a>       | 3-222 |
| <a href="#"><u>RESET DUMPS</u></a>          | 3-224 |
| <a href="#"><u>RESET TAPEMEDIA</u></a>      | 3-226 |
| <a href="#"><u>RESOLVE TRANSACTION</u></a>  | 3-228 |
| <a href="#"><u>SET AUDITTRAIL</u></a>       | 3-232 |
| <a href="#"><u>SET DATAVOLS</u></a>         | 3-236 |
| <a href="#"><u>SET DUMPS</u></a>            | 3-238 |
| <a href="#"><u>SET TAPEMEDIA</u></a>        | 3-243 |
| <a href="#"><u>SHOW AUDITTRAIL</u></a>      | 3-247 |
| <a href="#"><u>SHOW DATAVOLS</u></a>        | 3-250 |

### **3. TMFCOM Commands (continued)**

|  |       |
|--|-------|
| <a href="#">SHOW DUMPS</a>             | 3-252 |
| <a href="#">SHOW TAPEMEDIA</a>         | 3-254 |
| <a href="#">START TMF</a>              | 3-256 |
| <a href="#">STATUS AUDITDUMP</a>       | 3-259 |
| <a href="#">STATUS AUDITTRAIL</a>      | 3-261 |
| <a href="#">STATUS BEGINTRANS</a>      | 3-268 |
| <a href="#">STATUS CATALOG</a>         | 3-269 |
| <a href="#">STATUS DATAVOLS</a>        | 3-270 |
| <a href="#">STATUS OPERATIONS</a>      | 3-276 |
| <a href="#">STATUS RESOURCEMANAGER</a> | 3-282 |
| <a href="#">STATUS SERVER</a>          | 3-289 |
| <a href="#">STATUS TMF</a>             | 3-291 |
| <a href="#">STATUS TRANSACTIONS</a>    | 3-296 |
| <a href="#">STOP TMF</a>               | 3-305 |
| <a href="#">VOLUME</a>                 | 3-309 |
| <a href="#">!</a>                      | 3-311 |
| <a href="#">?</a>                      | 3-314 |

### **4. TMFCOM Tape Media Considerations**

|  |     |
|--|-----|
| <a href="#">Tape Density Settings</a>                      | 4-1 |
| <a href="#">Open-Reel Tape Drives</a>                      | 4-1 |
| <a href="#">Other Types of Tape Drives and Devices</a>     | 4-2 |
| <a href="#">Labeled-Tape Processing Interfaces</a>         | 4-2 |
| <a href="#">Additional Guidelines for Using Tape Media</a> | 4-3 |
| <a href="#">Monitoring the Tape Drives</a>                 | 4-3 |
| <a href="#">Using the VERIFYTAPE Option</a>                | 4-4 |
| <a href="#">Using the COPIES Option</a>                    | 4-4 |
| <a href="#">Choosing the Proper BLOCKSIZE</a>              | 4-4 |
| <a href="#">Selecting a Tape Drive Model Type</a>          | 4-4 |
| <a href="#">Considering File Size Factors</a>              | 4-5 |

### **A. TMFCOM Messages**

|   |     |
|---|-----|
| <a href="#">TMFCOM Message Overview</a> | A-1 |
| <a href="#">Message Types</a>           | A-2 |
| <a href="#">Message Numbers</a>         | A-2 |
| <a href="#">Error Codes in Messages</a> | A-2 |
| <a href="#">File Names in Messages</a>  | A-3 |
| <a href="#">Error Identifiers</a>       | A-3 |

## **A. TMFCOM Messages (continued)**

[Other Messages](#) A-4

[Error Information and Failure Data Capture](#) A-4

[Message Descriptions](#) A-6

## **B. TMFCOM Command Reference Summary**

## **C. TMF Configuration Limits and Defaults**

## **Index**

## **Figures**

[Figure 3-1. TMFCOM and TMFSERVE Processes](#) 3-196

[Figure 3-2. TMFSERVE on a Remote Node](#) 3-198

## **Tables**

[Table 1-1. TMF Objects](#) 1-2

[Table 3-1. User Security Restrictions and TMF State Requirements for TMFCOM Commands](#) 3-2

[Table 3-2. Stopping and Restarting TMF Processes](#) 3-89

[Table 3-3. DUMP FILES Output](#) 3-133

[Table 3-4. ENV Output](#) 3-139

[Table 3-5. HISTORY Output](#) 3-148

[Table 3-6. INFO AUDITDUMP Output](#) 3-151

[Table 3-7. INFO AUDITTRAIL Output](#) 3-153

[Table 3-8. INFO BEGINTRANS Output](#) 3-157

[Table 3-9. INFO CATALOG Output](#) 3-160

[Table 3-10. INFO DATAVOLS Output](#) 3-162

[Table 3-11. INFO DUMPS Output](#) 3-170

[Table 3-12. INFO PROCESS Output](#) 3-173

[Table 3-13. INFO RESOURCEMANAGER Output](#) 3-177

[Table 3-14. INFO TAPEMEDIA Output](#) 3-180

[Table 3-15. TMFCOM vs TMFSERVE Version Interaction: Operational Commands](#) 3-199

[Table 3-16. TMFCOM vs TMFSERVE Version Interaction: Informational Commands](#) 3-199

[Table 3-17. Resolving Distributed Transactions](#) 3-230

[Table 3-18. STATUS AUDITDUMP Output](#) 3-260

[Table 3-19. STATUS AUDITTRAIL Output](#) 3-263

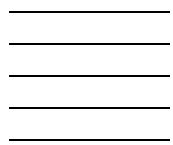
[Table 3-20. STATUS DATAVOLS Output](#) 3-275

[Table 3-21. STATUS OPERATIONS Output](#) 3-280

**Tables (continued)**

|                             |  |       |
|-----------------------------|--|-------|
| <a href="#">Table 3-22.</a> | <a href="#">STATUS RESOURCEMANAGER Output</a>                  | 3-286 |
| <a href="#">Table 3-23.</a> | <a href="#">STATUS SERVER Output</a>                           | 3-289 |
| <a href="#">Table 3-24.</a> | <a href="#">STATUS TMF Output</a>                              | 3-293 |
| <a href="#">Table 3-25.</a> | <a href="#">STATUS TRANSACTIONS Output</a>                     | 3-301 |
| <a href="#">Table C-1.</a>  | <a href="#">TMF Global Limits and Defaults</a>                 | C-1   |
| <a href="#">Table C-2.</a>  | <a href="#">Audit Trail Limits and Defaults</a>                | C-2   |
| <a href="#">Table C-3.</a>  | <a href="#">Transaction Limits and Defaults</a>                | C-2   |
| <a href="#">Table C-4.</a>  | <a href="#">Audit Dump and Online Dump Limits and Defaults</a> | C-3   |
| <a href="#">Table C-5.</a>  | <a href="#">Resource Manager Limits and Defaults</a>           | C-3   |
| <a href="#">Table C-6.</a>  | <a href="#">Process Limits and Defaults</a>                    | C-3   |
| <a href="#">Table C-7.</a>  | <a href="#">Default Program File Names for Processes</a>       | C-5   |





# What's New in This Manual

## Manual Information

### Abstract

This manual describes how to use the TMFCOM command interface to the HP NonStop™ Transaction Management Facility (TMF). This manual includes command syntax, semantics, and examples, and is intended for system managers and operators.

### Product Version

TMF H01

### Supported Releases

This publication supports H06.03 and all subsequent H-series release version updates (RVUs) until otherwise indicated by its replacement publication.

| Part Number | Published |
|-------------|-----------|
| 540138-001  | July 2005 |

### Document History

| Part Number | Product Version | Published |
|-------------|-----------------|-----------|
| 540138-001  | TMF H01         | July 2005 |

## New and Changed Information

This is a new manual.





# About This Manual

This manual describes the syntax and semantics of all TMFCOM commands for using the HP NonStop™ Transaction Management Facility (TMF) on HP NonStop servers.

To learn more about this manual, read the following topics:

| Topic  | Page                |
|--|---------------------|
| <a href="#">Who Should Read This Manual</a>  | <a href="#">ix</a>  |
| <a href="#">How This Manual Is Organized</a> | <a href="#">ix</a>  |
| <a href="#">TMF Documentation</a>            | <a href="#">x</a>   |
| <a href="#">Other Documentation</a>          | <a href="#">x</a>   |
| <a href="#">Notation Conventions</a>         | <a href="#">xii</a> |

## Who Should Read This Manual

This manual is intended for system managers and system operators responsible for configuring, maintaining, and overseeing the use of TMF on their computer systems.

To use this manual effectively, you should be familiar with NonStop server architecture and the information in the following publications:

- *TMF Introduction* (a prerequisite to this manual)
- *TMF Planning and Configuration Guide* (a corequisite)
- *TMF Operations and Recovery Guide* (a corequisite)

## How This Manual Is Organized

[Section 1, Introduction to TMF](#), introduces the TMF product. It provides a brief overview of the objects controlled by TMF, and introduces TMFCOM, the interactive command interface to TMF.

[Section 2, Using TMFCOM](#), explains how to use TMFCOM, including how to access this interface, enter TMFCOM commands, direct output to destinations other than your terminal, use partially qualified file names in commands, and request online help.

[Section 3, TMFCOM Commands](#), describes the individual TMFCOM commands, presenting their syntax, security restrictions, usage guidelines, and examples of their use.

[Section 4, TMFCOM Tape Media Considerations](#), presents information about magnetic tape media used with TMF, including tape density settings, labeled-tape processing software interfaces, and guidelines for using labeled tape.

[Appendix A, TMFCOM Messages](#), describes the messages sent by TMFCOM to your workstation or terminal screen, or to an alternate output device.

[Appendix B, TMFCOM Command Reference Summary](#), shows the TMFCOM command syntax, extracted from the command descriptions in Section 3 and presented in summarized form.

[Appendix C, TMF Configuration Limits and Defaults](#), lists the operational limits on parameters used by TMF.

## TMF Documentation

Readers of this manual may also refer to the following documentation describing TMF:

| <b>Audience</b>               | <b>Title</b>                                |
|-------------------------------|---|
| All readers                   | <i>TMF Introduction</i>                     |
|                               | <i>TMF Glossary</i>                         |
| System managers and operators | <i>TMF Planning and Configuration Guide</i> |
|                               | <i>TMF Operations and Recovery Guide</i>    |
|                               | <i>TMF Reference Manual</i> (This manual)   |
| Application programmers       | <i>TMF Application Programmer's Guide</i>   |
|                               | <i>TMF Management Programming Manual</i>    |

## Documentation Description

- *TMF Introduction*  
Read this manual first. It provides a general overview of TMF concepts and capabilities for business professionals, application designers and programmers, and system managers and administrators.
- *TMF Glossary*  
Refer to this manual to look up technical terms used in the TMF documentation set.
- *TMF Planning and Configuration Guide*  
Read this guide for information about how to plan, configure, and manage a TMF environment. This guide also describes how to move your applications between an earlier TMF product version and the current one.
- *TMF Operations and Recovery Guide*  
Read this guide for information about how to perform and monitor standard TMF operations, obtain online and audit dumps, and respond to a variety of TMF exception conditions.
- *TMF Reference Manual*  
Refer to this manual for information about how to use the TMFCOM command interface to the TMF software. This manual includes syntax, cautionary considerations, and command examples for TMFCOM.

- *TMF Application Programmer's Guide*  
Read this guide for information about how to design requester and server modules to run in the TMF programming environment. Also read it for information about a set of procedures that can help in examining the content of TMF audit trails.
- *TMF Management Programming Manual*  
Read this manual for information about how to use the Subsystem Programmatic Interface (SPI) to communicate programmatically with the TMF software. Also read it for information about how to programmatically interpret Event Management Service (EMS) event messages generated by TMF.

## Other Documentation

In addition to the TMFCOM command and SPI programmatic interfaces provided with the TMF, several other software products and interfaces support limited TMF operations. Details about the TMF functionality they provide appear in the manuals that cover these products and interfaces:

- NonStop™ Operating System Procedure Calls
- NonStop Operating System TACL Command Interface
- NonStop SQL Relational Database Management System
- NonStop Transaction Services (TS/MP)
- NonStop TUXEDO
- Distributed Systems Management/Software Configuration Manager (DSM/SCM)
- Event Management Service (EMS)
- File Utility Program (FUP)
- Measure Subsystem
- Pathway/TS
- Surveyor Subsystem
- Various programming languages, including COBOL85, Pathway SCREEN COBOL, FORTRAN, TAL, Pascal, C, C++, and SQL (NonStop SQL implementation).

# Notation Conventions

## Hypertext Links

Blue underline is used to indicate a hypertext link within text. By clicking a passage of text with a blue underline, you are taken to the location described. For example:

This requirement is described under [Backup DAM Volumes and Physical Disk Drives](#) on page 3-2.

## General Syntax Notation

The following list summarizes the notation conventions for syntax presentation in this manual.

**UPPERCASE LETTERS.** Uppercase letters indicate keywords and reserved words; enter these items exactly as shown. Items not enclosed in brackets are required. For example:

MAXATTACH

**lowercase italic letters.** Lowercase italic letters indicate variable items that you supply. Items not enclosed in brackets are required. For example:

*file-name*

**[ ] Brackets.** Brackets enclose optional syntax items. For example:

TERM [ *\system-name.* ] \$*terminal-name*

INT[ ERRUPTS ]

A group of items enclosed in brackets is a list from which you can choose one item or none. The items in the list may be arranged either vertically, with aligned brackets on each side of the list, or horizontally, enclosed in a pair of brackets and separated by vertical lines. For example:

```
FC [  num  ]
   [  -num ]
   [  text ]
```

K [ X | D ] *address-1*

**{ } Braces.** A group of items enclosed in braces is a list from which you are required to choose one item. The items in the list may be arranged either vertically, with aligned braces on each side of the list, or horizontally, enclosed in a pair of braces and separated by vertical lines. For example:

```
LISTOPENS PROCESS {  $appl-mgr-name  }
                  {  $process-name   }
```

ALLWSU { ON | OFF }

**| Vertical Line.** A vertical line separates alternatives in a horizontal list that is enclosed in brackets or braces. For example:

```
INSPECT { OFF | ON | SAVEABEND }
```

**... Ellipsis.** An ellipsis immediately following a pair of brackets or braces indicates that you can repeat the enclosed sequence of syntax items any number of times. For example:

```
M address-1 [ , new-value ]...  
[ - ] { 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 }...
```

An ellipsis immediately following a single syntax item indicates that you can repeat that syntax item any number of times. For example:

```
"s-char..."
```

**Punctuation.** Parentheses, commas, semicolons, and other symbols not previously described must be entered as shown. For example:

```
error := NEXTFILENAME ( file-name ) ;  
LISTOPENS SU $process-name.#su-name
```

Quotation marks around a symbol such as a bracket or brace indicate the symbol is a required character that you must enter as shown. For example:

```
"[ repetition-constant-list ]"
```

**Item Spacing.** Spaces shown between items are required unless one of the items is a punctuation symbol such as a parenthesis or a comma. For example:

```
CALL STEPMOM ( process-id ) ;
```

If there is no space between two items, spaces are not permitted. In the following example, there are no spaces permitted between the period and any other items:

```
$process-name.#su-name
```

**Line Spacing.** If the syntax of a command is too long to fit on a single line, each continuation line is indented three spaces and is separated from the preceding line by a blank line. This spacing distinguishes items in a continuation line from items in a vertical list of selections. For example:

```
ALTER [ / OUT file-spec / ] CONTROLLER  
  
[ , attribute-spec ]...
```

**Examples.** In the examples of commands, command entries and other user input appears in bold-face type. For instance:

```
TMF 5>STATUS BEGINTRANS
```

Command output appears in regular type:

```
BeginTrans Status: Enabled
```

## Notation for Messages

The following list summarizes the notation conventions for the presentation of displayed messages in this manual.

**Nonitalic text.** Nonitalic letters, numbers, and punctuation indicate text that is displayed or returned exactly as shown. For example:

Backup Up.

**lowercase italic letters.** Lowercase italic letters indicate variable items whose values are displayed or returned. For example:

*p-register*

*process-name*

**[ ] Brackets.** Brackets enclose items that are sometimes, but not always, displayed. For example:

Event number = *number* [ Subject = *first-subject-value* ]

A group of items enclosed in brackets is a list of all possible items that can be displayed, of which one or none might actually be displayed. The items in the list might be arranged either vertically, with aligned brackets on each side of the list, or horizontally, enclosed in a pair of brackets and separated by vertical lines. For example:

*proc-name* trapped [ in SQL | in SQL file system ]

**{ } Braces.** A group of items enclosed in braces is a list of all possible items that can be displayed, of which one is actually displayed. The items in the list might be arranged either vertically, with aligned braces on each side of the list, or horizontally, enclosed in a pair of braces and separated by vertical lines. For example:

*obj-type obj-name* state changed to *state*, caused by  
{ Object | Operator | Service }

*process-name* State changed from *old-objstate* to *objstate*  
{ Operator Request. }  
{ Unknown. }

**| Vertical Line.** A vertical line separates alternatives in a horizontal list that is enclosed in brackets or braces. For example:

Transfer status: { OK | Failed }

**% Percent Sign.** A percent sign precedes a number that is not in decimal notation. The % notation precedes an octal number. The %B notation precedes a binary number. The %H notation precedes a hexadecimal number. For example:

%005400

P=%*p-register* E=%*e-register*



## Change Bar Notation

Change bars are used to indicate substantive differences between this edition of the manual and the preceding edition. Change bars are vertical rules placed in the right margin of changed portions of text, figures, tables, examples, and so on. Change bars highlight new or revised information. For example:

The message types specified in the REPORT clause are different in the COBOL85 environment and the Common Run-Time Environment (CRE).

The CRE has many new message types and some new message type codes for old message types. In the CRE, the message type SYSTEM includes all messages except LOGICAL-CLOSE and LOGICAL-OPEN.



# 1

# Introduction to TMF

TMF protects databases in a wide variety of environments, servicing online transaction processing (OLTP), online query processing (OLQP), batch processing, and decision-support systems.

This section introduces the TMF product, summarizes the objects managed by TMF, and introduces the TMFCOM command interpreter used to operate on those objects:

| Topic                                      | Page                |
|--|---------------------|
| <a href="#">About TMF</a>                  | <a href="#">1-1</a> |
| <a href="#">TMF Objects</a>                | <a href="#">1-2</a> |
| <a href="#">TMFCOM Command Interpreter</a> | <a href="#">1-4</a> |

## About TMF

TMF protects databases by managing database transactions, monitoring database activity through the use of audit trails, and providing database recovery methods. This product performs its functions while sustaining high performance in a high-volume, distributed application environment.

A general overview of TMF appears in *TMF Introduction*. That manual includes conceptual information about TMF and summarizes its features, components, and functions. It introduces and explains the basic terms that appear in this and other TMF manuals. You should read that manual before using this one. For full definitions of all TMF terms, see *TMF Glossary*.

Information about the commands used to access TMF appears in Sections 1 through 3 of this manual.

# TMF Objects

You manage and operate TMF, and control your TMF environment, by issuing commands that create, alter, delete, and return information about various TMF objects. These objects, listed in [Table 1-1](#), include such items as dumps, files, processes, and TMF itself.

**Table 1-1. TMF Objects** (page 1 of 2)

| Object               | Object Keyword | Description   |
|----------------------|----------------|---|
| Audit dump           | AUDITDUMP      | A copy of an audit-trail file written to a tape or disk volume by an audit dump process; audit dumps are used to recover audited files that have become unusable.   |
| Audit trail          | AUDITTRAIL     | A collection of records that contain information about database transactions and TMF activity: specifically, a series of audit-trail files containing audit records for transactions that make changes to audited disk files. Audit-trail configurations include active-audit volumes, overflow-audit volumes, restore-audit volumes, and an overflow threshold that triggers overflow copying. |
| Catalog              | CATALOG        | A database that contains information about audit dumps, online dumps, tape volumes, and disk volumes. This database specifies which media dumped files reside upon and which dump media are available for reuse.  |
| Data volume          | DATAVOL[S]     | A disk volume configured to contain audited database files and thus used to generate audit records to be placed in an active-audit trail.   |
| Database file        | FILES          | A file in an application database, potentially selectable for audit by TMF.   |
| Disk media for dumps | DISKDUMPS      | The disk media used for dumps.  |
| Dump                 | DUMPS          | A copy of an audited disk file (online dump) or a copy of an audit-trail file (audit dump).   |

**Table 1-1. TMF Objects** (page 2 of 2)

| <b>Object</b>          | <b>Object Keyword</b> | <b>Description</b>   |
|------------------------|-----------------------|--|
| Resource Manager       | RESOURCEMANAGER       | A representation of anything that contains data, such as a database or a queue. TMF uses a resource manager facility that is a collection of routines that enables cooperation between different transaction management systems running on different platforms, used to support heterogeneous transaction processing.      |
| Tape media             | TAPEMEDIA             | A tape volume known to TMF.  |
| TMF                    | TMF                   | TMF itself; a software system used to protect databases in online transaction processing (OLTP) and other environments.  |
| TMF operations         | OPERATION[S]          | TMF operations such as starting TMF or performing a file recovery or an online dump.   |
| TMF process type       | PROCESS               | Processes of a specific kind (for instance, backout processes) running as part of TMF.   |
| TMFSERVE process       | SERVER                | The TMF process that receives and acts upon commands entered through the TMFCOM interactive command interpreter. This process also allows programmatic access through the Subsystem Programmatic Interface (SPI), making it possible to construct system management programs that monitor and control the TMF environment. |
| Transaction            | TRANSACTION[S]        | A series of changes that transform a database from one consistent state to another. In application programs, transactions are bounded by a BEGINTRANSACTION statement and an ENDTRANSACTION statement.   |
| Transaction processing | BEGINTRANS            | The system library procedure that initiates transactions to a database, under the control of TMF.  |

# TMFCOM Command Interpreter

As a system manager or operator, you operate on TMF objects by issuing commands through the TMFCOM command interpreter. Within TMF, the commands are received by the TMFSERVE server process, which either operates on the objects itself or passes necessary requests to other peer processes through a programmatic messaging system.

TMFCOM and TMFSERVE interact in a common client-server relationship. Through TMFCOM, you enter the commands and receive responses to them at your workstation or terminal. You submit the commands in the standard “command+object” form. You access TMFCOM as directed in [Section 2, Using TMFCOM](#).

---

**Note.** TMFCOM communicates with TMFSERVE through a buffer provided by the Subsystem Programmatic Interface (SPI). For more information about this buffer, its size, and its effect on command processing, see [Entering TMFCOM Commands](#) on page 2-6.

---

TMFCOM provides commands for TMF operations generally required by system managers and operators, and all are described in detail in [Section 3, TMFCOM Commands](#). For convenience, the commands and their syntax descriptions are also summarized in [Appendix B, TMFCOM Command Reference Summary](#).

Programmers can issue the same commands from within an application process. Programmatic access to TMFSERVE is furnished through SPI, which permits construction of system management programs that monitor and control the TMF environment. For information about using SPI, see the *TMF Management Programming Manual*.

Additional TMF operations are available through the SNOOP utility. For safe and effective use, however, several SNOOP commands require a high level of TMF background and, perhaps, the assistance of the Global Customer Support Center (GCSC) or your service provider. The SNOOP utility is provided with the TMF product, and is documented in the file named \$SYSTEM.SYS<sub>nn</sub>.SNOOPDOC.

# 2 Using TMFCOM

TMFCOM allows you to enter commands that initiate communication with TMF, request various TMF operations, and terminate communication with TMF. This section explains how to use TMFCOM, discussing the following topics:

| Topic  | Page                 |
|--|----------------------|
| <a href="#">Running TMFCOM</a>                               | <a href="#">2-1</a>  |
| <a href="#">Entering TMFCOM Commands</a>                     | <a href="#">2-6</a>  |
| <a href="#">Directing Output to an Alternate Destination</a> | <a href="#">2-8</a>  |
| <a href="#">Using Partially Qualified File Names</a>         | <a href="#">2-9</a>  |
| <a href="#">Requesting Online Help</a>                       | <a href="#">2-10</a> |

△ **Caution.** Some operational limits and the default operational behavior described in this section may have changed since this manual was published with RVU H06.03. If such changes were made, they will be described in *TMF Supplement for Limits and Defaults Changed after RVU H06.03*. If that Supplement currently exists, please read it for new information about limits and defaults.

## Running TMFCOM

TMFCOM runs under the Guardian user interface (normally the TACL command interpreter) to the NonStop operating system. To initiate communication with TMFCOM, you enter the TMFCOM command keyword at the current TACL prompt. This command starts a TMFCOM session that lets you enter TMFCOM commands interactively, noninteractively, or through a command file, as explained in the next few pages. The TMFCOM object program file, along with all other TMF object program files, resides on the \$SYSTEM.SYS<sub>nn</sub> subvolume.

**Note.** The previous version of TMFCOM is released as TMFCOM1, and is available for your convenience in the file \$SYSTEM.SYS<sub>nn</sub>.TMFCOM1.

## Command Syntax for Starting a Session

To start a TMFCOM session, use the following general command syntax. The specific options or parameters you enter depend, of course, on the choices you desire.

```
TMFCOM [/[IN command-file][,OUT output-file]/]  
        [command [; command]]...
```

IN *command-file*

specifies a command file from which TMFCOM commands are to be read. If you do not specify a command file, TACL automatically supplies the name of its own

input source—usually the workstation or terminal from which you issue the TMFCOM command.

*OUT output-file*

specifies a file to which TMFCOM output (other than prompts for entering TMFCOM commands) is to be written. This file, for example, might receive listings requested by INFO, SHOW, or STATUS commands. It might also receive TMFCOM commands generated by the OBEYFORM option as described in [Section 3, TMFCOM Commands](#).

---

**Note.** For a file specified in the OUT parameter, TMFCOM does not generate output from the following commands: EXIT, FC, OBEY, ?, and !.

---

If you do not specify an output file, TACL automatically supplies the name of its own output destination—usually the workstation or terminal from which you issue the TMFCOM command.

If you specify a file that does not exist in the OUT parameter, TMFCOM creates a new output file in edit (file code 101) format.

If you specify an existing file in the OUT parameter, TMFCOM uses that file as the output destination regardless of the file's format. TMFCOM purges any existing data in the file before it writes the new output.

*command*

is a command that requests a TMFCOM operation. Multiple commands are permitted.

- If commands are present and an input file is also specified in the IN option, TMFCOM executes them and terminates without reading the input file.
- If commands are not present but an input file is specified, TMFCOM executes the commands in the input file.
- If commands are not present and no input file is specified, TMFCOM prompts you to enter them.

Examples showing application of this syntax appear in the remainder of this section.

## Using TMFCOM Interactively

When you use TMFCOM interactively, you conduct a continuous online dialog with it through a series of prompts, commands, output displays, and messages.

### Starting a Session

To begin an interactive TMFCOM session, enter the TMFCOM command keyword at your TACL prompt:

```
1> TMFCOM
```



TMFCOM replies with version information and a prompt for input:

```
TMFCOM - T8652H01 - (01MAY2005 - TMF)
(C) 2005 Hewlett-Packard Development Company, L.P.
TMF 1>
```

“TMF 1>” is the TMFCOM input prompt, which tells you that TMFCOM is ready to accept your first command. When you enter this command, TMFCOM processes it and displays another input prompt for your next command. You continue interacting with TMFCOM in this way, repeatedly receiving a prompt and entering a command, until you explicitly end the session.

The input prompt includes a command line number that identifies the sequence in which the command is entered in your session. For each new command, TMFCOM increments the line number by 1. TMFCOM also lists the commands by line number in its session history buffer. At any time, you can display the contents of this buffer by using the HISTORY command, described in [Section 3, TMFCOM Commands](#).

If an unexpected event prematurely terminates TMFCOM at any time, you can restart TMFCOM by re-entering the TMFCOM command keyword at the TACL prompt.

## Ending a Session

When you want to end your session with TMFCOM, you can either issue the EXIT command or type Ctrl/Y.

---

**Note.** On most workstation and terminal keyboards, you enter Ctrl/Y by simultaneously pressing the control key and the Y key. On some keyboards, however, this escape function might be selected by a different key combination or a single key. Before using TMFCOM, determine how this function is selected on your keyboard.

---

## Interrupting Command Processing

You can interrupt TMFCOM processing by pressing the BREAK key at your workstation or terminal. TMFCOM responds as follows:

- If you press BREAK at the TMFCOM input prompt (TMF *line-number*>), TMFCOM returns control of the workstation or terminal to TMFCOM's parent process (typically, TACL) but continues execution. You can resume communication with TMFCOM by entering the operating system command PAUSE at the TACL prompt.
- If you press BREAK when a TMFCOM command that displays information (such as INFO DUMPS) is in progress, TMFCOM terminates execution of this command and prompts you for another one.
- If you press BREAK when a TMFCOM command that performs an active operation (such as ALTER TAPEMEDIA) is in progress, TMFCOM continues to execute this command while immediately prompting you for another one.

- If you press BREAK while TMFCOM is reading a continued line of a partially entered multiline command, TMFCOM does not interpret this command and prompts you instead for a new command.

---

**Note.** On your workstation or terminal keyboard, the BREAK key might not actually be labeled “BREAK.” All keyboards, however, have some key or combination of keys that perform the BREAK operation. Before using TMFCOM, determine how this operation is selected on your keyboard.

---

## Using TMFCOM Noninteractively

When you use TMFCOM noninteractively, you enter one or more commands to TMFCOM at the same time you start your session. TMFCOM executes all of those commands, and then terminates the TMFCOM session and returns control of your workstation or terminal to TACL.

To run TMFCOM noninteractively, you enter the TMFCOM keyword followed by one or more TMFCOM commands on the same line:

```
1> TMFCOM STATUS TMF
TMFCOM - T8652H01 - (01MAY2005 - TMF)
(C)2005 Hewlett-Packard Development Company, L.P.
TMF Status:
  System: \FOXII, Time: 05-May-2005 12:44:26
  State: started
  Transaction Rate: 0.08 TPS
AuditTrail Status:
  Master:
    Active audit trail capacity used: 17%
    First pinned file: $MAT1.ZTMFAT.AA000341
    Reason: Current File
    Current file: $AUDIT.ZTMFAT.AA000341
AuditDump Status:
  Master: State: enabled, Status: inactive
BeginTrans Status:  ENABLED
Catalog Status:
  Status: up
  .
  .
  .
2>
```

As this example illustrates, after the command is executed, the TMFCOM session automatically terminates and the TACL prompt again appears.

To specify multiple TMFCOM commands on a single line, you enter a semicolon (;) to separate each command from the preceding one. For example:

```
3> TMFCOM STATUS SERVER; STATUS TRANSACTION; STATUS DATAVOLS
  .
  .
  .
```

If any command on the line fails, TMFCOM reports the error and terminates without executing any of the subsequent commands.

## Using TMFCOM From a Command File

TMFCOM can also read commands from an input file known as a command file. A command file is simply a text file that contains the TMFCOM commands you want to execute, which you prepare using your standard text editor. You might, for example, create a command file named TMFINIT that contains the following commands:

```
SET AUDITTRAIL ACTIVEVOLS ($MAT1, $MAT2)
SET AUDITTRAIL OVERFLOWVOLS $MAT2
SET AUDITTRAIL RESTOREVOLS $MAT2
ADD AUDITTRAIL MASTER, AUDITDUMP ON, FILESPERVOLUME 3, &
  BEGINTRANSDISABLE 90, OVERFLOWTHRESHOLD 75
ALTER AUDITDUMP MASTER, COPIES 2 PARALLEL
ALTER CATALOG RETAINDEPTH 4
```

To execute the commands in this command file, enter the TMFCOM command keyword and specify the command file name in the IN parameter:

```
4> TMFCOM /IN TMFINIT/
```

TMFCOM begins the session, reading and executing each command from the command file. When TMFCOM encounters an EXIT command or reaches the end of the command file, TMFCOM terminates the session and returns control to TACL. If TMFCOM encounters an error while reading the command file, TMFCOM displays an error message and terminates the session.

If you are using TMFCOM interactively, you can use the OBEY command to execute a command file within your current session. If you regularly use a series of sequential operations in your interactive sessions, for example, you might want to specify these in a command file. Then, each time you need these operations, you can invoke them with a single OBEY command rather than with multiple individual commands. If you have created a command file named REPORT that contains several STATUS commands, for instance, you could execute all these commands by simply entering:

```
TMF 7> OBEY REPORT
```

In response, TMFCOM executes all the commands entered in the file named REPORT. If you decide later that you want different reports, you can change the command file and then enter the OBEY command again. Using command files makes performing repeated tasks very convenient.

---

△ **Caution.** Never include exclamation points (!) in command files. When TMFCOM encounters certain commands, it normally sends you a message asking you to confirm your request; an exclamation point in the command file, however, prevents transmittal of this message. Such messages and your responses to them can be a great help in avoiding errors.

---

A command file can, in turn, contain OBEY commands. OBEY commands can be nested up to four levels.

# Entering TMFCOM Commands

As you enter TMFCOM commands, you can:

- Include comments in command lines.
- Continue a single command over multiple lines.
- Enter multiple commands on a single line.

---

△ **Caution.** The previous version of TMFCOM, now named TMFCOM1, allowed some syntax errors, usually spaces in file names. These syntax errors are detected by the latest version of TMFCOM covered by this manual and may cause errors in OBEY FILES or TACL scripts. As an example, the following file name syntax, including the spaces, was valid using the previous version of TMFCOM but results in an error using the current version:

*\node .\$volume . subvolume. filename.*

---

**Note.** TMFCOM communicates with TMFSERVE through a buffer provided by the Subsystem Programmatic Interface (SPI). This buffer will contain all the SPI tokens and data needed for any particular command and is limited to 28672 characters (28 KB). If you enter a TMFCOM command that requires more than 28672 characters of buffer space, TMFCOM displays Error 1050, alerting you to the excessive command length.

No line of command text can exceed 2048 characters, including comments and spaces, before encountering a carriage return or a continuation character (&). For information about entering command lines longer than 2048 characters, see [Continuing Commands on Multiple Lines](#). The total length of all command text entered cannot exceed 28672 characters before encountering a carriage return, as also discussed under [Continuing Commands on Multiple Lines](#).

---

## Including Comments in Command Lines

You can enter comments within a TMFCOM command line. The comment can precede or follow the command, or can even be embedded within it. You begin the comment with a pair of hyphens (- -) and terminate it with another pair of hyphens:

```
TMF 9> ADD DATAVOLS -- this is a comment -- $DATA1
```

When TMFCOM processes the command line, it ignores all text between the two sets of hyphens.

Comments need not be terminated with a second set of hyphens if the rest of the line can be ignored and the line is the last or only line of the command.

In the next example, the comment in the first command does not need a second set of hyphens because the rest of the line can be ignored. The comment in the second command, however, requires matching pairs of hyphens because the text following the comment is part of the command and cannot be ignored:

```
TMF 11> ADD DATAVOLS $DATA2 -- a comment goes here
TMF 12> ADD -- another comment goes here -- DATAVOLS $DATA3
```

## Continuing Commands on Multiple Lines

You can continue a single TMFCOM command over two or more lines. For example, you must do this when a command contains too many parameters to fit into one 2048-character line (as sometimes occurs in OBEY command files). In other cases, lengthy comments on a command line force the text onto one or more additional lines. In still other cases, you want a command that spans multiple lines on your workstation or terminal screen, or in a command file, to break neatly.

To continue a command onto another line, enter an ampersand (&) as the last character in the line you want to continue. When you enter the command interactively, TMFCOM responds by displaying ">>>" on the next line to prompt you for the rest of the command. For example, in the first two lines of the following command, the ampersand indicates that you want to continue the command onto the next line:

```
TMF 16> ALTER AUDITDUMP AUX01, MEDIUM DISK, ADDDISKMEDIA &
>>>(\REMOTE1.$DATA01.RD, \REMOTE1.$DATA02.RD,      &
>>>\REMOTE1.$DATA05.RD)
TMF 17>
```

By using the ampersand as a continuation character, you can enter multiline commands that contain up to 28672 characters, excluding ampersands and trailing blanks.

Continuation takes precedence over comments. Thus, you should not include an ampersand within a comment in a single-line command, because the ampersand would not be considered part of the comment and would needlessly force continuation of the entry onto the next line.

The following command entry contains an ampersand and a comment, used in an appropriate way:

```
TMF 21> ADD -- comment -- &
>>>AUDITTRAIL MASTER, AUDITDUMP ON
```

This entry is equivalent to:

```
TMF 21> ADD AUDITTRAIL MASTER, AUDITDUMP ON
```

## Entering Multiple Commands on a Line

As shown earlier, TMFCOM also lets you use semicolons to indicate multiple commands on a single entry line:

```
TMF 22> STATUS TMF; STATUS TRANSACTION
```

This entry is equivalent to:

```
TMF 22> STATUS TMF
TMF 23> STATUS TRANSACTION
```

TMFCOM executes the commands in the order they are entered.

If you include a command that requires an answer to a confirming question (such as ABORT TRANSACTION) before TMFCOM processes that command further, and you respond with NO or STOP, the current command is discontinued, and the next command is executed.

After all the commands are executed, TMFCOM takes one of the following actions:

- If TMFCOM is running interactively, it prompts you for another command.
- If TMFCOM is running noninteractively, it automatically terminates the TMFCOM session and returns control to TACL.

## Directing Output to an Alternate Destination

In most cases, TMFCOM users receive command output at their workstations or terminals. However, you can direct output from most commands that generate it to other destinations as well. In particular, you can name a disk file, a process (such as the spooler), or a device (such as a printer, tape drive, or other terminal or workstation) as the destination.

To direct output to an alternate destination, you name that destination in the OUT option, entered between the command keyword and object keyword of the command.

---

**Note.** You cannot use the OUT option, however, for the following commands: DELETE TMF, EXIT, FC, OBEY, ?, and !.

---

The syntax for the OUT option is:

```
[ /OUT [\node.] $process [.#identifier]/      ]
[ /OUT [\node.][$volume.][subvolume.] file-id [!]/ ]
```

OUT [\node.] \$process [.#identifier]

specifies the name of a process or a device as the destination of the output generated by the TMFCOM command.

OUT [\node.][\$volume.][subvolume.] file-id [!]

specifies the name of a file as the destination of the output generated by the TMFCOM command. If you specify a disk file that does not exist, an edit file having the name you specified is automatically created. This edit file has file code 101, and output generated by the TMFCOM command is directed to it.

- If you specify a disk file that exists, output generated by the TMFCOM command is appended to that file regardless of the file's format.
- If you specify a disk file that exists and terminate the OUT option with the exclamation point (!), any data presently stored in the file is purged and the EOF indicator is set to 0 before any new data is written to the file.

- If you omit *\node*, *volume*, or *subvolume*, the default is your current node, volume, or subvolume, respectively.

If you omit the OUT option, the device from which you enter the TMFCOM command receives the output.

As an illustration of the OUT option, to direct the output from a STATUS TMF command to a disk file named \$OPS.STATUS.TMF, you enter the following command:

```
TMF 25> STATUS /OUT $OPS.STATUS.TMF/ TMF
```

## Using Partially Qualified File Names

For convenience and brevity, you can use partially qualified file names in TMFCOM commands. These names, for example, often specify the file but not the node (system), volume, or subvolume on which the file resides. The system then determines, by context, the node, disk volume, and subvolume where the file is stored. Specifically, partially qualified file names are expanded according to the default node, volume, and subvolume names in effect when TACL is started, or by default names established by TMFCOM commands as follows:

- For file names of command files, the defaults established by the TMFCOM commands OBEYVOL and VOLUME are used.
- For file names of output files, the defaults established by the TMFCOM command VOLUME are used.
- For file names of other files used in TMFCOM commands, the defaults established by the CMDVOL and VOLUME commands are used.

The default names established by TMFCOM override those established in TACL.

Suppose, for instance, that you use the following VOLUME command to set your TMFCOM command context to the default volume \$OPS and the default subvolume TMFLOC:

```
TMF 33> VOLUME $OPS.TMFLOC
```

Thereafter, when you enter a command to access any file in the current subvolume, you can omit the node, volume, and subvolume names from this command, and use only the file name:

```
TMF 34> DUMP FILES DFILE1
```

In this context, this command would be equivalent to entering:

```
TMF 35> DUMP FILES $OPS.TMFLOC.DFILE1
```

# Requesting Online Help

You can request online help to remember TMFCOM command names and syntax. This assistance is available through the HELP command, which can display:

- A list of the TMFCOM commands associated with a specific keyword, such as ADD, ALTER, INFO, or SET
- The syntax diagram for a specific TMFCOM command
- A list of all TMFCOM commands

---

**Note.** The HELP command output only shows commands and options available with the currently opened TMFSERVE process. If your TMFCOM is accessing an earlier version of TMFSERVE, some of the current HELP information may not appear.

---

To list all of the TMFCOM commands associated with a particular keyword, enter the HELP command followed by the keyword itself. For example, to display the commands associated with the ADD command, enter:

```
TMF 40> HELP ADD
```

TMFCOM responds by displaying:

```
-----  
|  ADD AUDITTRAIL  
|  ADD DATAVOLS  
|  ADD DUMPS  
|  ADD RESOURCEMANAGER  
|  ADD TAPEMEDIA  
|  
-----
```

As another example, to list all the TMFCOM commands associated with the keyword AUDITDUMP, enter:

```
TMF 41> HELP AUDITDUMP
```

TMFCOM displays:

```
-----  
|  ALTER AUDITDUMP  
|  DISABLE AUDITDUMP  
|  ENABLE AUDITDUMP  
|  INFO AUDITDUMP  
|  STATUS AUDITDUMP  
|  
-----
```



To obtain the syntax diagram for a specific TMFCOM command, enter the **HELP** command followed by the TMFCOM command that you want information about. To obtain the syntax for the **ADD AUDITTRAIL** command, for example, enter:

TMF 43> **HELP ADD AUDITTRAIL**

TMFCOM responds by displaying:

-----

ADD AUDITTRAIL Command

Adds one or more audit trails to the TMF configuration.

Security: super user group member.

-----

```

ADD AUDITTRAIL {MASTER | MAT           }
                {AUX[ILIARY]nn         }

[,AUDITDUMP {ON | OFF                   }]

[,FILESIZE integer                       ]

[,ACTIVEVOL[S] {volume                  }
                {(volume [,volume]...) }]

[,FILESPERVOLUME integer                 ]

[,OVERFLOWVOL[S] {volume                 }
                  {(volume [,volume]...) }]

[,OVERFLOWTHRESHOLD integer              ]

[,RESTOREVOL[S] {volume                  }
                 {(volume [,volume]...) }]

[,BEGINTRANSDISABLE integer              ]

```

-----

To list all TMFCOM commands, enter the HELP command followed by the keyword ALL:

TMF 44> **HELP ALL**

TMFCOM responds by listing all TMFCOM commands in alphabetical order.

```
-----  
| ABORT TRANSACTION  
| ADD AUDITTRAIL  
| ADD DATAVOLS  
| ADD DUMPS  
| ADD RESOURCEMANAGER  
| ADD TAPEMEDIA  
| ALTER AUDITDUMP  
| ALTER AUDITTRAIL  
| ALTER BEGINTRANS  
| .  
| .  
| .  
| STATUS TRANSACTIONS  
| STOP TMF  
| VOLUME  
| <?>  
| <!>  
|-----
```

# 3 TMFCOM Commands

The TMFCOM utility program and its associated commands enable you to operate and manage a TMF environment. The TMFCOM commands and their attributes are described in this section, which presents these topics:

| Topic  | Page                  |
|--|-----------------------|
| <a href="#">Command Description Elements</a>       | <a href="#">3-1</a>   |
| <a href="#">File Names and Process Identifiers</a> | <a href="#">3-6</a>   |
| <a href="#">Commands:</a>                          |                       |
| <a href="#">AB - AD</a>                            | <a href="#">3-10</a>  |
| <a href="#">AL - CA</a>                            | <a href="#">3-43</a>  |
| <a href="#">CL - DE</a>                            | <a href="#">3-98</a>  |
| <a href="#">DI - FC</a>                            | <a href="#">3-116</a> |
| <a href="#">HE - IN</a>                            | <a href="#">3-145</a> |
| <a href="#">NE - RE</a>                            | <a href="#">3-189</a> |
| <a href="#">SE - SH</a>                            | <a href="#">3-232</a> |
| <a href="#">ST - VO</a>                            | <a href="#">3-256</a> |
| <a href="#">!</a>                                  | <a href="#">3-311</a> |
| <a href="#">?</a>                                  | <a href="#">3-314</a> |

---

△ **Caution.** Some limits and default values for the TMFCOM command options and parameters, and the operational behavior described in this section may have changed since this manual was published with RVU H06.03. If such changes were made, they will be described in *TMF Supplement for Limits and Defaults Changed after RVU H06.03*. If that Supplement currently exists, please read it for new information about limits and defaults.

---

## Command Description Elements

In the description of each TMFCOM command, the following elements are covered:

- Purpose, syntax, and options available with the command
- Security restrictions
- TMF state requirement
- Usage guidelines (where applicable)
- Output displayed (where applicable)
- Examples

## Purpose, Syntax, and Parameters

Each description begins by explaining the purpose or function of the command. Then, a syntax diagram and complete parameter descriptions indicate what keywords and variables make up the command: which are required, which are optional, and what the default assignments for the optional items are. The meanings of the symbols used in the syntax diagrams are described under [Notation Conventions](#) on page xii.

## Security Restrictions

While several TMFCOM commands are available to all users, others have great impact on the entire TMF system and are restricted to particular users only. The specific security requirements for a command appear in the command description under the heading “Security Restrictions.” In general:

- The following TMFCOM commands are available to all users: CMDVOL, DISPLAY OPERATIONS, ENV, EXIT, FC, HELP, HISTORY, INFO, OBEY, OBEYVOL, OPEN SERVER, RESET, SET, SHOW, STATUS, VOLUME, ?, and !.
- The other TMFCOM commands are restricted for use only by members of the super user group.
- Persons with the super ID (user ID 255,255) can issue any TMFCOM command.

For your convenience, the security restrictions for all TMFCOM commands are also summarized in [Table 3-1](#). This table shows which commands all users can enter, which commands are restricted to members of the super user group, and which commands are limited to persons with the super ID.

## TMF State Requirement

Some TMFCOM commands can only be entered after TMF has been started; others must be entered before TMF has been started or after it has been stopped. In each command description, these constraints are listed under the heading “TMF State Requirement.” For your convenience, the state requirements are also summarized in Table 3-1.

**Table 3-1. User Security Restrictions and TMF State Requirements for TMFCOM Commands** (page 1 of 4)

| Command             | Security Restrictions   | TMF State Requirement                             |
|---------------------|-------------------------|---|
| ABORT TRANSACTION   | Super user group member | Started   |
| ADD AUDITTRAIL      | Super user group member | TMF configuration deleted and TMF not yet started |
| ADD DATAVOLS        | Super user group member | Started   |
| ADD DUMPS           | Super user group member | Started   |
| ADD RESOURCEMANAGER | Super user group member | Started   |

**Table 3-1. User Security Restrictions and TMF State Requirements for TMFCOM Commands** (page 2 of 4)

| <b>Command</b>            | <b>Security Restrictions</b> | <b>TMF State Requirement</b>   |
|---------------------------|------------------------------|--|
| ADD TAPEMEDIA             | Super user group member      | Started  |
| ALTER AUDITDUMP           | Super user group member      | Any state  |
| ALTER AUDITTRAIL          | Super user group member      | Once configured, TMF must be started for altering AUDITDUMP to OFF or specifying ADDACTIVEVOLS or DELETEACTIVEVOLS |
| ALTER BEGINTRANS          | Super user group member      | Started  |
| ALTER CATALOG             | Super user group member      | Any state  |
| ALTER DATAVOLS            | Super user group member      | Started  |
| ALTER DUMPS               | Super user group member      | Started  |
| ALTER PROCESS             | Super user group member      | Any state  |
| ALTER TAPEMEDIA           | Super user group member      | Started  |
| ALTER TMF                 | Super user group member      | Stopped for CONFIGVOL option; Any state for SWITCHPROCESS TMP option   |
| CANCEL OPERATION          | Super user group member      | Started  |
| CLOSE<br>RESOURCEMANAGER  | Super user group member      | Started  |
| CMDVOL                    | All users                    | Any state  |
| DELETE CATALOG            | Super ID                     | Started  |
| DELETE DATAVOLS           | Super user group member      | Started, and data volumes disabled before their deletion   |
| DELETE DUMPS              | Super user group member      | Started  |
| DELETE<br>RESOURCEMANAGER | Super user group member      | Started  |
| DELETE TAPEMEDIA          | Super user group member      | Started  |
| DELETE TMF                | Super ID                     | Stopped  |
| DELETE TRANSACTION        | Super ID                     | Started  |
| DISABLE AUDITDUMP         | Super user group member      | Any state, but audit trail associated with dump must be configured for dumping                                     |
| DISABLE BEGINTRANS        | Super user group member      | Started  |
| DISABLE DATAVOLS          | Super user group member      | Started  |
| DISPLAY OPERATIONS        | Read-access to EMS log files | Any state  |

**Table 3-1. User Security Restrictions and TMF State Requirements for TMFCOM Commands** (page 3 of 4)

| <b>Command</b>       | <b>Security Restrictions</b> | <b>TMF State Requirement</b>   |
|----------------------|------------------------------|--|
| DUMP FILES           | Super user group member      | Started  |
| ENABLE AUDITDUMP     | Super user group member      | Any state, but audit trail associated with dump must be configured for dumping |
| ENABLE BEGINTRANS    | Super user group member      | Started  |
| ENABLE DATAVOLS      | Super user group member      | Started  |
| ENV                  | All users                    | Any state  |
| EXIT                 | All users                    | Any state  |
| FC                   | All users                    | Any state  |
| HELP                 | All users                    | Any state  |
| HISTORY              | All users                    | Any state  |
| INFO AUDITDUMP       | All users                    | Any state  |
| INFO AUDITTRAIL      | All users                    | Any state  |
| INFO BEGINTRANS      | All users                    | Started  |
| INFO CATALOG         | All users                    | Any state  |
| INFO DATAVOLS        | All users                    | Started  |
| INFO DUMPS           | All users                    | Any state  |
| INFO PROCESS         | All users                    | Any state  |
| INFO RESOURCEMANAGER | All users                    | Started  |
| INFO TAPEMEDIA       | All users                    | Started  |
| INFO TMF             | All users                    | Any state  |
| NEXT AUDITTRAIL      | Super user group member      | Started  |
| OBEY                 | All users                    | Any state  |
| OBEYVOL              | All users                    | Any state  |
| OPEN SERVER          | All users                    | Any state  |
| RECOVER FILES        | Super user group member      | Started  |
| RELOCATE DISKDUMPS   | Super user group member      | Started  |
| RESET AUDITTRAIL     | All users                    | Any state  |
| RESET DATAVOLS       | All users                    | Any state  |
| RESET DUMPS          | All users                    | Any state  |
| RESET TAPEMEDIA      | All users                    | Any state  |
| RESOLVE TRANSACTION  | Super user group member      | Started  |
| SET AUDITTRAIL       | All users                    | Any state  |

**Table 3-1. User Security Restrictions and TMF State Requirements for TMFCOM Commands** (page 4 of 4)

| <b>Command</b>         | <b>Security Restrictions</b> | <b>TMF State Requirement</b>   |
|------------------------|------------------------------|--|
| SET DATAVOLS           | All users                    | Any state  |
| SET DUMPS              | All users                    | Any state  |
| SET TAPEMEDIA          | All users                    | Any state  |
| SHOW AUDITTRAIL        | All users                    | Any state  |
| SHOW DATAVOLS          | All users                    | Any state  |
| SHOW DUMPS             | All users                    | Any state  |
| SHOW TAPEMEDIA         | All users                    | Any state  |
| START TMF              | Super user group member      | Stopped, and audit trails configured   |
| STATUS AUDITDUMP       | All users                    | Started  |
| STATUS AUDITTRAIL      | All users                    | Started  |
| STATUS BEGINTRANS      | All users                    | Started  |
| STATUS CATALOG         | All users                    | Any state  |
| STATUS DATAVOLS        | All users                    | Started, or shutting down  |
| STATUS OPERATIONS      | All users                    | Any state  |
| STATUS RESOURCEMANAGER | All users                    | Any state, but displays only open volatile resource managers if TMF is stopped |
| STATUS SERVER          | All users                    | Any state  |
| STATUS TMF             | All users                    | Any state  |
| STATUS TRANSACTIONS    | All users                    | Started  |
| STOP TMF               | Super user group member      | Started (when ABRUPT option not used)<br>Any state (when ABRUPT option used)   |
| VOLUME                 | All users                    | Any state  |
| ?                      | All users                    | Any state  |
| !                      | All users                    | Any state  |

## Usage Guidelines

Details about the proper use of a command appear under “Usage Guidelines.” These details include information about when to apply the command, possible constraints and side-effects of the command, hints on enhancing performance and avoiding errors, and other considerations.

When a command is extremely basic or straightforward, the command description omits the “Usage Guidelines.” This is the case, for instance, with many commands that simply display information at your terminal or workstation.

## Output Displayed

For the TMFCOM commands that display output at your terminal (for example, the INFO, SHOW, and STATUS series of commands), the specific elements that appear are explained under “Output Displayed.”

## Examples

Under the heading “Examples,” each command description presents one or more examples illustrating the use of the command. For commands that display information, sample displays also appear. Command entries and other user input appear in bold-face type; TMFCOM output appears in regular type.

The sample displays shown in this manual, although produced on earlier G-series servers, will appear the same (except for some dates displayed) on H-series servers.

## File Names and Process Identifiers

File names and process identifiers appear often as parameters in TMFCOM commands.

These names typically identify objects such as disk files, tape volumes, and processes. Errors can result from improperly specifying these names in commands. Because the same syntax applies to them in almost every case, the syntax rules for the most frequently encountered names appear together, preceding the individual command descriptions, rather than spread throughout. Where exceptions to these rules occur, they are noted in the command descriptions.

The system does not distinguish between uppercase and lowercase alphabetic characters in a file name. If all the optional left-hand parts of a file name are present, it is called a fully qualified file name; if any of the optional left-hand parts are missing, it is called a partially qualified file name.

For more information about file names and process identifiers and the rules that govern them, see the *Guardian Procedure Calls Reference Manual*.



## Disk File Names

The syntax for a file name that identifies a disk file is:

```
[node.][[volume.]subvol.]file-id
```

or

```
[node.][volume.]temp-file-id
```

*node*

specifies the name of the node (system) on which the file resides. A node name consists of a backslash (\) followed by one to seven alphanumeric characters; the first alphanumeric character must be a letter.

*volume*

specifies the name of the volume on which the file resides. A volume name consists of a dollar sign (\$) followed by one to seven alphanumeric characters; the first alphanumeric character must be a letter.

*subvol*

specifies the name of the subvolume on which the file resides. A subvolume name has one to eight alphanumeric characters; the first character must be a letter.

*file-id*

specifies the file identifier (or name) of a permanent disk file. A permanent-file identifier has one to eight alphanumeric characters; the first character must be a letter.

*temp-file-id*

specifies the file identifier (or name) of a temporary disk file. A temporary-file identifier consists of a pound-sign (#) followed by four to seven numeric characters. The operating system assigns file identifiers to temporary files.

The following is an example of a fully qualified disk file name:

```
\hdq.$mkt.reports.finance
```

Some TMFCOM commands allow the asterisk (\*) as a wild-card character in place of the *volume*, *subvol*, and *file-id* names in the *volume.subvol.file-id* construct used for identifying permanent disk files. The asterisk indicates “all.” Thus, the following example specifies all files in the subvolume Y on the disk volume \$DATA1:

```
$DATA1.Y.*
```

The next example specifies all files in all subvolumes on the volume \$DATA5:

```
$DATA5.*.*
```

The last example specifies all files in all subvolumes on all volumes (in other words, all files stored at the node):

```
*.*.*
```

Notice that, as a general rule, the asterisk can only replace a complete volume name, subvolume name, or file-id. Thus, you cannot use an asterisk to replace just a portion of an element name; for example, you cannot specify a volume named \$DATA\* or a file identified as BAS\*12.

---

**Note.** Exceptions to this rule do exist: for a few TMFCOM commands, the asterisk (\*) and the question mark (?) can be used to replace only portions of volume, subvolume, and file names. Directions for this precise pattern matching appear in the discussions of the particular commands.

---

## Nondisk Device Names

The syntax for a file name that identifies a nondisk device is:

```
[node.]device-name[.qualifier]
```

*node*

specifies the name of the node on which the device resides. A node name consists of a backslash (\) followed by one to seven alphanumeric characters; the first alphanumeric character must be a letter.

*device-name*

specifies the name of a device. A device name consists of a dollar sign (\$) followed by one to seven alphanumeric characters; the first alphanumeric character must be a letter.

*qualifier*

is an optional qualifier. It consists of a pound sign (#) followed by one to seven alphanumeric characters; the first alphanumeric character must be a letter.

## Process Identifiers

TMFCOM commands can refer to (and display information about) both unnamed and named processes.

### Identifiers for Unnamed Processes

The syntax used to identify an unnamed process is:

`(cpu,pin)`

*cpu*

specifies the CPU number of the processor in which the process is running. *cpu* is one or two digits representing a value in the range 0 through 15. A leading zero must be suppressed.

*pin*

specifies the process identification number of the process. *pin* is one to five digits representing a value in the range 0 through the maximum value allowed for the CPU. Leading zeros must be suppressed.

The following are examples of identifiers for unnamed processes:

`(2,850)`  
`(6,2101)`

### Identifiers for Named Processes

The syntax used to identify a named process is:

`[node.]process-name`

*node*

specifies the name of the node on which the process is running. A node name consists of a backslash (\) followed by one to seven alphanumeric characters; the first alphanumeric character must be a letter.

*process-name*

specifies the name of the process. A process name consists of a dollar sign (\$) followed by one to five alphanumeric characters; the first alphanumeric character must be a letter.

The following are examples of identifiers for named processes:

`$beta`  
`$redd1`  
`\NY.$dm04`

# ABORT TRANSACTION

The ABORT TRANSACTION command backs out (cancels) the effects of an active transaction, or reattempts to back out an aborting or hung transaction, and releases the locks held by that transaction.

```
ABORT TRANS[ACTION[S]]

    {transaction-id
    {(transaction-id [,transaction-id]...)}

    [,IGNOREDATAERRORS | ,AVOIDHANGING]

where transaction-id is:

    [node .
    [node-number .
    [node(tm-flags) .
    [node-number(tm-flags) .
    [(tm-flags) .
```

*transaction-id*

identifies one or more transactions to be aborted:

*node*

is the name of the node (system) on which the transaction originated, which is known as the home node for the transaction. The syntax rules for specifying a node name appear under [Disk File Names](#), earlier in this section. The default for *node* is the node on which the TMFSERVE process communicating with your TMFCOM process is running.

*node-number*

is the number of the node on which the transaction originated (the home node). A node number specification consists of a backslash followed by the number assigned to the node. The default for *node-number* is the node on which the TMFSERVE process communicating with your TMFCOM process is running.

*tm-flags*

is a number representing flags used internally by TMF. If this number is zero, it does not appear in displays of the *transaction-id* presented by the STATUS TRANSACTIONS command. For more information about *tm-flags*, see the [DELETE TMF](#) command discussion.

---

**Note.** If the *tm-flags* value is a number other than zero, you must include this number in parentheses in the ABORT TRANSACTION command. For example:

---

TMF 10> **ABORT TRANSACTION \SYSB(1).3.45678**

---

*cpu*

is the number of the processor on which the transaction originated.

*seq-num*

is a sequence number assigned by TMF to the transaction, which distinguishes it from other transactions originating in the same processor on the same home node under the current TMF configuration. (A new TMF configuration is defined when the first START TMF command following a DELETE TMF command is executed.)

#### IGNOREDATAERRORS

requests that data errors be ignored while the specified transactions are being undone. Normally, when the backout process encounters data errors while undoing a transaction, it places the transaction in the hung state. When IGNOREDATAERRORS is specified, however, the backout process undoes as much of the transaction as possible and, for those files yielding data errors, ignores the errors as if the undo had been performed completely. If you omit IGNOREDATAERRORS, the backout process reacts in the normal way, acknowledging the errors and placing the transaction in the hung state.

- 
- △ **Caution.** Contact the Global Customer Support Center (GCSC) or your service provider before issuing this command with IGNOREDATAERRORS specified.

Specifying IGNOREDATAERRORS can lead to database corruption and loss of data integrity. Include this parameter only if all efforts to abort a transaction fail and data availability is more important than data integrity.

---

#### AVOIDHANGING

requests that files with data errors be noted and removed from access while the specified transactions are being undone.

If you specify AVOIDHANGING and a transaction involves one or more files with data errors, the backout process:

- Undoes as much of the transaction as possible.
- For those files yielding data errors:
  - Writes an audit trail record that preserves information about what parts of the transaction could not be undone.
  - Marks the file "undo-needed," so that it is no longer accessible to applications. (To restore the file to the system, you must typically use the file recovery process.)
- Aborts the transaction.

If you omit AVOIDHANGING and an error occurs during backout, the backout process does not abort the transaction but instead leaves it in the hung state.

## Security Restrictions

You can issue the ABORT TRANSACTION command only if you are a member of the super user group.

## TMF State Requirement

Before you can enter the ABORT TRANSACTION command, TMF must be started.

## Usage Guidelines

Only transactions that are in the active, aborting, or hung state can be aborted using this command. An active transaction is one that is in progress; an aborting transaction is one that is being backed out; a hung transaction is one that TMF attempted to abort but could not.

For best performance when aborting more than one transaction, issue a single ABORT TRANSACTION command with a list of transactions, rather than multiple ABORT TRANSACTION commands.

When you specify a list of transactions, the ABORT TRANSACTION command options apply to all the transactions in the list. TMFCOM prompts you to confirm the whole list collectively, rather than the specific transactions individually.

If a process fails while performing audited work on a temporary file, a hung transaction can result. If the transaction affected only temporary files that have now been removed from the system, issue an ABORT TRANSACTION command with the IGNOREDATAERRORS option, specifying the hung transaction.

---

△ **Caution.** If the transaction also affected any permanent disk files, however, do not use the IGNOREDATAERRORS option; in this case, IGNOREDATAERRORS can result in data corruption. Instead, either use the AVOIDHANGING option as discussed in the following paragraphs, or refrain altogether from entering the ABORT TRANSACTION command.

---

Use the AVOIDHANGING option when you want to remove a hung transaction without compromising the integrity of data. Use this option, for example, when data integrity is more important than the availability of a particular set of files.

When you specify AVOIDHANGING, however, the files that are affected by a transaction that could not be undone are marked “undo-needed”; these files are not available to any application until a volume or file recovery is done. All applications that opened the files before they were marked as “undo-needed” receive an error when they attempt to access those files again.

As an example, suppose you specify AVOIDHANGING in an ABORT TRANSACTION command for a transaction that affects two files. If the undo operation completes successfully on the first file but fails on the second file, only the second file is marked “undo-needed.”

The IGNOREDATAERRORS and AVOIDHANGING options are mutually exclusive, and cannot both be specified in the same ABORT TRANSACTION command.

## Examples

In the following example, the STATUS TRANSACTION command indicates that the transaction identified as \TSII.2.1958908 is active:

```
TMF 12> STATUS TRANSACTION
Transaction Identifier      Process      State  Parent  Children
-----
\TSII.2.1958908           (2,118)    active
\TSII.2.1968200           (2,118)    active
```

The ABORT TRANSACTION command, followed by the Y response to the confirm prompt, aborts that transaction:

```
TMF 15> ABORT TRANSACTION \TSII.2.1958908
Transaction Identifier      Process      State  Parent  Children
-----
\TSII.2.1958908           (2,118)    active
Confirm: y
```

The next STATUS TRANSACTION command verifies that the transaction, in fact, was aborted:

```
TMF 18> STATUS TRANSACTION
Transaction Identifier      Process      State  Parent  Children
-----
\TSII.2.1968200           (2,118)    active
```

# ADD AUDITTRAIL

The ADD AUDITTRAIL command adds an audit trail to the TMF configuration.

```
ADD AUDITTRAIL {MASTER | MAT
                {AUX[ILINARY]nn
                }
                [,AUDITDUMP {ON | OFF
                             }
                [,FILESIZE integer
                ]
                [,ACTIVEVOL[S] {volume
                                {(volume [,volume]...)
                                }
                ]
                [,FILESPERVOLUME integer
                ]
                [,OVERFLOWVOL[S] {volume
                                   {(volume [,volume]...)
                                   }
                ]
                [,OVERFLOWTHRESHOLD integer
                ]
                [,RESTOREVOL[S] {volume
                                 {(volume [,volume]...)
                                 }
                ]
                [,BEGINTRANSDISABLE integer
                ]
```

MASTER | MAT  
AUX[ILINARY]*nn*

identifies the audit trail to be added to the TMF configuration. MASTER or MAT specifies the master audit trail. AUXILIARY<sub>*nn*</sub> or AUX<sub>*nn*</sub> specifies an auxiliary audit trail. *nn* is a two-digit number ranging from 01 through 15 that specifies one of the 15 potential auxiliary audit trails in the configuration. (When the value of *nn* is 09 or less, you must specify the leading zero.)

Audit trails must be added to the TMF configuration in sequence, starting with the master audit trail, then the auxiliary audit trails AUX01, AUX02, and so forth.

The NonStop Remote Database Facility (RDF) relies on TMF for many of its functions. For information about RDF interoperability with TMF audit trails, see the NonStop RDF System Management Manual for your version of RDF.

AUDITDUMP {ON | OFF}

determines whether audit dumping is performed for the audit trail. During audit dumping, TMF automatically copies to tape or disk an audit-trail file that has become full. ON specifies that audit dumping takes place, and OFF prevents audit dumping. The default is ON.



If AUDITDUMP is ON, use the [ALTER AUDITDUMP](#) command to modify other parameters relating to audit dumping. These parameters are explained in the ALTER AUDITDUMP command description. Their default values are as follows:

BLOCKSIZE, the tape data block length: 28 (x 1024 bytes)

COPIES, the number of copies to be made in series or in parallel of each dump: 1 parallel

SYSTEM, the node (system) on which tape copies are made: the local node

VERIFYTAPE, the tape-verification selection: OFF

MEDIUM, the medium on which the dump is recorded: TAPE

---

**Note.** An auxiliary audit trail cannot be configured with AUDITDUMP ON unless the master audit trail is also configured with AUDITDUMP ON. A master audit trail cannot be configured with AUDITDUMP OFF unless the auxiliary audit trails are also configured with AUDITDUMP OFF. However, a master audit trail can be configured with AUDITDUMP ON regardless of whether AUDITDUMP is ON or OFF for the auxiliary audit trails. The possible combinations are:

| MAT   | AUX   | Allowed/Disallowed |
|-------|-------|--------------------|
| ----- | ----- | -----              |
| ON    | ON    | Allowed            |
| OFF   | OFF   | Allowed            |
| ON    | OFF   | Allowed            |
| OFF   | ON    | Disallowed         |

---

#### FILESIZE *integer*

specifies the size of the audit-trail files, in megabytes (MB). The FILESIZE attribute value, once assigned with the ADD AUDITTRAIL command, cannot be changed with the ALTER AUDITTRAIL command. To alter this value, you must purge the existing TMF configuration with the DELETE TMF command and then create an entirely new configuration.

The minimum value for FILESIZE is 1 and the maximum is either 2047 or the maximum disk size (if less than 2047 megabytes). The default is 100, resulting in files of 100 megabytes (100 x 1,048,576 bytes) each.

#### ACTIVEVOL[ S ]

specifies the disk volume that holds the audit-trail files. If ACTIVEVOL[S] was specified in a previous SET AUDITTRAIL command in this session, this parameter is optional in the ADD AUDITTRAIL command; otherwise, this parameter is required.

You must specify at least one volume; up to 16 are allowed. A single volume cannot be configured as an active volume for more than one audit trail. A volume configured as an active-audit volume cannot be configured as a data volume. To

add more audit volumes to the configuration later, you can use the ALTER AUDITTRAIL command.

---

**Note.** TMF is shipped from the factory with the \$AUDIT disk volume configured as the active-audit volume. You are free to retain \$AUDIT for this purpose, replace \$AUDIT with another volume as the recipient of audit trails, or add other active-audit volumes to the configuration. See also the note about the preconfiguration of data volumes that appears under the discussion of the [ADD DATAVOLS](#) command.

---

Volumes of the same size should be used for a specific audit trail. TMF allocates the same number of files on each volume, and all volumes must be able to hold all of those files. If a mixture of large and small disks is used for an audit trail, some space on the large disks will remain unused because this space will receive neither the active-audit trail nor the audited files. The unused space, however, could be configured as overflow space or audit restore space.

FILESPERVOLUME *integer*

specifies the number of files in the audit trail that reside on each active-audit volume configured. The audit trail's capacity is determined by the number of audit-trail files per volume, the number of active-audit volumes, and the file sizes. TMF pre-allocates these files when the ADD AUDITTRAIL command is executed. If TMF cannot allocate all the files, this command fails.

The minimum value for FILESPERVOLUME is 1 and the maximum is 100. The default is 5.

---

△ **Caution.** The number of active-audit volumes multiplied by the value of FILESPERVOLUME must be 2 or more.

---

To change the FILESPERVOLUME value later, you can use the ALTER AUDITTRAIL command.

OVERFLOWVOL[S]

specifies the volume used as overflow space for the audit trail. Although this attribute is not required by TMF, if you omit it, TMF issues a message at TMF startup, indicating that an overflow-audit volume should be configured.

You can specify from 1 to 16 overflow-audit volumes. These can be the same volumes as the active-audit volumes; they can also be data volumes or any volumes not configured for other TMF purposes. To change the overflow-audit volumes later, you can use the [ALTER AUDITTRAIL](#) command.

For the [START TMF](#) command to execute successfully, all overflow-audit volumes must be up.

OVERFLOWTHRESHOLD *integer*

specifies the percentage of audit trail capacity that can be consumed before TMF begins overflow copying. This is an optional parameter. The minimum value is 50 and the maximum is 100. The default is 80(%).

RESTOREVOL[S]

specifies the volumes used when restoring audit dumps during execution of a [RECOVER FILES](#) command. If audit dumping is configured for the audit trail, you must specify at least one volume for this purpose.

You can specify from 1 to 16 restore-audit volumes. These can be the same volumes as the active-audit volumes; they can also be data volumes or any volumes not configured for other TMF purposes. To change the restore-audit volumes later, you can use the [ALTER AUDITTRAIL](#) command.

BEGINTRANSDISABLE *integer*

specifies the percentage of audit trail capacity that can be consumed before TMF disables new transactions. When the consumed capacity falls below this threshold, new transactions are no longer disabled by audit-trail usage. (However, new transactions can be disabled for other reasons, as explained in the [ALTER BEGINTRANS](#) command description.) The minimum value is 50 and the maximum is 100. The default is 90(%).

## Security Restrictions

You can issue the ADD AUDITTRAIL command only if you are a member of the super user group.

## TMF State Requirement

You can only issue ADD AUDITTRAIL commands under either of two conditions:

1. Before you start TMF for the first time with the [START TMF](#) command.
2. After you stop TMF with the STOP TMF command and purge the existing TMF configuration with the [DELETE TMF](#) command but before you issue the next [START TMF](#) command.

Under either of these conditions, you can issue multiple ADD AUDITTRAIL commands.

## Usage Guidelines

There must be exactly one master audit trail in a TMF system. In addition, there can be up to 15 optional auxiliary audit trails. Auxiliary audit trails receive the audit information generated by a specified set of data volumes; they do not contain TMF control information. Auxiliary audit trails can be added to a TMF configuration to increase overall auditing capacity and bandwidth, or to logically separate operations

related to sets of data volumes. Each auxiliary audit trail is uniquely identified by the prefix AUX followed by a two-digit number: for example, AUX03.

For attributes not specified in the ADD AUDITTRAIL command, you can assign values by entering [SET AUDITTRAIL](#) commands earlier in the same TMFCOM session; otherwise, the default values, if they exist, are used.

Before it takes effect, the ADD AUDITTRAIL command verifies that the volumes specified do not contain any files in the ZTMFAT subvolume, which is reserved for use by audit trails created under the current TMF configuration. You can use the [DELETE TMF](#) command to purge any audit-trail files left over from a previous TMF configuration.

TMF uses the FILESPERVOLUME value to pre-allocate the number of files in the active-audit trail that reside on each volume; if they cannot all be allocated, the ADD AUDITTRAIL command fails. You can change the FILESPERVOLUME value by using the [ALTER AUDITTRAIL](#) command, whether TMF is started or not.

TMF calculates disk extent size on the basis of 16 extents per file, with all extents (both primary and secondary) being the same size.

Overflow-audit volumes are used to store files when the active-audit trail becomes too full. At this point, the oldest audit-trail files are copied to the overflow volumes so that the space occupied by these audit-trail files can be reused. In the OVERFLOWVOLS parameter, you can specify the overflow-audit volumes you want TMF to use for this purpose. In the OVERFLOWTHRESHOLD parameter, you can specify the capacity level of the active-audit trail that will trigger overflow copying. For example, if you want overflow copying to begin when 95% of the available audit trail space has been used, set OVERFLOWTHRESHOLD to 95. The time it takes to reach the overflow threshold depends on the audit generation rate and the total audit trail capacity.

Once overflow copying begins, it continues until the capacity used falls below the threshold, even though space subsequently becomes available on the active-audit volume. When they are no longer needed, overflow files are automatically purged by TMF.

Restore-audit volumes are used to receive audit-trail files restored from an audit dump as part of a recovery procedure. You can specify the restore-audit volumes associated with the audit trail through the RESTOREVOLS parameter.

Configuring an active-audit volume as an overflow- or restore-audit volume might result in contention for the disk, adversely impacting performance. To avoid this, you should consider disk capacity, available free space, and probable audit activity when configuring.

Another attribute related to audit trail capacity is the BEGINTRANSDISABLE parameter. This value indicates the percentage of audit trail capacity that can be consumed before TMF automatically disables new transactions. This value must be low enough to ensure that existing transactions have enough room in the active-audit trail for completion. The appropriate value depends on the transaction load, the demands of the applications, and other factors prevailing in the system.

If the primary transaction management process (TMP) fails before the ADD AUDITTRAIL command completes, the backup TMP takes over and undoes the effects of that command before processing any new requests from TMFCOM. If both TMPs fail before the ADD AUDITTRAIL command completes, TMF also undoes the effects of that command.

If the ADD AUDITTRAIL command returns an error, files that were preallocated when active volumes were added might not be purged from the volumes; this occurs if the TMP loses access to these volumes. If you attempt to add these volumes as active again with another ADD AUDITTRAIL command, the existing files are purged before the new ones are created.

For more information about master and auxiliary audit trails, overflow volumes, and restore-audit volumes, see the *TMF Operations and Recovery Guide*.

## Examples

The following ADD AUDITTRAIL command adds a master audit trail using the disk volumes \$MAT1, \$MAT2, and \$MAT3 for the audit-trail files:

```
TMF 10> ADD AUDITTRAIL MAT, &
>>>ACTIVEVOLS($MAT1,$MAT2,$MAT3), &
>>>FILESIZE 20, FILESPERVOLUME 3,&
>>>OVERFLOWVOLS($DATA1,$DATA2),&
>>>OVERFLOWTHRESHOLD 80, BEGINTRANSDISABLE 95,&
>>>RESTOREVOLS $DATA1, &
>>>AUDITDUMP ON
```

The next ADD AUDITTRAIL command adds an auxiliary audit trail identified as AUX01, using the disk volume \$AUX1 for the active-audit volume and \$DATA3 for the overflow and restore locations:

```
TMF 11> ADD AUDITTRAIL AUX01, FILESPERVOLUME 6, ACTIVEVOLS &
>>>$AUX1, OVERFLOWVOLS $DATA3, RESTOREVOLS $DATA3, AUDITDUMP ON
```

# ADD DATAVOLS

The ADD DATAVOLS command adds one or more data volumes to the TMF configuration, including management attributes such as the audit trail to receive audit records from these volumes.

```
ADD DATAVOL[S] { volume
                  { (volume [,volume]...)
                    * [,NOT volume]
                    * [,NOT (volume[,volume]...)] }
                  [ ,AUDITTRAIL { MASTER | MAT
                                { AUX[ILIARY]nn } } ]
                  [ ,RECOVERYMODE { ONLINE | ARCHIVE } ]
                  [ ,IGNOREPREVIOUSCONFIG ]
```

*volume*

identifies a disk volume to be added as a data volume.

---

**Note.** TMF is shipped from the factory with the \$DSMSCM disk volume configured as a TMF data volume. Although you are free to add other TMF data volumes to the configuration, you should not remove \$DSMSCM as a TMF data volume. See also the note about the preconfiguration of active-audit volumes that appears under the discussion of the [ADD AUDITTRAIL](#) command.

---

\*

adds, as data volumes, all accessible disk volumes in the system that are not already configured as data volumes or active-audit volumes. Exceptions are the volumes listed in the NOT option, if any.

NOT *volume*

used with the asterisk indicator (\*) to identify one or more disk volumes that are not to be added as data volumes. Use this option to exclude specific volumes from the configuration.

```
AUDITTRAIL { MASTER | MAT }
            { AUX[ILIARY]nn }
```

identifies the audit trail to which the newly added data volumes send their audit records. MASTER or MAT specifies the master audit trail. AUXILIARY<sub>nn</sub> or AUX<sub>nn</sub> specifies an auxiliary audit trail. *nn* is a two-digit number ranging from 01 through 15 that specifies one of the 15 potential auxiliary audit trails in the configuration. (When the value of *nn* is 09 or less, you must specify the leading zero.) The default is MASTER.

The NonStop Remote Database Facility (RDF) relies on TMF for many of its functions. For information about RDF interoperability with TMF audit trails, see the NonStop RDF System Management Manual for your version of RDF.

`RECOVERYMODE { ONLINE | ARCHIVE }`

determines whether audit-trail files are held on disk until they are no longer required for volume recovery.

ONLINE retains sufficient audit-trail files on disk so that volume recovery does not need to restore audit information from an archived audit dump. Retaining audit-trail files on disk can affect audit trail management and transaction processing; transaction processing, for example, is disabled if it causes the audit trail to exceed a capacity threshold.

ARCHIVE releases audit-trail files so that they are not retained on disk solely for the purpose of volume recovery. If volume recovery becomes necessary, one or more audit dumps might need to be restored. ARCHIVE is allowed only if audit dumping is configured for the associated audit trail.

The default is ONLINE.

You can change the RECOVERYMODE attribute later by using the ALTER DATAVOLS command.

`IGNOREPREVIOUSCONFIG`

specifies that a data volume is to be added to the configuration regardless of whether it was associated with a previous TMF configuration and thus might be inconsistent.

If you omit IGNOREPREVIOUSCONFIG and TMF encounters either of the following conditions, the ADD DATAVOLS command fails:

- Any volume designated for the current TMF configuration was part of a previous TMF configuration and was not deleted from that configuration.
- Any volume designated for the current TMF configuration contains files that might have been in an inconsistent state when the volume was deleted.

Specifying IGNOREPREVIOUSCONFIG overrides this protective mechanism and allows TMF to add the volume despite possible inconsistencies.

---

△ **Caution.** Volume recovery cannot repair any inconsistencies that exist when a data volume is added with IGNOREPREVIOUSCONFIG specified, even if the data volume is returned to its previous configuration environment.

---

## Security Restrictions

You can issue the ADD DATAVOLS command only if you are a member of the super user group.

## TMF State Requirement

Before you can enter the ADD DATAVOLS command, TMF must be started.

## Usage Guidelines

When the ADD DATAVOLS command completes successfully, TMF automatically recognizes the specified volumes and immediately attempts to start them.

In the ADD DATAVOLS command, you can specify only unconfigured volumes; furthermore, the audit trail you specify must already exist in the configuration.

Attributes not specified in the ADD DATAVOLS command are given values set by any [SET DATAVOLS](#) commands issued earlier in the same TMFCOM session; otherwise, the default values, if they exist, are used.

After a data volume is added to the TMF configuration, you cannot change its audit trail usage until you first delete the volume from the configuration. The ADD DATAVOLS command associates each data volume with an audit trail. If you have configured more than one audit trail, and you want to switch the data volume from one audit trail to another, you must delete the data volume from the configuration, and then add it again.

Adding a data volume potentially increases the number of audit records generated. So, the audit trail you specify to store records for the new data volume must be large enough to accommodate those records: you may have to increase the size of the audit trail.

Setting RECOVERYMODE to ONLINE means that the audit-trail files required to recover the data volume remain on the active or overflow-audit volumes. This way, the volume recovery process does not have to restore audit trail files from archive media. Because ONLINE keeps audit-trail files pinned on the active-audit trail longer, there must be sufficient space to hold the files until they are no longer needed for volume recovery. (A pinned file is an audit-trail file that cannot be reused or purged from an active or overflow-audit volume because it contains or is receiving audit information.)

Setting RECOVERYMODE to ARCHIVE means that the audit-trail files required to recover this data volume are copied to tape or disk. ARCHIVE is allowed only if the audit trail for this data volume is configured for audit dumping. ARCHIVE frees audit trail space sooner, which can prevent overflow or begin-transaction-disable conditions.

If you specify ARCHIVE, be sure that the TMF catalog retains sufficient audit dumps to recover the data volume. Perform regular online dumps to ensure that the audit dumps stay current in the catalog. If the catalog does not contain an online dump for any file on a data volume configured with RECOVERYMODE set to ARCHIVE, TMF may reuse the media containing the required audit dumps, making volume recovery impossible.

You should usually configure data volumes with online recovery mode, which is the default. Archive mode might be used only in exceptional circumstances; it could be used, for example, when a volume will be down for a long time while other OLTP operations must not be disrupted, and the volume will later require recovery. In these



circumstances, consider temporarily deleting a data volume if it was shut down cleanly or does not contain critical data.

## Examples

The following ADD DATAVOLS command adds three data volumes to the TMF configuration:

```
TMF 30> ADD DATAVOLS ($DATA, $DATA2, $DATA17), AUDITTRAIL MAT  
$DATA -- datavol added.  
$DATA2 -- datavol added.  
$DATA17 -- datavol added.
```

The next ADD DATAVOLS command attempts to add a data volume that is already configured as a data volume:

```
TMF 31> ADD DATAVOLS $DATA2  
ERROR: $DATA2 -- not added (already configured as a datavol).
```

# ADD DUMPS

The ADD DUMPS command adds dump entries to the TMF catalog. This command is used to reenter online and audit dump entries accidentally deleted from the catalog, or to recover a destroyed catalog. Use the ADD DUMPS command only under one of these conditions:

- The catalog files residing on the configuration volume in the ZTMFCONF subvolume (*volume.ZTMFCONF*) have been damaged or lost.
- Entries for dumps have been mistakenly deleted from the catalog with a [DELETE DUMPS](#) command.
- A file is mistakenly purged.
- The AUDIT attribute of a file is mistakenly changed to NO AUDIT.

---

**Note.** The ADD DUMPS command only re-creates entries in the TMF catalog, it does not re-create actual lost or damaged dumps themselves. The dump entries are only effective if the dumps they describe actually exist on available media. That is, a dump to tape must remain on a tape that has not been overwritten; a dump to disk must have been restored from backup disk files.

HP recommends that you use output from a previous INFO DUMPS, OBEYFORM command as input for executing ADD DUMPS commands. The INFO DUMPS, OBEYFORM command formats online and audit-dump entries from the TMF catalog as command file text that can be saved and executed later by TMFCOM. If you periodically issue the INFO DUMPS, OBEYFORM command, you can readily capture up-to-date information for rebuilding the entries in the TMF catalog. For more information about this approach, see “Usage Guidelines” later in this command discussion, and also read the [INFO DUMPS](#) command discussion.

---

For further information, refer to “Usage Guidelines” later in this command description, and to the *TMF Operations and Recovery Guide*.

There are two forms of the ADD DUMPS command: one for adding an audit dump entry to the catalog and another for adding one or more online dump entries to the catalog. The syntax for each form follows.

## Adding an Audit Dump to the Catalog

To add an audit dump entry to the catalog, use the following syntax:

```
ADD DUMPS audit-trail-file

    ,SERIAL serial-number

    ,TYPE AUDITDUMP

    { ,DISKMEDIA disk }
    { ,TAPEMEDIA { tape } }
    { ( tape [, tape] ... ) }

    [ ,TIME { month[-]day[-]year, hr:min[:sec] [GMT]}
           { day[-]month[-]year, hr:min[:sec] [GMT]}]
```

*audit-trail-file*

identifies the audit-trail file dumped, using the following file name format:

`$ZTMFAT.ZTMFAT.file-id`

`$ZTMFAT` and `ZTMFAT` are pseudo or dummy volume and subvolume names, respectively, used to support TMF's ability to spread audit-trail files across multiple disk volumes.

SERIAL *serial-number*

specifies the serial number of the audit dump. You must not omit *serial-number* from this command. The value for *serial-number* should be the serial number of the original audit dump, which you can find in the output from an [INFO DUMPS](#) command entered before that dump was deleted. If you enter a number less than 64 for *serial-number*, TMF returns an error message.

TYPE AUDITDUMP

identifies the dump as an audit dump. You must include this parameter if you are adding an audit dump entry to the catalog.

DISKMEDIA

for an audit dump to disk, identifies the disk volume on which the dump is located. Only one disk is allowed when specifying *disk* for an audit dump. The format for *disk* is:

`[node.]volume.{xx}serial-number.{zz}sequence-number`

*node* is the node where the dump is located.

*volume* is the volume on which the dump is located.

`{xx}serial-number` identifies the dump. *xx* is a two-character prefix, and *serial-number* is the six-digit serial number of the dump. When *node* specifies

the node at which your TMFSERVE process is running, *xx* can be any two alphabetic characters. When *node* is different than the one at which your TMFSERVE process is running, *xx* can be any two alphabetic characters except ZT. If *node* is not specified, ZT is assigned as *xx* by default. Examples are:

AB001015 (for a node local to your TMFSERVE process)

ZT001016 (for a node local to your TMFSERVE process)

XY001017 (for a node remote to your TMFSERVE process)

{*zz*}*sequence-number* identifies the audit trail sequence. *zz* is the audit dump prefix (AA for the master audit trail, BB for the first auxiliary audit trail, and so forth). *sequence-number* is the six-digit current sequence number of the audit trail when dumped. For example:

AA000036

When the disk resides on a local node, specifying the node name is optional. If the node is not specified, the default is the node on which your TMFSERVE process is running.

When the disk resides on a remote node, *node* is required and *volume* must not exceed seven characters, counting the leading dollar sign (\$).

Use the output from a previous INFO DUMPS command to obtain the audit dump name and the exact time of the dump.

If you specify DISKMEDIA, you must not specify TAPEMEDIA; if you do, the command fails and an error is reported.

#### TAPEMEDIA

for an audit dump to tape, identifies the tape volume on which the dump is stored. *tape* is specified as:

*tape-volume:part:copy*

where:

*tape-volume*

is the name of the tape volume.

*part*

specifies which part of the audit dump is stored on this tape volume. For example, if each copy of the dump requires three tape volumes, *part* is 1, 2, or 3. Because *part* must be specified, it does not have a default value.

*copy*

specifies the number of the dump copy represented on this tape volume. The maximum value for *copy* is 2. For serial copies of audit dumps being added to the catalog, *copy* should be 1. Because *copy* must be specified, it does not

have a default value. If you specify 2 for *copy* in the ADD DUMPS command, always supply the part numbers of both copies of the dump in the TAPEMEDIA parameter, for example:

```
TAPEMEDIA (TAPE1:1:2, TAPE2:2:2)
```

#### TIME

specifies the time that the audit dump was created. If TIME is not specified, the current TMFSERVE time is assumed by default—that is, the current time is recorded into the catalog with the audit dump whose name is being added.

- 
- △ **Caution.** If recovery is needed, problems could occur if the time listed in the catalog is not the time when the audit dump actually took place; if you specify the TIME option, make sure to enter the date and time completely and accurately.
- 

To obtain the exact time, to the second, of the audit dump, use the output from a previous INFO DUMPS command. In the TIME parameter, the subparameters are the following:

#### *month*

is a three-character abbreviation for the name of the month:

```
{JAN|FEB|MAR|APR|MAY|JUN|JUL|AUG|SEP|OCT|NOV|DEC}
```

#### *day*

is a two-digit integer for the day of the month.

#### *year*

is a four-digit integer for the calendar year.

---

**Note.** You can use either dashes (-) or spaces, or both, as delimiters to separate the *month*, *day*, and *year* values from one another, but you must use at least one of these delimiters.

---

#### *hr*

is an integer for the hour, from 0 through 23.

#### *min*

is an integer for the minute, from 00 through 59.

#### *sec*

is an integer for the second, from 00 through 59.

#### GMT

indicates that the TIME value is specified in Greenwich mean time. If GMT is omitted, the default is the local time at your site.

## Adding Online Dump File Names to the Catalog

To add one or more online dump entries to the catalog, use the following syntax:

```
ADD DUMPS {online-dump-file
           {(online-dump-file [,online-dump-file]...)}

           ,SERIAL serial-number

           ,TYPE ONLINEDUMP (master, data)

           {,DISKMEDIA {disk
                        {(disk [,disk]...)}}}
           {,TAPEMEDIA {tape
                        {(tape [,tape]...)}}}

           [,TIME {month[-]day[-]year, hr:min[:sec] [GMT]}
                  {day[-]month[-]year, hr:min[:sec] [GMT]}]
```

*online-dump-file*

identifies an audited file dumped by the online dump process, using the following file name format:

```
[volume.][subvolume.]file-id
```

If you specify two or more files, you must list them in the same order in which they are recorded on the dump tape or disk.

SERIAL *serial-number*

specifies the serial number of the online dump. The value for *serial-number* should be the serial number of the original online dump. If you enter a number less than 64, an error message is returned.

TYPE ONLINEDUMP (*master*, *data*)

identifies the dump as an online dump. You must include this parameter.

*master*

is an integer specifying the master audit trail sequence number. You can obtain MAT sequence numbers by examining output from a previous INFO DUMPS command or a previous DUMP FILES command.

*data*

is an integer specifying the auxiliary audit trail sequence number. You can obtain audit trail sequence numbers by examining output from a previous INFO DUMPS command, or a previous DUMP FILES command.

---

**Note.** If only a master audit trail is configured, the values for *master* and *data* are identical.

---

## DISKMEDIA

for an online dump to disk, identifies the disk volume on which the dump is located. Any number of disks can be specified for an online dump, without limit. The format for *disk* is:

*[node.]volume.{xx}serial-number.Ffile-number*

*node* is the node where the dump is located.

*volume* is the volume on which the dump is located.

*{xx}serial-number* identifies the dump. *xx* is a two-character prefix, and *serial-number* is the six-digit serial number of the dump. When *node* specifies the node at which your TMFSERVE process is running, *xx* can be any two alphabetic characters. When *node* is different than the node at which your TMFSERVE process is running, *xx* can be any two alphabetic characters except ZT. If *node* is not specified, ZT is assigned as *xx* by default. Examples are:

JK001020 (for a node local to your TMFSERVE process)

ZT001021 (for a node local to your TMFSERVE process)

YZ001022 (for a node remote to your TMFSERVE process)

*file-number* is the number of the online dump file that indicates its sequence in the fileset. There are multiple files for each online dump. The *file-number* is always preceded by the prefix F. Thus, the first file recorded is identified as F0000001, the second file is identified as F0000002, and so forth.

When the disk resides on a local node, specifying the node name is optional. If the node is not specified, the default is the node on which your TMFSERVE process is running.

When the disk resides on a remote node, *node* is required and *volume* must not exceed seven characters, counting the leading dollar sign (\$).

To obtain the online dump name and the exact time of the dump, use the output from a previous INFO DUMPS command.

If you specify DISKMEDIA, you must not specify TAPEMEDIA; if you do, the command fails and an error is reported.

## TAPEMEDIA

for an online dump to tape, identifies the tape volume on which the online dump is stored. *tape* is specified as:

*tape-volume:part:copy*

where:

*tape-volume*

is the name of the tape volume.

*part*

specifies which part of the online dump is stored on this tape volume. For example, if each copy of the dump requires three tape volumes, *part* is 1, 2, or 3. Because *part* must be specified, it does not have a default value.

*copy*

specifies the copy number of the online dump stored on this tape volume. The maximum for *copy* is 2. Because *copy* must be specified, it does not have a default value. If you specify 2 for *copy* in the ADD DUMPS command, always supply the part numbers of both copies of the dump in the TAPEMEDIA parameter, for example:

```
TAPEMEDIA (TAPE1:1:2, TAPE2:2:2)
```

## TIME

specifies the time that the online dump was created. If TIME is not specified, the current TMFSERVE time is assumed by default—that is, the current time is recorded into the catalog with the online dump whose name is being added.

---

△ **Caution.** If recovery is needed, problems could occur if the time listed in the catalog is not the time when the online dump actually took place; if you specify the TIME option, make sure to enter the date and time completely and accurately.

---

To obtain the exact time, to the second, of the online dump, use the output from a previous INFO DUMPS command. In the TIME parameter, the subparameters are:

*month*

is a three-character abbreviation for the name of the month:

{JAN|FEB|MAR|APR|MAY|JUN|JUL|AUG|SEP|OCT|NOV|DEC}

*day*

is a two-digit integer for the day of the month.



*year*

is a four-digit integer for the calendar year.

---

**Note.** You can use either dashes (-) or spaces, or both, as delimiters to separate the *month*, *day*, and *year* values from one another, but you must use at least one of these delimiters.

---

*hr*

is an integer for the hour, from 0 through 23.

*min*

is an integer for the minute, from 00 through 59.

*sec*

is an integer for the second, from 00 through 59.

GMT

indicates that the TIME option is specified in Greenwich mean time. If GMT is omitted, the default is the local time at your site.

## Security Restrictions

You can issue the ADD DUMPS command only if you are a member of the super user group.

## TMF State Requirement

Before you can enter the ADD DUMPS command, TMF must be started.

## Usage Guidelines

When an audit or online dump occurs, information describing the dump is recorded in the TMF catalog. Dump entries contain the names of the files dumped, the names of the tape or disk volumes containing the dumps, dump serial numbers, and other related information. If a dump entry is damaged, lost, or mistakenly deleted, you can restore it by issuing the ADD DUMPS command.

---

**Note.** When working with the ADD DUMPS command, you can use output from a previous INFO DUMPS, OBEYFORM command rather than manually entering the data based on dump reports. This approach can make your effort much simpler and easier.

At frequent intervals, use the INFO DUMPS, OBEYFORM command provided by TMFCOM to create a command file for re-adding the dump entries. This command reads the audit or online dump entries in the TMF catalog and creates a command file containing TMFCOM ADD DUMPS commands referencing those dump entries. If you save this file, you can use it later to collectively re-add the entries to the TMF catalog at any time through the TMFCOM OBEY command.

If you do not use the INFO DUMPS, OBEYFORM command, however, the only alternative way to create a command file that has information for rebuilding dump entries is to use the SCANADD DUMPS command provided by the SNOOP utility. This command reads the dumps on an audit dump tape or online dump tape and creates a command file containing TMFCOM ADD DUMPS commands referencing those dumps. You can then re-add the dump entries by using the command file as input to the TMFCOM OBEY command. The SNOOP utility is provided with the TMF product, and is documented in the file named \$SYSTEM.SYSnn.SNOOPDOC. For more information about the SCANADD DUMPS command, see the SNOOPDOC file.

---

The ADD DUMPS command can add to the TMF catalog dump entries for files on virtual and physical volumes managed by the NonStop Storage Management Foundation (SMF). Physical volumes within the SMF storage pool must be enabled for TMF, however, or TMFCOM Error 283 may occur.

Each audit dump to disk is associated with two disk file records: one for the DISKMEDIA file and the other for the LABELS file. When you enter the ADD DUMPS command, TMF adds the DISKMEDIA record and also automatically creates the LABELS file and adds its record.

Always include the TIME option in this command to avoid problems during recovery. If you do not specify TIME, the time and date that you issue an ADD DUMPS command are included in the dump entry inserted into the TMF catalog.

- 
- △ **Caution.** Without the TIME option, even if you add the dump entries in the order in which the dumps were made (that is, in ascending dump serial-number sequence), you risk destruction of your database if recovery procedures become necessary; when you issue a RECOVER FILES command to initiate a file recovery to the timestamp, you will probably not recover all the data you need.
- 

You must use separate ADD DUMPS commands to add to the TMF catalog entries for dumps having different audit trail sequence numbers.

You must specify either the TAPEMEDIA option or the DISKMEDIA option, but you cannot specify both.

You do not have to use the [ADD TAPEMEDIA](#) command to insert entries for the tape volumes into the TMF catalog; they are inserted automatically by the ADD DUMPS command.

When adding entries for a multi-copy dump to the TMF catalog, be sure to specify part numbers for all copies of the dump in the TAPEMEDIA option of the same ADD DUMPS command. Otherwise, the command might either abend or add the dump information incorrectly.

When adding an online dump entry for a file that begins on a tape reel other than the first reel identified by the dump serial number, also add an online dump entry for at least one file on each preceding reel associated with the same dump serial number. This ensures that the correct dump reel number is entered in the TMF catalog, permitting subsequent RECOVER FILES commands that use the added dump to execute correctly. Without this precaution, the RECOVER FILES commands can fail, resulting in an EMS error of the following format:

```
*ERROR* ArchTape-8125: Wrong tape mounted (Found tape #14, expected tape #1).
      TMF Catalog has incorrect information.
```

For complete information about audit and online dumps and how they are used, and details about the TMF catalog and its contents, see the *TMF Operations and Recovery Guide*.

## Examples

The first example shows how to add an audit dump entry; the other two examples illustrate adding online dump entries.

The following ADD DUMPS command restores to the TMF catalog an entry for an audit dump of the audit-trail file \$ZTMFAT.ZTMFAT.AA000024, which was dumped to the disk file \$DUMP.ZT001055.AA000024 at 17:20:55 on March 4, 2002:

```
TMF 32> ADD DUMPS $ZTMFAT.ZTMFAT.AA000024, SERIAL 1055,&
>>>TYPE AUDITDUMP,DISKMEDIA                                &
>>>$DUMP.ZT001055.AA000024, TIME MAR 4 2002, 17:20:55
```

This ADD DUMPS command restores an entry for an audit dump of the audit-trail file \$ZTMFAT.ZTMFAT.AA000341, which was dumped to the tape volume TF0258 at 01:42:35 on February 26, 2002:

```
TMF 33> ADD DUMPS $ZTMFAT.ZTMFAT.AA000341, SERIAL 4253, &
>>>TYPE AUDITDUMP, TIME 26-Feb-2002 01:42:35 GMT, &
>>>TAPEMEDIA (TF0258:1:1)
```

The next ADD DUMPS command restores to the TMF catalog an entry for an online dump for the file \$DATA3.TMFNS02F.FILEA, which is stored on the tape volume T6250:

```
TMF 34> ADD DUMPS $DATA3.TMFNS02F.FILEA, SERIAL 121, &
>>>TAPEMEDIA T6250:1:1, TYPE ONLINEDUMP (18,18), &
>>>TIME MAR 12 2002, 10:15:12
```

The subsequent ADD DUMPS command restores to the catalog an entry for an online dump of the file \$DATA3.TMFN12.MAILFNS, which was dumped to the disk file \REMOTE.\$DUMP.YC001000.F0000001 at 10:20:15 on March 12, 2002:

```
TMF 35> ADD DUMPS $DATA3.TMFN12.MAILFNS, SERIAL 1000, &
>>>TYPE ONLINEDUMP (50,50), DISKMEDIA &
>>>\REMOTE.$DUMP.YC001000.F0000001, TIME MAR 12 2002, 10:20:15
```

The next series of ADD DUMPS commands was written to a command file by using an INFO DUMPS, OBEYFORM command. This example illustrates how easy it is to generate multiple ADD DUMPS commands if you use this approach rather than entering the commands manually. When the command file is executed, these commands restore entries for online dumps that created two parallel copies of 999 files, using the tape sets TMF10, TMF20, and TMF30 for the first copy and TMF11, TMF21, and TMF31 for the second copy. FILE025 continues from the first to the second volume, and FILE500 continues from the second to the third volume. Notice that for each file dumped, INFO DUMPS, OBEYFORM generates a separate ADD DUMPS command. (For more information about producing ADD DUMPS commands in this way, see [INFO DUMPS](#) on page 3-165.)

```
ADD DUMPS $DATA1.HRDATA.FILE001, SERIAL 101, &
TYPE ONLINEDUMP (3,3), TIME 14-Mar-2002 05:30:15 GMT, &
TAPEMEDIA (TAPE10:1:1, TAPE11:1:2)
```

```
ADD DUMPS $DATA1.HRDATA.FILE002, SERIAL 101, &
TYPE ONLINEDUMP (3,3), TIME 14-Mar-2002 05:30:15 GMT, &
TAPEMEDIA (TAPE10:1:1, TAPE11:1:2)
```

```
.
.
.
```

```
ADD DUMPS $DATA1.HRDATA.FILE024, SERIAL 101, &
TYPE ONLINEDUMP (3,3), TIME 14-Mar-2002 05:30:15 GMT, &
TAPEMEDIA (TAPE10:1:1, TAPE11:1:2)
```

```
ADD DUMPS $DATA1.HRDATA.FILE025, SERIAL 101, &
TYPE ONLINEDUMP (3,3), TIME 14-Mar-2002 05:32:15 GMT, &
TAPEMEDIA (TAPE10:1:1, TAPE11:1:2, TAPE20:2:1, TAPE21:2:2)
```

```
ADD DUMPS $DATA1.HRDATA.FILE026, SERIAL 101, &
TYPE ONLINEDUMP (3,3), TIME 14-Mar-2002 05:34:15 GMT, &
```

```
TAPEMEDIA (TAPE20:1:1, TAPE21:1:2)
```

```
.  
. .  
.
```

```
ADD DUMPS $DATA1.HRDATA.FILE499, SERIAL 101, &  
  TYPE ONLINEDUMP (3,3), TIME 14-Mar-2002 05:34:15 GMT, &  
  TAPEMEDIA (TAPE20:1:1, TAPE21:1:2)
```

```
ADD DUMPS $DATA1.HRDATA.FILE500, SERIAL 101, &  
  TYPE ONLINEDUMP (3,3), TIME 14-Mar-2002 05:32:15 GMT, &  
  TAPEMEDIA (TAPE20:1:1, TAPE21:1:2, TAPE30:2:1, TAPE31:2:2)
```

```
ADD DUMPS $DATA1.HRDATA.FILE501, SERIAL 101, &  
  TYPE ONLINEDUMP (3,3), TIME 14-Mar-2002 05:34:15 GMT, &  
  TAPEMEDIA (TAPE30:1:1, TAPE31:1:2)
```

```
.  
. .  
.
```

```
ADD DUMPS $DATA1.HRDATA.FILE999, SERIAL 101, &  
  TYPE ONLINEDUMP (3,3), TIME 14-Mar-2002 05:34:15 GMT, &  
  TAPEMEDIA (TAPE30:1:1, TAPE31:1:2)
```

# ADD RESOURCEMANAGER

The ADD RESOURCEMANAGER command adds one or more recoverable resource managers to the resource manager directory.

```
ADD {RESOURCEMANAGER | RM} {rm-name
                               { (rm-name [ ,rm-name ] ... ) }
    ,OWNER {username | userid}
where username is:
    group-name.user-name
and userid is:
    (group-id,user-id)
```

*rm-name*

is the name of the recoverable resource manager to be added to the directory. This name can contain a maximum of 128 characters; the first character must be a letter, and the others can be alphanumeric or any of the following special characters:

\$ \_ - ^ @ &

Names that begin with ZZ-VOLATILE- are not allowed in this parameter. Names containing adjoining hyphens (--) are not allowed, although names using multiple hyphens separated by one or more intervening characters are legal. (TMFCOM interprets two consecutive hyphens as the start of a comment.) Thus, the name "ALPHA--1" is not permitted, but "ALPHA-N-1" is allowed.

*username*

specifies the user name, in the format *group-name.user-name*, of the user who owns the resource manager. This identifier restricts the set of users who can open or remove the resource manager from the directory: only processes running under this user name or the super user group can open or remove the resource manager.

*userid*

specifies the user id, in the format *(group-id,user-id)*, of the user who owns the resource manager. This identifier restricts the set of users who can open or remove the resource manager from the directory: only processes running under this user ID or the super user group can open or remove the resource manager.

## Security Restrictions

You can issue the ADD RESOURCEMANAGER command only if you are a member of the super user group.

## TMF State Requirement

Before you can enter the ADD RESOURCEMANAGER command, TMF must be started.

## Usage Guidelines

In heterogeneous transaction processing, TMF can start a transaction and then subcontract portions (branches) of the transaction to one or more foreign transaction management systems operating on platforms other than the NonStop system. Alternatively, a foreign transaction management system can begin a transaction and then subcontract parts to TMF. After completing the transaction, TMF and the foreign transaction manager participate in an agreement protocol to determine the outcome of the transaction. This cooperation between different transaction management systems running on different platforms relies on collections of routines called resource managers.

---

**Note.** Heterogeneous transaction processing, and the resource managers that support this processing, are used in products such as NonStop TUXEDO. Although TMFCOM and the TMFSERVE programmatic interface provide TMF commands for operating on resource managers, system managers issuing these commands do so in the context of issues involving multiple software subsystems and inter-platform considerations. Thus, although this manual describes resource-manager commands and touches briefly on related terminology in the following paragraphs, complete discussion of the context in which these commands are used lies beyond the scope of the TMF manual library.

For more information about heterogeneous transaction processing, see the Open Group TRANSACTION PROCESSING Publications, available from X/Open Publications at the following Web location:

<http://www.opengroup.org/pubs/catalog/tp.htm>

These publications can also be ordered through many bookstores.

---

Resource managers encode the transactional semantics of foreign transaction management systems, or provide abstractions of foreign database management systems on the NonStop server platform. They run in a process environment called a gateway process. A gateway process executes a resource manager routine to export a transaction branch from TMF to a foreign transaction management system, import a transaction branch from a foreign transaction management system to TMF, and participate in the agreement protocol between the two transaction management systems.

A resource manager that does not need to be recovered after a failure is called a volatile resource manager. A volatile resource manager manages volatile elements: objects in main memory, user processes maintaining transactional variables, servers maintaining context during transaction execution, and so forth.

A resource manager that requires recovery after a failure is called a recoverable resource manager. A recoverable resource manager, unlike a volatile resource manager, must be registered with TMF in the resource manager directory before it can

be used by a gateway process. This directory is maintained by the TMP. Registration creates an entry for the resource manager in the directory.

Associated with each resource manager is a resource manager file. When a gateway process asks to open a recoverable resource manager, the TMP verifies that the resource manager is registered and authenticates the resource manager's authorization to open this file. Gateway processes that open recoverable resource managers must also close them explicitly.

If the gateway process fails while accessing the recoverable resource manager, the resource manager is closed implicitly.

If active transactions exist when the recoverable resource manager is closed, those transactions are aborted.

If in-doubt transactions exist when the recoverable resource manager is closed, they remain in-doubt and the resource manager is considered to have crashed.

If a recoverable resource manager is in the crashed state when it is reopened, the in-doubt transactions are passed to the new opening process for resolution.

Entries can be added to the resource manager directory either by a gateway process that registers the resource manager name programmatically, or by a user who enters the ADD RESOURCEMANAGER command.

There is a configurable limit on the total number of resource managers allowed in the system. You can adjust this limit through the RECRMOUNT option of the [ALTER BEGINTRANS](#) command. You can display the current limit with the [INFO BEGINTRANS](#) command.

There is also a limit on the total number of resource managers, both recoverable and volatile, that can be open concurrently in a processor by gateway processes. You can alter this limit through the RMOPENPERCPU option of the [ALTER BEGINTRANS](#) command. You can display the current limit with the [INFO BEGINTRANS](#) command.

## Examples

The following ADD RESOURCEMANAGER command adds a resource manager named TUX-A1\_UNIX@SUN to the resource manager directory:

```
TMF 42> ADD RESOURCEMANAGER TUX-A1_UNIX@SUN, OWNER SUPER.TUXEDO
```

The next command adds a resource manager named TUX-A2\_NT@MICROSOFT to the directory:

```
TMF 43> ADD RM TUX-A2_NT@MICROSOFT, OWNER (255,27)
```



# ADD TAPEMEDIA

The ADD TAPEMEDIA command adds one or more tape volumes to the TMF catalog and, optionally, labels these volumes.

```
ADD TAPEMEDIA { tape-volume
                { (tape-volume [, tape-volume] ...) }

                { tape-labeling-options }
                { tape-status-options }
```

where *tape-labeling-options* are:

```
,LABEL ON [!]  
  
,DRIVE [node.]device-name  
  
[,DENSITY density ]  
  
[,UNLOAD {ON | OFF}]
```

and *tape-status-options* are:

```
,LABEL OFF  
  
,STATUS { ASSIGNED }  
         { BAD }  
         { RELEASED }  
         { SCRATCH }
```

## *tape-volume*

identifies the tape volume to be added, specifying the name magnetically encoded on that volume. Names for tape volumes begin with either a digit or a letter. If the name begins with a digit, up to five digits can follow. If the name begins with a letter, up to five digits or letters in any combination can follow. Letter case is not significant; for example, TMF001 and tmf001 are equivalent entries. You cannot specify the name of a tape volume that already exists in the catalog.

## *tape-labeling options*

specifies the tape-labeling options, as follows:

LABEL ON [!]

labels the tape volume on the tape drive specified. If you choose this option, you must also specify the name of the tape drive in the DRIVE parameter. TMF invokes labeled-tape processing for the specified tape volumes to determine whether each one has been labeled successfully.

The exclamation point [!] is optional. It specifies that the tape volume is to be labeled whether or not it already has a label. If you omit the exclamation point

and the volume has already been labeled, the ADD TAPEMEDIA command fails.

DRIVE [*node.*]*device-name*

identifies the tape drive on which you want to label tape volumes. Guardian labeled-tape processing must be enabled on the node (system) you specify in *node*. This parameter is required if you want to label tapes.

*node* is the name of the local or remote node at which the tape drive is connected. The syntax rules for specifying a node name appear under [Disk File Names](#), earlier in this section. If the node is not specified, the default is the node on which your TMFSERVE process is running.

*device-name* is the name of the tape drive.

DENSITY *density*

specifies the density at which tapes are written. See [Section 4, TMFCOM Tape Media Considerations](#), for the DENSITY settings appropriate for the kind of tape drive you are using.

UNLOAD {ON | OFF}

determines whether the tape volume is unloaded after it is labeled. ON unloads the tape volume; OFF suppresses unloading. If the UNLOAD option is completely omitted, the default is ON.

*tape-status-options*

specifies labeling suppression or tape status, as follows:

LABEL OFF

suppresses tape volume labeling. When recovering the catalog, use this option to add tape volumes back into it.

STATUS

describes the status of the tape volume:

ASSIGNED

indicates that the tape volume contains valid dumps. If the catalog already contains an entry for the tape volume specified, an error message is returned.

**BAD**

indicates that the tape volume is defective and cannot be used later for dumping files or for file recovery.

**RELEASED**

indicates that the contents of the tape volume are no longer required for file recovery, and that the tape is not available for re-use in dumping or for file recovery.

**SCRATCH**

indicates that the tape volume can be used for dumping files.

## Security Restrictions

You can issue the ADD TAPEMEDIA command only if you are a member of the super user group.

## TMF State Requirement

Before you can enter the ADD TAPEMEDIA command, TMF must be started.

## Usage Guidelines

For each tape volume known to TMF, an entry describing that volume must be included in the TMF catalog. Each entry contains the name of the tape volume and its current status. To define a new set of tape volumes, you add entries for these volumes by issuing a series of ADD TAPEMEDIA commands.

In most cases, you initially define a tape volume with scratch status, which means that it is available for dumping. The volume automatically moves to assigned status when it receives an audit dump or online dump. The volume is automatically returned to scratch status (or to released status, if you specified RELEASED ON in an [ALTER CATALOG](#) command) when the TMF catalog process determines that none of the dumps stored on it are needed for file recovery.

If an attempt to label a tape volume fails, the catalog is not updated and the configuration is unchanged.

If you issue an ADD TAPEMEDIA command through a TMFCOM process that uses a remote home terminal for displays from TMFSERVE, the command fails and TMFCOM displays Error Message 266. Enter this command only through a TMFCOM process that uses a local home terminal.

Each name you assign to a TMF tape volume must be unique among all TMF tape-volume names defined on all nodes at your site. It is up to you to ensure this uniqueness; TMF does no cross-node checking on your behalf.

For more information about tape media, dumps directed to them, and catalog entries describing them, see the *TMF Operations and Recovery Guide*.

## Example

The following ADD TAPEMEDIA command labels a tape volume with a new name on the tape drive \$TAPE, and adds this name (TT04) to the catalog:

```
TMF 51> ADD TAPEMEDIA TT04, LABEL ON!, DRIVE $TAPE
```

It is often convenient to add multiple tapes by way of a single command, as this example illustrates:

```
TMF 52> ADD TAPEMEDIA (TMF001, TMF002, TMF003, TMF004, &  
>>>TMF005, TMF006, TMF007, TMF008, TMF009, TMF010), &  
>>>LABEL ON!, DRIVE $TAPE1
```

To add a tape on a remote system (perhaps to dump to that tape), you must qualify the tape drive name with the remote system name in the DRIVE parameter of the ADD TAPEMEDIA command, as shown in the next example:

```
TMF 53> ADD TAPEMEDIA (TMF001, TMF002, TMF003, TMF004, &  
>>>TMF005, TMF006, TMF007, TMF008, TMF009), LABEL ON! &  
>>>DRIVE \REMOTE.$TAPE1
```

# ALTER AUDITDUMP

The ALTER AUDITDUMP command changes the configured attribute values of an audit dump process.

```
ALTER AUDITDUMP {MASTER | MAT }
                  {AUX[ILINARY]nn}

    [,BLOCKSIZE {4 | 8 | 12 | 16 | 20 | 24 | 28 | 32 | 36
                  | 40 | 44 | 48 | 52}]
    [,RESET BLOCKSIZE

    [,COPIES 1 [PARALLEL | SERIES]]
    [,COPIES 2 [PARALLEL | SERIES]]
    [,COPIES 3 SERIES
    [,RESET COPIES

    [,SYSTEM {node
              {(node, node)}]
    [,RESET SYSTEM

    [,VERIFYTAPE {OFF | ON}]

    [,UNLOAD {ON | OFF

    [, [MEDIUM {TAPE}
              {DISK}]
    [,RESET MEDIUM]]

    [,ADDDISKMEDIA {disk
                   {(disk [,disk]...)}}]

    [,DELDISKMEDIA {disk
                   {(disk [,disk]...)}}
```

MASTER | MAT  
AUX[ILINARY]*nn*

specifies the audit trail associated with the audit dump process whose attributes are to be altered, thus identifying that process. MASTER or MAT specifies the master audit trail. AUXILIARY*nn* or AUX*nn* specifies an auxiliary audit trail. *nn* is a two-digit number ranging from 01 to 15 that specifies one of the 15 potential auxiliary audit trails in the configuration. (When the value of *nn* is 09 or less, you must specify the leading zero.)

The specified audit trail must be configured with the AUDITDUMP ON option. (For more information about that option, see the ADD AUDITTRAIL command description.)

```
BLOCKSIZE { 4 | 8 | 12 | 16 | 20 | 24 | 28 | 32 | 36
            | 40 | 44 | 48 | 52 }
```

for dumps to tape, sets the tape data block length when multiplied by 1024 bytes. Using large data block lengths is more efficient, but length is limited by the density in bits per inch (bpi) at which the tape drive writes tapes. The following are recommended values:

|               |     |      |      |
|---------------|-----|------|------|
| Density (bpi) | 800 | 1600 | 6250 |
| BLOCKSIZE     | 4   | 8    | 28   |

The originally assigned default is 28.

---

**Note.** For this and other optional parameters, the **originally assigned default** is the value initially assigned to the parameter by TMF. The originally assigned default is used for the parameter when no one has explicitly specified or altered this parameter's value in a previous command. For more information about how defaults are assigned, see [Usage Guidelines](#).

---

If you specify a BLOCKSIZE value larger than 28, but the tape drive selected cannot handle data blocks this long, TMF uses a block size of 28 KB for the dump and generates a warning message.

RESET BLOCKSIZE

for dumps to tape, resets the value for the tape data block size to the originally assigned default, which is 28.

```
COPIES 1 [ PARALLEL | SERIES ]
COPIES 2 [ PARALLEL | SERIES ]
COPIES 3 SERIES
```

for dumps to tape, specifies the number of copies made of each audit dump. These copies can be produced in parallel or series. Specifying multiple copies is recommended so that file recovery is still possible if one of the copies cannot be read.

PARALLEL

produces copies of the audit dump simultaneously on two tape drives. To request this option, you must have two tape drives available. If you specify two node names in the SYSTEM parameter, TMF uses one tape drive on each node.

SERIES

produces copies of the audit dump, one at a time. Select this option if you want to make multiple copies of audit dumps using a single tape drive. TMF reads the audit-trail file on disk once for each serial copy produced; thus, this option can adversely affect the performance of a busy system.

If the audit dump process fails while making serial copies, completed copies are kept in the catalog. When the audit dump process resumes, the number of completed copies is subtracted from the number of copies specified to determine the number of copies remaining to be made.

The originally assigned default is COPIES 1 PARALLEL, which produces one copy, starting with the next tape dump operation.

#### RESET COPIES

for dumps to tape, resets the COPIES setting to the originally assigned default, COPIES 1 PARALLEL, which produces one copy, starting with the next tape dump operation.

#### SYSTEM

for dumps to tape, specifies the node (system) on which all tape copies are made. If you specify a remote node, Guardian labeled-tape processing must be enabled on that node and must be running on the NonStop operating system. TMF, however, need not be configured or running on the remote node. The syntax rules for specifying a node name appear under [Disk File Names](#), earlier in this section.

Specifying two nodes produces parallel copies of tapes, one copy per node. Whether they are produced locally or remotely, the tapes used are registered in the local TMF catalog.

The originally assigned default is the local node.

#### RESET SYSTEM

for dumps to tape, resets the node on which tape copies are made to the originally assigned default, which is the local node.

#### VERIFYTAPE

for dumps to tape, determines whether tape verification is performed after a tape is written.

##### OFF

suppresses verification of tape volumes.

##### ON

performs tape verification. After each tape volume has been written, it is rewound, read completely, and rewound again, and unloaded. The tape-record checksums, tape-record sequence numbers, and tape format are verified. The checks are the same as those made during file recovery; however, tape verification does not include comparing the tape data to the audit-trail files on disk.

If the UNLOAD option is OFF, the tape is not unloaded after it is verified.

If tape verification fails, an error message appears, the audit dump terminates, and the audit dump copy is marked invalid in the catalog. Tape verification approximately doubles the time needed for an audit dump.

The originally assigned default is VERIFYTAPE OFF.

#### UNLOAD

for dumps to tape, determines whether the tape is unloaded after it is written.

##### ON

unloads the tape after the dump is written and the tape is rewound.

##### OFF

suppresses unloading, so that the tape is rewound but not dismounted (unloaded).

The originally assigned default is UNLOAD ON.

#### MEDIUM

determines the medium on which the audit dump is recorded:

##### TAPE

indicates a tape drive.

##### DISK

indicates a disk volume.

If ADDDISKMEDIA or DELDISKMEDIA is specified, MEDIUM DISK must also be specified.

The originally assigned default is TAPE.

#### RESET MEDIUM

deletes all disk media previously configured through the ADDDISKMEDIA option, clears the list of disks established by this option for dumping to disk, and resets the dump medium to the originally assigned default, which is dump to tape.

#### ADDDISKMEDIA

adds to the audit dump process configuration individual disk volumes that can serve as the destination for audit dumps. At any particular time, up to 16, each specified in *disk*, can be configured for the process; an attempt to add a seventeenth, however, results in an error.



The format for *disk* is as follows:

- For disks on local nodes: *[node.]volume[.prefix]*
- For disks on remote nodes: *node.volume.prefix*

*prefix* is two alphabetic characters, to which TMF appends a six-digit integer to name the audit dump file.

If you specify a list of disks, TMF tries to send the audit dump to the first disk listed. If that disk cannot be accessed or is out of space, TMF tries the next disk, and so forth.

If you specify a disk that does not exist, TMF does not report this error now; however, it does report the error when the next dump is attempted.

The optional and required syntax items for *disk* and the naming conventions for *prefix* depend on whether the specified disk is on a remote or local node.

When the disk resides on a remote node:

- The name specified for *disk* must be fully qualified.
- *volume* must not exceed seven characters, counting the dollar sign (\$).
- You cannot specify ZT as the prefix.

When the disk resides on a local node:

- *node* and *prefix* are optional.
- If *node* is not specified, the node on which your TMFSERVE process is running is assigned. *prefix* can have any value.
- If *prefix* is not specified, the local node default ZT is automatically assigned as the prefix.

## DELDISKMEDIA

deletes individual disk volumes from the configuration. The optional and required syntax items and the naming conventions for the prefix are the same as those for ADDDISKMEDIA.

---

**Note.** The following ALTER AUDITDUMP parameters are only valid when the MEDIUM parameter is set to TAPE or is omitted altogether:

- BLOCKSIZE
- RESET BLOCKSIZE
- COPIES
- RESET COPIES
- SYSTEM
- RESET SYSTEM
- VERIFYTAPE
- UNLOAD

The ADDDISKMEDIA and DELDISKMEDIA parameters are only valid when MEDIUM is DISK.

---

## Security Restrictions

You can issue the ALTER AUDITDUMP command only if you are a member of the super user group.

## TMF State Requirement

You can enter the ALTER AUDITDUMP command at any time, whether or not TMF has been started.

## Usage Guidelines

The ALTER AUDITDUMP command is only used to change the attribute values of the audit dump process, initially set to default values if the AUDITDUMP ON option was specified in the ADD AUDITTRAIL command. Whether or not audit trails are actually dumped is determined by the setting of the AUDITDUMP option in the ADD AUDITTRAIL command, and by any changes made to the audit dump process attributes in the ALTER AUDITTRAIL, SET AUDITTRAIL, or RESET AUDITTRAIL commands.

The attributes you change take effect when the next audit dump process starts.

Attributes that you do not specify remain unchanged. So, if you omit an optional parameter, one of the following actions takes place:

- If one or more previous TMFCOM commands (such as ALTER AUDITDUMP or ADD AUDITTRAIL) specified the parameter, the value entered in the last of these commands remains assigned. For example, if the last ALTER AUDITDUMP command assigned the node \NYORK for the SYSTEM parameter, the current ALTER AUDITDUMP command would use the node \NYORK.
- If no previous TMFCOM command specified the parameter, the originally assigned default remains assigned. For example, if no earlier ALTER AUDITDUMP command specified the node in the SYSTEM parameter, the current ALTER AUDITDUMP command would use the local node.

If you attempt any of the following actions, an error occurs:

- Adding a disk media attribute that already exists.
- Deleting a disk media attribute that is not currently configured.
- Specifying the same disk media name for the same option (ADDDISKMEDIA or DELDISKMEDIA) more than once in the same command. For example, you cannot specify the same disk volume more than once in the ADDDISKMEDIA attribute, nor more than once in the DELDISKMEDIA attribute. However, you can use DELDISKMEDIA to delete a disk and then use ADDDISKMEDIA in the same command to add that same disk back into the configuration; this capability allows you, for example, to respecify the order in which disks are used.

If you specify ADDDISKMEDIA or DELDISKMEDIA, you must also specify MEDIUM DISK.

If you specify MEDIUM DISK:

- You cannot specify any attributes that apply to tape-based dumps, namely: BLOCKSIZE, COPIES, SYSTEM, VERIFYTAPE, and UNLOAD.
- There must be a minimum of 1 and a maximum of 16 ADDDISKMEDIA or DELDISKMEDIA attributes configured when the command completes.

If the audit dump configuration was set to MEDIUM DISK and the ALTER AUDITDUMP command specifies MEDIUM TAPE and is successful, all previously configured disk media attributes are deleted from the configuration.

Specifying two copies of an audit dump is recommended for dumping to tape. By using the SYSTEM (*node*, *node*) and COPIES 2 PARALLEL parameter constructs, you can produce parallel dumps on two nodes (one dump per node). Both nodes can be remote nodes, or one can be a remote node and the other your local node. All reels of a specific copy (Copy 1 or Copy 2), however, must be made on the same node.

Specifying two nodes in the SYSTEM parameter produces the dump on parallel copies of tapes, one copy per node. If TMF later tries to restore the dump, TMF always requests the dump tape to be mounted on any available drive on the first node specified in the SYSTEM parameter.

## Examples

The following command causes two parallel copies of each future audit dump for the master audit trail to be directed to magnetic tape drives on the remote system \LA01, changes the blocksize to 16, and requests that tape verification be performed:

```
TMF 11> ALTER AUDITDUMP MAT, SYSTEM \LA01, COPIES 2 PARALLEL, &
>>>BLOCKSIZE 16, VERIFYTAPE ON
```

The next command specifies that two parallel copies of each audit dump for the master audit trail be directed to two magnetic tape drives, one on the system \GOTHAM and the other on the system \NEWARK:

```
TMF 12> ALTER AUDITDUMP MAT, SYSTEM (\GOTHAM, \NEWARK), &
>>>COPIES 2 PARALLEL
```

The following command causes future audit dumps for the master audit trail to be directed first to the disk volume \$ADUMP1, and then to the disk volume \$ADUMP2 (when \$ADUMP1 is full). (This command assumes that the disk media list is empty.)

```
TMF 13> ALTER AUDITDUMP MAT, MEDIUM DISK, &
>>>ADDDISKMEDIA ($ADUMP1, $ADUMP2)
```

The next command directs all future audit dumps for the master audit trail first to the disk volume \$BACKUP, then to the disk volume \$SET (when \$BACKUP is full), and finally to the remote disk volume \NEWYORK.\$LAST (when both \$BACKUP and \$SET are full). (This command assumes that the disk media list is empty.)

```
TMF 14> ALTER AUDITDUMP MAT, MEDIUM DISK, &
>>>ADDDISKMEDIA ($BACKUP, $SET, &
>>>\NEWYORK.$LAST)
```

The following command removes the disk volume \$ADUMP3 from, and adds \$ADUMP4 to, the disk media list for the auxiliary audit trail AUX01:

```
TMF 15> ALTER AUDITDUMP AUX01, MEDIUM DISK, &
>>>ADDDISKMEDIA $ADUMP4, &
>>>DELDISKMEDIA $ADUMP3
```

The next command resets the AUX01 audit dump to the default, which is dump to tape, and clears the disk media list:

```
TMF 16> ALTER AUDITDUMP AUX01, RESET MEDIUM
```

# ALTER AUDITTRAIL

The ALTER AUDITTRAIL command changes the configured attribute values of an audit trail and can execute while TMF is running.

```
ALTER AUDITTRAIL {MASTER | MAT}
                  {AUX[ILINARY]nn}

[,AUDITDUMP {ON | OFF}]

[,ADDACTIVEVOL[S] {volume
                  {(volume [,volume]...) }
                  }]
[,DELETEACTIVEVOL[S] {volume
                     {(volume [,volume]...) }
                     }]

[,FILESPERVOLUME integer]
[,RESET FILESPERVOLUME ]

[,ADDOVERFLOWVOL[S] {volume
                    {(volume [,volume]...) }
                    }]
[,DELETEOVERFLOWVOL[S] {volume
                       {(volume [,volume]...) }
                       }]

[,OVERFLOWTHRESHOLD integer]
[,RESET OVERFLOWTHRESHOLD ]

[,ADDRESTOREVOL[S] {volume
                   {(volume [,volume]...) }
                   }]
[,DELETERESTOREVOL[S] {volume
                      {(volume [,volume]...) }
                      }]

[,MAXRETAINEDATFILES integer]
[,RESET MAXRETAINEDATFILES ]

[,BEGINTRANSDISABLE integer]
[,RESET BEGINTRANSDISABLE ]
```

MASTER | MAT  
AUX[ILINARY]nn

identifies the audit trail whose attributes are to be altered. MASTER or MAT specifies the master audit trail. AUXILIARY<sub>nn</sub> or AUX<sub>nn</sub> specifies an auxiliary audit trail. *nn* is a two-digit number ranging from 01 to 15 that specifies one of the 15 potential auxiliary audit trails in the configuration. (When the value of *nn* is 09 or less, you must specify the leading zero.)

AUDITDUMP {ON | OFF}

determines whether audit dumping is performed for the audit trail. ON specifies that audit dumping takes place, and OFF prevents audit dumping.

Unless you also issue an [ALTER AUDITDUMP](#) command, the attributes of the audit dump process default to the following values:

BLOCKSIZE, the tape data block length: 28 (x 1024 bytes)

COPIES, the number of copies to be made in series or in parallel of each dump: 1 parallel

SYSTEM, the node (system) on which tape copies are made: the local node

VERIFYTAPE, the tape-verification selection: OFF

MEDIUM, the medium on which the dump is recorded: TAPE

The originally assigned default is ON.

---

**Note.** For this and other optional parameters, the **originally assigned default** is the value initially assigned to the parameter by TMF. The originally assigned default is used for the parameter when no one has explicitly specified or altered this parameter's value in a previous command. For more information about how defaults are assigned, see [Usage Guidelines](#).

---

If you specify AUDITDUMP OFF, commands related to AUDITDUMP that are executed during the session return a message that the audit trail is not configured for dumping. You can specify AUDITDUMP OFF only while TMF is started or prior to its first start operation for the current configuration.

If you omit the AUDITDUMP option, the AUDITDUMP attribute value remains unchanged.

ADDACTIVEVOL[ S ]  
DELETEACTIVEVOL[ S ]

adds or deletes active-audit volumes for holding audit-trail files. The total number of active-audit volumes configured for an audit trail must be between 1 and 16, inclusive. After TMF is configured and started for the first time, volumes can be added or deleted only while TMF is started.

---

**Note.** TMF is shipped from the factory with the \$AUDIT disk volume configured as the active-audit volume. You are free to retain \$AUDIT for this purpose, replace \$AUDIT with another volume as the recipient of audit trails, or add other active-audit volumes to the configuration. See also the note about the preconfiguration of data volumes that appears under the discussion of the [ALTER DATAVOLS](#) command.

---

Adding an active-audit volume adds the number of files specified in the FILESPERVOLUME attribute to the audit trail capacity. Before it can successfully add a volume, TMF must preallocate to the volume the number of files specified in FILESPERVOLUME. If the volume has insufficient space for these files, it cannot be added to the configuration.

Active-audit volumes are added to the end of the list of available volumes maintained by TMF; therefore, they might not be used immediately. Audit-trail files are written to active-audit volumes sequentially, but TMF tries to avoid writing two

audit-trail files in succession to the same volume. When TMF is ready to write to an active-audit volume, it uses the next preallocated file that resides on that volume. Eventually, all preallocated files on the volume are used.

If you specify `DELETEACTIVEVOL` for an active-audit volume that holds files that are needed by TMF, the volume is not immediately removed from the configuration. Instead, the volume is marked as “deleting”; any preallocated files stored on it are purged immediately, and new files are not written to it. When no more audit-trail files reside on the audit volume, it is removed from the configuration. However, you can re-add an audit volume marked as “deleting” to the configuration without waiting for the deletion to complete.

The audit volume that holds the current audit-trail file cannot be deleted. If you want to delete that volume, issue a [NEXT AUDITTRAIL](#) command before you request the deletion. The [NEXT AUDITTRAIL](#) command redirects the current audit-trail file to the next volume available, so that this next volume becomes the new destination of the audit information.

In addition, the volume that holds the audit-trail file assigned by the [NEXT AUDITTRAIL](#) command cannot be deleted if that file has been prepared to receive audit records. If you want to delete this volume, you must enter another [NEXT AUDITTRAIL](#) command and redirect the audit information to still another file on another volume.

Deleting a volume from a configuration reduces the capacity of the active-audit trail by the number of files specified in the `FILESPERVOLUME` attribute.

`FILESPERVOLUME` *integer*

specifies the number of audit-trail files that reside on each active-audit volume configured. The minimum value is 1, and the maximum is 100. The originally assigned default is 5.

If the number of files per volume is increased, then extra preallocated files are created immediately on each active-audit volume to make up the difference between the old and new values. If this allocation fails, then the `ALTER AUDITTRAIL` command returns an error and the `FILESPERVOLUME` value stays the same as before.

If the number of files is decreased, the action taken varies. Any preallocated files on the active-audit volumes are purged up to the point where each active volume contains the new number of files per volume. If an active-audit volume still contains more files than the new `FILESPERVOLUME` value allows, TMF purges them (rather than renaming them) when they are no longer needed.

Increasing the `FILESPERVOLUME` value is the recommended way of increasing the audit trail capacity, if there is sufficient space on the currently configured active-audit volumes. Using this technique evenly distributes the new files among all volumes, rather than concentrating them on any added volumes.

**RESET FILESPERVOLUME**

resets the FILESPERVOLUME attribute to the originally assigned default, which is 5.

**ADDOVERFLOWVOL[ S ]****DELETEOVERFLOWVOL[ S ]**

adds or deletes overflow-audit volumes.

If you specify DELETEOVERFLOWVOL for a volume that contains an overflow-audit file, the volume is marked as “deleting” until the overflow-audit file is no longer needed.

**OVERFLOWTHRESHOLD *integer***

specifies the percentage of audit trail capacity in use above which TMF begins overflow copying. This is an optional parameter. The minimum value is 50, and the maximum is 100. The originally assigned default is 80(%).

**RESET OVERFLOWTHRESHOLD**

resets the OVERFLOWTHRESHOLD attribute to the originally assigned default, which is 80(%).

**ADDRESTOREVOL[ S ]****DELETERESTOREVOL[ S ]**

adds or deletes restore-audit volumes used during execution of a RECOVER FILES command. At least one restore-audit volume must be configured for each audit trail that has audit dumping configured. You can configure a maximum of 16 restore-audit volumes.

If you specify DELETERESTOREVOL for a volume that is being used as a restore location, the restore operation continues until it successfully completes and the restore-audit volume is marked as “deleting” until the volume is no longer needed for this operation.

If AUDITDUMP ON is specified, or if AUDITDUMP is already ON and has not been altered, you cannot delete the last restore-audit volume configured.

**MAXRETAINEDATFILES *integer***

specifies the maximum number of restored audit-trail files to retain on disk in restore-audit volumes for the specified audit trail after the restored audit-trail files are no longer needed by a recovery process. This option applies to files restored from tape only. The minimum value is 1, and the maximum is 9999. The originally assigned default is 0 (no files are retained).

**RESET MAXRETAINEDATFILES**

resets the MAXRETAINEDATFILES attribute to the originally assigned default, which is 0.



`BEGINTRANSDISABLE integer`

specifies the percentage of audit trail capacity that can be consumed before TMF disables new transactions. When the consumed capacity exceeds this value, new transactions are not allowed. When the consumed capacity falls back below this threshold, new transactions are no longer disabled by audit-trail usage (however, new transactions can be disabled for other reasons, as explained in the [ALTER BEGINTRANS](#) command description).

The minimum value is 50, and the maximum is 100. The originally assigned default is 90(%).

`RESET BEGINTRANSDISABLE`

resets the BEGINTRANSDISABLE attribute to the originally assigned default, which is 90(%).

## Security Restrictions

You can issue the ALTER AUDITTRAIL command only if you are a member of the super user group.

## TMF State Requirement

You can enter the ALTER AUDITTRAIL command at any time, whether or not TMF has been started. However, after TMF is configured and started for the first time, you can add or delete active-audit volumes or alter AUDITDUMP to OFF only while TMF is up and running.

## Usage Guidelines

Before any auxiliary audit trail can be dumped, you must configure the master audit trail (MAT) for this purpose.

---

**Note.** For both online dumps and audit dumps, Expand links to remote disk volumes must support 56 KB input/output.

---

Attributes that you do not specify remain unchanged. So, if you omit an optional parameter, one of the following actions takes place:

- If one or more previous TMFCOM commands (such as [ADD AUDITTRAIL](#) or [ALTER AUDITTRAIL](#)) specified the parameter, the value entered in the last of these commands remains assigned.
- If no previous TMFCOM command specified the parameter, the originally assigned default remains assigned.

## AUDITDUMP Option Considerations

If you set the AUDITDUMP attribute to ON after TMF has been running with AUDITDUMP OFF, the first file dumped is the file to which audit information is being written (the current file). For example, if audit information is being written to file sequence number 12 when AUDITDUMP ON is specified, the first file to be dumped will be sequence number 12.

Changing AUDITDUMP from ON to OFF may impact file recovery: after an undumped audit-trail file is renamed or purged, file recovery cannot be performed through that file. For example, if file sequence number 23 is not yet dumped when AUDITDUMP is set to OFF, then, after file 23 is renamed or purged, file recovery cannot recover files beyond audit-trail file number 22 (however, file recovery to a timestamp before the end of file number 22 is still possible).

To set AUDITDUMP to OFF for the master audit trail, you must also set it to OFF for all auxiliary audit trails in the configuration. To set AUDITDUMP to OFF for any audit trail, you must also set RECOVERYMODE to ONLINE for all data volumes that use this audit trail.

When you issue an ALTER AUDITTRAIL command that sets AUDITDUMP to OFF, TMF displays a confirmation message that allows you to either verify that you want to take this action or to indicate that you do not.

## MAXRETAINEDATFILES Option Considerations

During volume recovery or file recovery, if TMF requires an audit-trail file that is no longer available on the system but is archived on tape, TMF restores that file from an audit dump on the tape to a restore-audit volume on disk. When the recovery process completes, TMF purges this file and all other unneeded files from the restore-audit volume. When TMF is processing multiple recovery operations, whether serially or in parallel, a recovery process often must wait for TMF to restore the audit-trail file repeatedly to support each recovery operation.

To make TMF keep audit-trail files restored from tape on the restore-audit volume indefinitely, enter the ALTER AUDITTRAIL command with the MAXRETAINEDATFILES option. This action overrides the standard restore-and-purge mechanism and enhances performance by:

- Eliminating or minimizing the time a recovery process must wait for TMF to restore an audit-trail file.
- Minimizing disk contention for audit-trail files being read from, or restored to, the same disk.

---

**Note.** For audit-trail files dumped to disk, TMF opens the disk-dump copy directly, rather than copying it to a restore-audit volume. Therefore, for audit-trail files on disk, the entire disk-restore time is eliminated and all recovery processing (even the first) need not wait for this restore operation. Therefore, you do not specify MAXRETAINEDATFILES for audit-trail files on disk.

---

When you expect to conduct many recovery operations at or near the same time, and you wish to keep as many potentially required files on disk as possible, use the MAXRETAINEDATFILES option. When the recovery load diminishes, use the RESET MAXRETAINEDATFILES option to return to the standard restore-and-purge mechanism for tape files, freeing disk space for other purposes. Remember: MAXRETAINEDATFILES optimizes access time for files archived on tape; RESET MAXRETAINEDATFILES maximizes available disk space.

To specify the amount of disk space available for retaining audit-trail files restored from tape, use the MAXRETAINEDATFILES option to indicate this amount in terms of the number of files. If you enter MAXRETAINEDATFILES while TMF is running, this option takes effect immediately. If you enter MAXRETAINEDATFILES while TMF is stopped, MAXRETAINEDATFILES takes effect when TMF restarts. The MAXRETAINEDATFILES *integer* value specifies the number of audit-trail files to be retained. This value applies only to the audit trail specified in the current ALTER AUDITTRAIL command; TMF manages each audit trail individually with regard to its MAXRETAINEDATFILES attribute.

The *integer* value specifies the total number of audit-trail files retained across the entire set of restore-audit volumes. In other words, whether you have one restore-audit volume or five restore-audit volumes, setting the value to 50 would give you a maximum of 50 retained audit-trail files in either case.

The *integer* value does not affect the restored audit-trail files that are currently in use by a recovery process, only what happens to those files when they are no longer in use.

If you decrease *integer* from its current value, and the present number of retained audit-trail files exceeds the new limit, TMF immediately purges the lowest-numbered retained audit-trail files from disk until the new limit is reached. (For example, TMF purges the files named AA000032 and AA000033 before AA000034.) If you increase *integer* from its current value, TMF allocates no new space but does retain more audit-trail files from this moment on, until the currently allocated space runs out.

If you set the *integer* value too high, attempts by a recovery process to restore required audit-trail files may fail with File System Error 43. In this case, you can correct the problem without restarting the recovery operation by taking any of these actions:

- Lower the MAXRETAINEDATFILES value.
- Free more restore-audit-volume disk space.
- Add more restore-audit volumes to the TMF configuration.

Before using the MAXRETAINEDATFILES option, be sure that enough space actually exists on the restore-audit volumes to contain the restored audit dumps. This precaution is particularly important where restore-audit volumes overlap between configured audit trails.

MAXRETAINEDATFILES does not guarantee that an audit-trail file will be restored from tape only once. Space requirements and the impact of other recoveries all play a part in determining which audit-trails files are retained indefinitely.

The current MAXRETAINEDATFILES value remains in effect until you explicitly change it by entering a new value or using RESET MAXRETAINEDATFILES, or until you request a DELETE TMF operation:

- If you enter an ALTER AUDITTRAIL command without the MAXRETAINEDATFILES option, the current MAXRETAINEDATFILES value is unchanged.
- If you never enter an ALTER AUDITTRAIL command with the MAXRETAINEDATFILES option, or if you enter this command with RESET MAXRETAINEDATFILES specified, the MAXRETAINEDATFILES attribute is set to its default value of 0.

If you specify RESET MAXRETAINEDATFILES, any retained audit-trail files presently on disk are purged immediately after they are no longer needed.

---

**Note.** You cannot use the ADD AUDITTRAIL command to set the MAXRETAINEDATFILES value; the MAXRETAINEDATFILES option is only available with ALTER AUDITTRAIL.

Each time TMF is started, all audit-trail files on restore-audit volumes are purged automatically. Thus, any subsequent recovery is always forced to restore any needed audit trails that were dumped to tape.

---

## Other Operational Considerations

The FILESIZE attribute, whose initial value is assigned with the [ADD AUDITTRAIL](#) command, cannot be changed with the ALTER AUDITTRAIL command. To alter this value, you must purge the existing TMF configuration with the [DELETE TMF](#) command and then create an entirely new configuration.

- 
- △ **Caution.** Before you decrease the size of the audit trail, determine whether such a change might interrupt transaction processing. TMF does not alert you to the possibility that the audit trail could exceed its BEGINTRANSDISABLE threshold when you issue an ALTER AUDITTRAIL command to decrease the files per active volume or delete an active volume from the configuration.

If the audit trail exceeds its BEGINTRANSDISABLE threshold, new transactions cannot start. In this case, transaction processing is suspended until audit trail capacity drops below the threshold.

If the OVERFLOW threshold is exceeded as a result of reduced audit trail capacity, the first pinned audit-trail file is copied to the overflow volume.

Be careful not to lower the BEGINTRANSDISABLE or OVERFLOW threshold below your TMF system's current capacity levels; they will engage precisely in accordance with your request.

---

When you specify the ALTER AUDITTRAIL command with DELETEACTIVEVOL for an active-audit volume or DELETEOVERFLOWVOL for an overflow volume, and that

volume holds audit-trail files needed by TMF for backout, volume recovery, or system restart, the volume is marked as “deleting.” Any preallocated files stored on the volume are purged immediately, and new files are not written to the volume. When no more audit-trail files reside on the volume, it is removed from the TMF configuration.

Once a disk volume is defined in the TMF configuration as an active or overflow volume, that volume cannot be taken down through an SCF STOP DISK command until all audit-trail files are purged from it by TMF and the volume is removed from the audit-trail configuration. All volumes defined as active or overflow volumes must be accessible before TMF can be started.

If the primary transaction management process (TMP) fails before the ALTER AUDITTRAIL command completes, the backup TMP takes over and undoes the effects of that command before processing any new requests from TMFCOM. If both TMPs fail before the ALTER AUDITTRAIL command completes, however, the effects of the command are not undone; to get the configuration to the desired state, use the [INFO AUDITTRAIL](#) command to check the configuration and then enter ALTER AUDITTRAIL commands to compensate for the changes needed.

If the ALTER AUDITTRAIL command returns an error, files that were preallocated when active volumes were added might not be purged from the volumes; this occurs if the TMP loses access to these volumes. If you attempt the ALTER AUDITTRAIL command again, adding these volumes as active, the existing preallocated files are purged before the new ones are created.

TMF includes a transaction abort threshold based on master audit trail (MAT) capacity. This threshold dictates that if the oldest pinned MAT file is pinned because of currently active transactions, and 45% or more of the MAT's active capacity is filled, TMF automatically aborts those transactions involved in this pinning. (TMF reports these aborts with EMS event message 289.) The MAT-based threshold cannot be adjusted by any operator command. Instead, if this threshold is aborting transactions that should not be aborted, you must increase the audit trail capacity. You can do this either by increasing the number of audit-trail files per volume or by adding another active-audit volume.

## Examples

The following command changes the FILESPERVOLUME value for the master audit trail to 4:

```
TMF 21> ALTER AUDITTRAIL MAT, FILESPERVOLUME 4
```

The subsequent command adds an active-audit volume and changes the number of files for the volume to six for the AUX02 audit trail:

```
TMF 22> ALTER AUDITTRAIL AUX02, ADDACTIVEVOL $AUDIT2, &  
>>>FILESPERVOLUME 6
```

The next command sets the AUDITDUMP attribute to OFF for the master audit trail:

```
TMF 23> ALTER AUDITTRAIL MAT, AUDITDUMP OFF
```

WARNING: After AUDITDUMP is turned OFF, potentially neither volume nor file recovery can recover your files because audittrail files could be purged.

Please confirm the option, Auditdump OFF? (Y/N) **y**

.  
.  
.

Assume that a restore-audit volume was originally configured on the disk drive \$TMF01, but now needs to be moved to another disk drive, \$TMF02. The following example shows how this can be done with a single ALTER AUDITTRAIL command. This command both creates a restore-audit volume on \$TMF02 and deletes the one on \$TMF01:

```
TMF 24> ALTER AUDITTRAIL MAT, ADDRESTOREVOL $TMF02, &  
>>>DELETERESTOREVOL $TMF01
```

The following ALTER AUDITTRAIL command requests TMF to retain up to the last 50 highest-numbered MAT audit-trail files in the restore-audit volumes, even after all recovery processes are finished:

```
TMF 25> ALTER AUDITTRAIL MAT, MAXRETAINEDATFILES 50
```

The next ALTER AUDITTRAIL command purges from the restore-audit volumes all MAT audit-trail files that are not being used in a recovery operation:

```
TMF 26> ALTER AUDITTRAIL MAT, RESET MAXRETAINEDATFILES
```

# ALTER BEGINTRANS

The ALTER BEGINTRANS command changes options related to the automatic disabling and reenabling of transaction processing, control of the TMF autoabort feature, and limits on resource managers and transaction branches used in heterogeneous transaction processing.

```
ALTER BEGINTRANS

{[,TRANSCOUNTTHRESH (integer1, integer2)]
[,RESET TRANSCOUNTTHRESH ]

[,TMFLIBMEMTHRESH (integer3, integer4) ]
[,RESET TMFLIBMEMTHRESH ]

[,TMPMEMTHRESH (integer5, integer6) ]
[,RESET TMPMEMTHRESH ]

[,TRANSPERCPU (integer7, integer8) ]
[,RESET TRANSPERCPU ]

[,AUTOABORT timeout | 0 ]
[,RESET AUTOABORT ]

[,RECRMOUNT integer9 ]
[,RESET RECRMOUNT ]

[,RMOPENPERCPU integer10 ]
[,RESET RMOPENPERCPU ]

[,BRANCHESPERRM integer11 ]
[,RESET BRANCHESPERRM ] }
```

TRANSCOUNTTHRESH (*integer1*, *integer2*)  
RESET TRANSCOUNTTHRESH

specifies the upper and lower limits at which the number of active transactions in the system automatically disables and reenables new transactions. When the transaction count reaches *integer2*, new transactions are disabled. When the transaction count falls below *integer1*, they are reenabled, if the other conditions needed for reenabling (listed under “Usage Guidelines”) exist. You must specify both values in the command; *integer1* must be 1 or greater, but less than *integer2*. Neither *integer1* nor *integer2* can be greater than 1073741824. To prevent transaction processing from constantly switching back and forth between the enabled and disabled states, a recommended approach is to set the values of *integer1* and *integer2* to result in a large range (such as 100) between the limits.

RESET TRANSCOUNTTHRESH selects the originally assigned default limits, which are 1500 for *integer1* and 1600 for *integer2*.

---

**Note.** For this and other optional parameters, the **originally assigned default** is the value initially assigned to the parameter by TMF. The originally assigned default is used for the parameter when no one has explicitly specified or altered this parameter's value in a previous command. For more information about how defaults are assigned, see [Usage Guidelines](#).

---

TMFLIBMEMTHRESH (*integer3*, *integer4*)  
RESET TMFLIBMEMTHRESH

specifies the upper and lower limits at which the percentage of processor memory allocated to TMFLIB automatically disables or reenables new transactions. When the percentage reaches *integer4*, new transactions are disabled. When the percentage falls below *integer3*, they are automatically reenabled, if the other conditions needed for reenabling exist. You must specify both values in the command, and *integer3* must be less than *integer4*. The minimum value allowed for both *integer3* and *integer4* is 1, and the maximum is 100.

RESET TMFLIBMEMTHRESH selects the originally assigned default percentages, which are 85 for *integer3* and 90 for *integer4*.

TMPMEMTHRESH (*integer5*, *integer6*)  
RESET TMPMEMTHRESH

specifies the upper and lower limits at which the percentage of processor memory allocatable to the TMP's extended segment automatically disables or reenables new transactions. When the percentage reaches *integer6*, new transactions are disabled. When the percentage falls below *integer5*, they are automatically reenabled, if the other conditions needed for reenabling (listed under "Usage Guidelines") exist. Because the TMP's extended segment can grow to about 1 GB, transactions typically are not disabled very quickly. You must specify both values in the command, and *integer5* must be less than *integer6*. The minimum value allowed for both *integer5* and *integer6* is 1, and the maximum is 100.

RESET TMPMEMTHRESH selects the originally assigned default percentages, which are 90 for *integer5* and 95 for *integer6*.

TRANSPERCPU (*integer7*, *integer8*)  
RESET TRANSPERCPU

specifies the number of transactions that can run in TMP and non-TMP processors: *integer7* indicates the number of new transactions that can be started concurrently in processors where the TMP does not run, and *integer8* determines the number that can be started in processors where the TMP runs. The *integer8* count includes any network transactions begun on a remote node that involve the local node. Neither *integer7* nor *integer8* can be less than 1 or greater than 1073741824. Because of implementation dependencies, both *integer7* and *integer8* must be set to a power of 2; if either value entered



does not meet this requirement, it is rounded up to the next higher power of 2. As an example, if you enter 1000 for either of these values, TMF sets the value to 1024; if you enter 2000, it sets the value to 2048.

RESET TRANSPERCPU selects the originally assigned default values, which are 1024 for both *integer7* and *integer8*.

```
AUTOABORT [timeout | 0 ]
RESET AUTOABORT
```

controls the autoabort feature. If you specify *timeout*, all transactions are aborted automatically when they reach the duration indicated by this parameter. Specify *timeout* in the following format:

```
integer { HOUR[S] }
        { MINUTE[S] }
        { SECOND[S] }
```

The minimum value for *integer* is 20 seconds. The maximum value is 21474836 seconds, 357913 minutes, or 5965 hours; these values amount to approximately eight months. For most applications in typical OLTP environments, values close to one minute are probably appropriate.

The entry 0 (zero) turns off the autoabort function for all transactions, eliminating any time limit.

RESET AUTOABORT selects the originally assigned default *timeout* value, which is two hours.

```
RECRMOUNT integer9
RESET RECRMOUNT
```

specifies the maximum number of recoverable resource managers that can be registered for the entire TMF system (spanning a single EXPAND node). The specified value is rounded to the nearest multiple of 16. The minimum value allowed is 16, and the maximum is 16384. The default value is 256.

RESET RECRMOUNT selects the originally assigned default value for *integer9*, which is 256.

```
RMOPENPERCPU integer10
RESET RMOPENPERCPU
```

specifies the maximum number of resource managers, both recoverable and volatile, that can be open currently for each CPU. The specified value is rounded to the nearest multiple of 16. The minimum value allowed is 16, and the maximum is 1024. The default value is 128.

RESET RMOPENPERCPU selects the originally assigned default value for *integer10*, which is 128.

```
BRANCHESPERRM integer11
RESET BRANCHESPERRM
```

specifies the maximum number of transaction branches that can be concurrently imported or exported by each recoverable or volatile resource manager. The specified value is rounded to the nearest multiple of 16. The minimum value allowed is 16, and the maximum is 1024. The default value is 128.

RESET BRANCHESPERRM selects the originally assigned default value for *integer11*, which is 128.

## Security Restrictions

You can issue the ALTER BEGINTRANS command only if you are a member of the super user group.

## TMF State Requirement

Before you can enter the ALTER BEGINTRANS command, TMF must be started.

## Usage Guidelines

If you alter TRANSCOUNTTHRESH, TMFLIBMEMTHRESH, or TMPMEMTHRESH, your changes take effect immediately. If you alter TRANSPERCPU, RECRMOUNT, RMOPENPERCPU, or BRANCHESPERRM, however, your changes do not take effect until the next time a START TMF command is issued.

When the AUTOABORT *timeout* value is changed, new transactions become subject to the new *timeout* value. However, currently active transactions remain subject to the previously set *timeout* value.

- 
- △ **Caution.** TMF aborts any transaction pinning the oldest MAT file if the file is pinned because of currently active transactions and if audit information is filling 45% or more of the MAT capacity. Beyond this, however, when the AUTOABORT feature is off (threshold set to 0), there is no mechanism to abort runaway transactions automatically; if multiple runaway transactions go unnoticed, they can cumulatively pin many audit-trail files, preventing them from being recycled and prohibiting the start of new transactions.
- 

TMF disables and reenables transaction processing by disallowing and allowing new application calls to the BEGINTRANSACTION system library procedure. If transactions are disabled, BEGINTRANSACTION returns File System Error 86.

Attributes that you do not specify remain unchanged. So, if you omit an optional parameter, one of the following actions takes place:

- If one or more previous TMFCOM commands (such as a previous ALTER BEGINTRANS command) specified the parameter, the value entered in the last of these commands remains assigned.

- If no previous TMFCOM command specified the parameter, the originally assigned default remains assigned.

---

**Note.** You can specify multiple parameters in the ALTER BEGINTRANS command. You must specify at least one parameter, or an error results.

---

For transaction processing to be reenabled after it has been disabled because one of the upper limits set by ALTER BEGINTRANS has been reached, all of the following conditions must exist:

- TMF must be started.
- No [DISABLE BEGINTRANS](#) operation must be in progress.
- No audit trail must presently exceed its BEGINTRANSDISABLE threshold.
- All ALTER BEGINTRANS command's TRANSCOUNTTHRESH, TMFLIBMEMTHRESH, and TMPMEMTHRESH parameters whose upper limits have been exceeded must have subsequently fallen below their lower limits.

---

**Note.** The RECRMOUNT, RMOPENPERCPU, and BRANCHESPERRM options apply to resource managers, which support heterogeneous transaction processing. Resource managers are used in products such as NonStop TUXEDO. Although TMFCOM and the TMFSERVE programmatic interface provide TMF commands and options for operating on resource managers, system managers who use them do so in the context of issues involving multiple software subsystems and inter-platform considerations. Complete discussion of heterogeneous transaction processing and the context in which resource-manager commands and options are used lies beyond the scope of the TMF manual library.

For more information about heterogeneous transaction processing, see the Open Group TRANSACTION PROCESSING Publications, available from X/Open Publications at the following Web location:

<http://www.opengroup.org/pubs/catalog/tp.htm>

These publications can also be ordered through many bookstores.

---

In very rare cases, increasing the TRANSPERCPU value could require increasing the TMFMON process' EXTENDEDSEGSIZE value as well. However, in most cases, the default value assigned for the TMFMON EXTENDEDSEGSIZE parameter (12 MB) is more than sufficient.

The value of the TMP process' node disconnect timer, if modified through the SNOOP utility, can impact the use of the AUTOABORT option in the ALTER BEGINTRANS command. The disconnect timer specifies the maximum number of seconds that a distributed transaction is allowed to exist without receiving any replies from a participating node; when this time limit expires, the TMP terminates the transaction. If you use the SNOOP utility to change the disconnect timer value and later issue a TMFCOM ALTER BEGINTRANS command to set the AUTOABORT *timeout* parameter lower than the new disconnect timer value, the ALTER BEGINTRANS command fails and an error message appears. To correct this situation, you must

either use SNOOP to lower the disconnect timer value and then reenter the ALTER BEGINTRANS command with the same AUTOABORT *timeout* value, or reenter the ALTER BEGINTRANS command with an AUTOABORT *timeout* value higher than the present disconnect timer value.

---

**Note.** This problem occurs only if you change the disconnect timer setting from its default value (30 seconds); that is, the AUTOABORT *timeout* value can be set lower than the disconnect timer value if the disconnect timer value is not changed from its default.

---

The SNOOP utility is provided with the TMF product, and is documented in the file named \$SYSTEM.SYS<sub>nn</sub>.SNOOPDOC.

## Examples

The following ALTER BEGINTRANS command directs TMF to disable transaction processing if the transaction count rises above 1750, and to reenable it if the count falls below 1450:

```
TMF 31> ALTER BEGINTRANS, TRANSCOUNTTHRESH (1450,1750)
```

The next ALTER BEGINTRANS command directs TMF to disable transaction processing when the amount of processor memory allocatable to the TMP's extended segment rises above 85%, and to reenable it if the percentage falls below 80%:

```
TMF 32> ALTER BEGINTRANS, TMPMEMTHRESH (80,85)
```

The following ALTER BEGINTRANS command specifies that when the percentage of processor memory allocated to TMFLIB reaches 90%, new transactions are disabled, and when this percentage falls to 85%, new transactions are again enabled:

```
TMF 33> ALTER BEGINTRANS, TMFLIBMEMTHRESH (85,90)
```

The following ALTER BEGINTRANS command requests TMF to abort any transactions that run longer than two minutes:

```
TMF 34> ALTER BEGINTRANS, AUTOABORT 2 MINUTES
```

The next ALTER BEGINTRANS command disables the autoabort feature for all transactions:

```
TMF 35> ALTER BEGINTRANS, AUTOABORT 0
```

The ALTER BEGINTRANS command shown next changes the maximum number of recoverable resource managers that can be registered to 128:

```
TMF 36> ALTER BEGINTRANS, RECRMOUNT 128
```

The final ALTER BEGINTRANS command presented alters the maximum number of open resource managers per processor to 32 and the maximum number of transaction branches per resource manager to 16:

```
TMF 37> ALTER BEGINTRANS, RMOPENPERCPU 32, BRANCHESPERRM 16
```

# ALTER CATALOG

The ALTER CATALOG command changes the configured attribute values of the catalog process that affect the management of online and audit dumps.

```
ALTER CATALOG, attribute [,attribute]
```

where *attribute* is:

```
[RELEASED {OFF | ON}]
```

```
[RETAINDEPTH integer]  
[RESET RETAINDEPTH ]
```

```
[ROUNDROBIN {OFF | ON}]
```

RELEASED {OFF | ON}

determines the status of an online or audit dump tape volume after all the dumps on it have been marked for deletion.

- RELEASED OFF specifies that when all the dumps on a tape volume have been marked for deletion, the status of the tape volume is changed to SCRATCH. This status indicates that the tape can be reused for dumping and that its contents are no longer available for file recovery.
- RELEASED ON specifies that when all the dumps on a tape volume have been marked for deletion, the status of the tape volume is changed to RELEASED. This status indicates that the tape is not available for reuse in dumping or for file recovery.

Dumps are marked for deletion according to the RETAINDEPTH option specified.

The originally assigned default is RELEASED OFF.

---

**Note.** For this and other optional parameters, the **originally assigned default** is the value initially assigned to the parameter by TMF. The originally assigned default is used for the parameter when no one has explicitly specified or altered this parameter's value in a previous command. For more information about how defaults are assigned, see [Usage Guidelines](#).

---

RETAINDEPTH *integer*

determines the number of online dump generations that remain in the catalog for purposes of file recovery. Each generation is an instance of an online dump that provides a point from which the file recovery can start; if three dumps were made of the same set of files, the first dump would constitute the first generation, the second dump would be the second generation, and so forth. The range for *integer* is 1 through 99. The originally assigned default is 3.

When the number of generations specified by *integer* is reached for all the files in an online dump, the status of the tape volume is changed as follows:

- To SCRATCH if you specified RELEASED OFF
- To RELEASED if you specified RELEASED ON

Increasing *integer* causes the number of tape volumes required by TMF to increase, because more online dump generations are retained in the TMF catalog. In addition, more audit dumps are retained because they are needed for use with the online dump generations during file recovery.

#### RESET RETAINDEPTH

resets the number of online dump generations retained to the originally assigned default of 3.

#### ROUNDROBIN {OFF | ON}

determines the method TMF catalog processes use to select the next scratch tape available to receive new online or audit dumps:

- ROUNDROBIN OFF determines that each scratch-tape request starts its search from the beginning of the list of tapes in the catalog, and selects the first tape available for dumping (in other words, the scratch tape that has the lowest identifying number).
- ROUNDROBIN ON determines that each scratch-tape request starts its search from the tape following the last one used from the list, and selects the next scratch tape available. With ROUNDROBIN ON, all tapes in the list are selected before any one of them is used another time.

The originally assigned default is ROUNDROBIN OFF.

## Security Restrictions

You can issue the ALTER CATALOG command only if you are a member of the super user group.

## TMF State Requirement

You can enter the ALTER CATALOG command at any time, regardless of whether TMF has been started.

## Usage Guidelines

The TMF catalog maintains entries for audit dumps, online dumps, and tape media. Dump entries contain the names of the files dumped, the names of the tape or disk volumes containing the dumps, dump serial numbers, and other information for online dumps and audit dumps. The tape media entries contain information about all tape volumes known to TMF; each entry includes the name of the tape volume and its

current status. The catalog is maintained by the catalog process, which starts when the START TMF command is issued and stops when the STOP TMF command completes execution.

The catalog process starts in the same processors as the transaction management process (TMP), unless otherwise directed by the [ALTER PROCESS](#) command.

Attributes that you do not specify remain unchanged. So, if you omit an optional parameter, one of the following actions takes place:

- If one or more previous TMFCOM commands (such as an earlier ALTER CATALOG command) specified the parameter, the value entered in the last of these commands remains assigned.
- If no previous TMFCOM command specified the parameter, the originally assigned default remains assigned.

If you enter a DELETE CATALOG or a DELETE TMF command, the originally assigned defaults for ALTER CATALOG again take effect when a new catalog is created.

In the ALTER CATALOG command, the RELEASED ON option is useful if you want to retain TMF catalog entries for use in archiving dumps that are no longer needed for file recovery. For example, if you want to retain all audit trails indefinitely, you can use the TMF catalog in this way to keep track of them. If you use this option and later want to remove the dumps from the catalog, however, you must manually delete them.

Whenever a tape is needed to satisfy a tape request, a TMF catalog process searches the list of tapes in the catalog, looking for the next available scratch tape. The process bypasses all tapes that are not available and uses the first available tape found. The ROUNDROBIN option lets you determine how the tape search is conducted:

- When ROUNDROBIN is OFF, a catalog process always starts its search from the beginning of the list. As a result, the catalog process always uses the lowest numbered scratch tape available. Over time, this method typically results in lower numbered tapes being used often while tapes near the end of the list are used rarely (or perhaps not at all).
- When ROUNDROBIN is ON, the catalog process begins its search with the tape after the one last selected and selects the next available tape. If no available tape is encountered, the process goes through the list completely before starting over again. Over time, this method ensures that all scratch tapes are used before any tape is re-used. In this way, catalog processes distribute tape use over the entire pool of available tapes, minimizing tape wear throughout the pool. (Tape wear is especially problematical for customers using tape silos in round-the-clock operation.) An additional advantage of ROUNDROBIN ON is that data on a tape is not overwritten until the last possible moment, thereby extending the data-

protection window for your files. Because of these advantages, ROUNDROBIN ON is the generally recommended setting for this option.

---

**Note.** ROUNDROBIN does not support the ability to divide the TMF tapes into multiple pools. In any TMF configuration, there is only one pool of tapes from which TMF can select.

---

To display the complete list of tapes in the TMF catalog, use the INFO TAPEMEDIA command.

For complete information about the TMF catalog and its contents, see the *TMF Operations and Recovery Guide*.

## Examples

The INFO CATALOG command displays the current catalog process attribute settings:

```
TMF 41> INFO CATALOG
```

```
Catalog configuration:
  retaindepth 2, released on, roundrobin off
```

The ALTER CATALOG command changes the RETAINDEPTH setting from 2 to 4:

```
TMF 43> ALTER CATALOG, RETAINDEPTH 4
```

Now, for each online dump, the TMF catalog retains four generations.

The next INFO CATALOG command displays the new catalog process attribute settings:

```
TMF 43> INFO CATALOG
```

```
Catalog configuration:
  retaindepth 4, released off, roundrobin off
```

The following command selects round-robin tape searching:

```
TMF 44> ALTER CATALOG, ROUNDROBIN ON
```

The next command disables round-robin, specifying the standard method of tape searching:

```
TMF 45> ALTER CATALOG, ROUNDROBIN OFF
```



# ALTER DATAVOLS

The ALTER DATAVOLS command changes the configured attribute values of a data volume.

```
ALTER DATAVOL[S] { volume
                    { (volume [, volume] ...) }
                    [, RECOVERYMODE { ONLINE | ARCHIVE } ]
```

*volume*

identifies the data volume whose attributes are to be altered.

---

**Note.** TMF is shipped from the factory with the \$DSMSCM disk volume configured as a TMF data volume. Although you are free to add other TMF data volumes to the configuration, you should not remove \$DSMSCM as a TMF data volume. See also the note about the preconfiguration of active-audit volumes that appears under the discussion of the [ALTER AUDITTRAIL](#) command.

---

RECOVERYMODE { ONLINE | ARCHIVE }

determines whether audit-trail files associated with the data volume are held on disk until they are no longer required for volume recovery.

ONLINE retains sufficient audit-trail files on disk so that volume recovery does not need to restore audit records from an archived audit dump. Retaining audit-trail files on disk, however, can affect audit trail management and transaction processing. As an example, if this retention causes an audit trail to exceed a capacity threshold before the TMP is allowed to purge the audit-trail file, transaction processing is disabled.

ARCHIVE releases audit-trail files so that they are not retained on disk solely for volume recovery; therefore, if volume recovery becomes necessary, one or more audit dumps might need to be restored. You can only select ARCHIVE mode if audit dumping is configured for the associated audit trail.

The originally assigned default is ONLINE.

---

**Note.** For this optional parameter, the **originally assigned default** is the value initially assigned by TMF. The originally assigned default is used for the parameter when no one has explicitly specified or altered this parameter's value in a previous command. For more information about how defaults are assigned, see [Usage Guidelines](#).

---

## Security Restrictions

You can issue the ALTER DATAVOLS command only if you are a member of the super user group.

## TMF State Requirement

Before you can enter the ALTER DATAVOLS command, TMF must be started.

## Usage Guidelines

The volumes acted on by this command must be in the started, disabled, or down state. (These states, and others, are explained in the discussion of the [STATUS DATAVOLS](#) command.)

- 
- △ **Caution.** TMF does not guarantee the availability of audit dumps required to perform volume recovery for data volumes configured with RECOVERYMODE ARCHIVE; those audit dumps must still exist in the TMF catalog when the recovery is attempted. Sufficient audit dumps are normally retained if any file on the volume has an online dump recorded in the catalog.
- 

Altering a data volume's RECOVERYMODE from ARCHIVE to ONLINE does not restore to disk any previously purged audit dumps that are required for online volume recovery. The volume recovery process itself, however, can restore the required dumps if they are still available.

Attributes that you do not specify remain unchanged. So, if you omit the optional RECOVERYMODE parameter, one of the following actions takes place:

- If one or more previous TMFCOM commands (such as [ADD DATAVOLS](#) or ALTER DATAVOLS) specified the parameter, the value entered in the last of these commands remains assigned.
- If no previous TMFCOM command specified the parameter, the originally assigned default remains assigned.

## Examples

The following ALTER DATAVOLS command changes the RECOVERYMODE attribute to ARCHIVE:

```
TMF 51> ALTER DATAVOLS $DATA, RECOVERYMODE ARCHIVE
$DATA -- datavol altered.
TMF 52>
```

# ALTER DUMPS

The ALTER DUMPS command changes dump entries in the TMF catalog.

```
ALTER DUMPS {file-set
              {(file-set [,file-set]...)}

              [,INVALID {OFF | ON}   ]
              [,RELEASED {OFF | ON} ]
              [,SERIAL serial-number]
```

## *file-set*

identifies the dump files whose entries are to be changed.

For online dumps, enter *file-set* in the following format:

```
[[{volume | *}].]{subvolume | *}.]{file-id | *}
```

An asterisk (\*) indicates “all”; thus, \*.\* indicates all files in all subvolumes on the default volume.

For audit dumps, enter *file-set* in the following format:

```
$ZTMFAT.ZTMFAT.{file-id | *}
```

An asterisk (\*) indicates all files in the \$ZTMFAT.ZTMFAT subvolume. \$ZTMFAT and ZTMFAT are pseudo or dummy volume and subvolume names, respectively, used to support TMF's ability to spread audit-trail files across multiple disk volumes.

INVALID {OFF | ON}

determines whether a dump file is available for file recovery.

OFF denotes that the file is readable and can be used for file recovery. When a file is created, its catalog entry reflects this condition.

ON changes the catalog entry for each specified file to indicate that the file is unreadable and therefore cannot be used.

RELEASED {OFF | ON}

determines whether the specified dumps are available for file recovery.

OFF changes the entries for the specified dumps to indicate that you want these dumps to remain available. Use this option if TMF has marked a dump for deletion from the catalog, but you still want to be able to use that dump.

ON changes the entries for the specified dumps to indicate you no longer need these dumps. Use this option when recent dumps exist in the catalog but have been damaged or marked as deleted.

If the catalog shows only one valid dump for a given serial number, and you release that dump, the entire dump serial number is removed from the catalog.

`SERIAL serial-number`

directs this ALTER DUMPS command to change the catalog entries for only the dumps with the specified serial number. You can obtain information about dump serial numbers by using the [INFO DUMPS](#) command, or by examining reports produced for an audit dump or online dump.

## Security Restrictions

You can issue the ALTER DUMPS command only if you are a member of the super user group.

## TMF State Requirement

Before you can enter the ALTER DUMPS command, TMF must be started.

## Usage Guidelines

Use the ALTER DUMPS command to change the INVALID flag or the RELEASED flag for one or more dump entries in the catalog. As an example, if you deleted a file that also caused a dump to be deleted from the TMF catalog, you could use the ALTER DUMPS command to correct the catalog entry to recover the file.

A database file is unavailable if it is purged or if the FUP ALTER command is used to assign the NO AUDIT option to the file. If all files in a dump are unavailable, all records for the dump are removed from the catalog.

After TMF's TMFSERVE process completes the changes requested by the ALTER DUMPS command, it searches the TMF catalog for other candidate dump entries that might require changing. This search might take a long time, perhaps even appearing like TMFSERVE is in an endless loop. If you encounter this situation but can determine that all relevant changes have been applied, you can stop the TMFSERVE process.

## Examples

The following ALTER DUMPS command changes the catalog entry for a dump to indicate that the dump is not available for use by file recovery. The dump is identified by Serial Number 137.

```
TMF 56> ALTER DUMPS $SYSTEM.TMFTP10F.SRDSTP01,RELEASED ON, &  
>>>SERIAL 137
```

# ALTER PROCESS

The ALTER PROCESS command changes the configured attribute values of various TMF process types.

```
ALTER PROCESS { process-type
                { (process-type [, process-type] ...)
                *
              }

    [, CPUS cpu-list]
    [, RESET CPUS      ]

    [, PRI integer]
    [, RESET PRI      ]

    [, SWAP volume-name]
    [, RESET SWAP      ]

    [, MAKESAVEABEND {ON | OFF}]

    [, DEBUG {OFF | ON}]

    [, INSPECT {ON | OFF}]

    [, PROGRAMFILE file-id]
    [, RESET PROGRAMFILE  ]

    [, TERM file-id]
    [, RESET TERM        ]

    [, EXTENDEDSEGSIZE (integer1, integer2)]
    [, RESET EXTENDEDSEGSIZE                ]

    [, PFSSIZE integer]
    [, RESET PFSSIZE  ]
```

*process-type*

identifies the type of TMF process whose configuration attributes are to be altered:

| Process-type    | Process-type Description   |
|-----------------|--|
| AUDITDUMP       | Audit dump process, which dumps audit-trail files to disk or tape.   |
| AUDITRESTORE    | Audit restore process, which restores audit dumps from disk or tape.   |
| BACKOUT         | Backout process, which undoes the effects of incomplete transactions.  |
| CATALOG         | Catalog process, which manages the TMF catalog.  |
| COPYAUDITOVFLOW | Audit overflow process, which writes audit-trail files to overflow-audit volumes.  |
| FILERECOVERY    | File recovery process, which reconstructs unusable audited files during recovery operations.   |
| FILERECOVLIST   | File recovery list process, which supports the file recovery process.  |
| ONLINEDUMP      | Online dump process, which dumps audited database files to tape or disk.   |
| ONLINERESTORE   | Online restore process, which restores online dumps from tape or disk.   |
| TMFMON          | TMF monitor process, which is responsible for highest-priority, performance-critical operations.   |
| TMFMON2         | Surrogate process that provides a processing environment for TMFLIB operations.  |
| TMP             | Transaction management process pair that coordinates TMF configuration and distributed transactions.   |
| VOLUMERECOVERY  | Volume recovery process that recovers database files to their most recent consistent state if they become inconsistent because of a disk volume or system failure. |

---

**Note.** The *process-type* parameter refers to all TMF processes of a specific kind. For example, the AUDITDUMP process type identifies all audit dump processes in the system. More details about the TMF processes appears under [Process Types and Processes](#) on page 3-79.

---

\*

changes the configuration attributes of all process types in the preceding list.

```
CPUS cpu-list
RESET CPUS
```

identifies the processor in which the selected process type runs. *cpu-list* can range from 0 through 15. Enter *cpu-list* in one of the following formats:

```
{cpu-num          }
{(cpu-num [cpu-num]...)}
```

RESET CPUS clears the *cpu-list*.

The processor is assigned as follows:

- If the command specifies a *cpu-list*, this list is searched from beginning to end to beginning again, until a processor is found.
- If the command does not specify a *cpu-list* and the operating system's CMON process is running, by default the CMON process chooses a processor.
- If the command does not specify a *cpu-list* and the CMON process is not running, by default the following assignments take place:
  - For the BACKOUT, FILERECOVERY, and VOLUMERECOVERY process types, TMF assigns the first available processor with the fewest instances of the same process type.
  - For all other configurable process types, TMF assigns the same processor on which the TMP is running.

For the TMFMON and TMFMON2 process types, the CPUS parameter has no effect. These process types run in all processors.

```
PRI integer
RESET PRI
```

specifies the execution priority for the process type. *integer* can range from 1 through 199. If a CMON process is running, the system consults the CMON process and may override your specification with an alternate value. For the TMFMON, TMFMON2, and TMP process types, the PRI parameter has no effect.

RESET PRI resets the default value. For all process types other than TMFMON, TMFMON2, and TMP, the default is 149. For the TMFMON, TMFMON2, and TMP process types, the defaults are 211, 200, and 204 respectively.

```
SWAP volume-name
RESET SWAP
```

identifies the swap volume for the process type's extended segment. The swap volume for the process' extended data segment is always determined by the Kernel-Managed Swap Facility (KMSF) regardless of what value you specify for this attribute or if you omit this parameter altogether. (Also, the swap volume for the process' data segment is always determined by the KMSF.)

RESET SWAP explicitly directs the KMSF to select the default swap volume.

For the TMFMON and TMP process types, the SWAP parameter has no effect.

MAKESAVEABEND {ON | OFF}

determines whether an INSPECT program save-abend file is created if a process of the specified type terminates abnormally. MAKESAVEABEND ON creates the file if abnormal termination occurs. MAKESAVEABEND OFF suppresses creation of the file. The default is MAKESAVEABEND ON.

For the TMFMON process type, the MAKESAVEABEND parameter has no effect.

DEBUG {OFF | ON}

determines whether a process of the specified type enters the DEBUG or INSPECT debugger soon after it starts. DEBUG ON directs the process to enter the debugger. DEBUG OFF prevents the process from entering the debugger (unless other events cause this to happen). The default is DEBUG OFF.

For the TMFMON process type, the DEBUG parameter has no effect.

INSPECT {ON | OFF}

determines whether INSPECT is used if a process of the specified type falls into a debugger after it is started. DEBUG must be specified to take advantage of this option. INSPECT ON selects INSPECT. INSPECT OFF selects DEBUG. The default is INSPECT ON.

For the TMFMON process type, the INSPECT parameter has no effect.

PROGRAMFILE *file-id*  
RESET PROGRAMFILE

identifies the program file that runs for the process type. This parameter must indicate a valid local file name.

RESET PROGRAMFILE selects the default program file, which is the normal program file from the current \$SYSTEM.SYS<sub>nn</sub> subvolume.

For the TMFMON process, the PROGRAMFILE parameter has no effect.

TERM *file-id*  
RESET TERM

identifies the home terminal for the process type; it is useful to specify this option when you are debugging.

RESET TERM selects the default home terminal, which is the home terminal of the TMP system process.

For the TMFMON process type, the TERM parameter has no effect.



```
EXTENDEDSEGSIZE (integer1, integer2)
RESET EXTENDEDSEGSIZE
```

determines the size, in megabytes, of the process' extended segment. The *integer1* and *integer2* values are the initial size and the growth limit, respectively, for the extended segment. The growth limit must be equal to or greater than the initial size and must be a multiple of the initial size. The valid extended segment size range for most process types is 1 to 128; for exceptions and details, see [EXTENDEDSEGSIZE Option Considerations](#) on page 3-85.

RESET EXTENDEDSEGSIZE selects the default extended segment size, a value chosen by the process.

For the TMFMON process type, *integer1* is the size of the TMFLIB segment and *integer2* has no effect. For the CATALOG process type, *integer1* and *integer2* have no effect. For the TMP process type, *integer1* has no effect and *integer2* is used as the initial size of the segment as described under [EXTENDEDSEGSIZE Option Considerations for the TMP](#) on page 3-87.

```
PFSSIZE integer
RESET PFSSIZE
```

identifies the size, in bytes, of the process file segment (PFS) for the process type. All processes are assigned a PFS size that can increase to a maximum of 8 MB. If you specify the PFSSIZE or RESET PFSSIZE option in an ALTER PROCESS command, TMF ignores your specification.

## Security Restrictions

You can issue the ALTER PROCESS command only if you are a member of the super user group. In addition, to alter the DEBUG attribute, you must be logged on under the super ID. (The super ID is the person with the user ID 255,255—in effect, the group manager of the super user group.)

## TMF State Requirement

You can enter the ALTER PROCESS command at any time, whether or not TMF has been started.

## Usage Guidelines

You can use the ALTER PROCESS command to change the configuration currently assigned to one or more TMF process types.

## Process Types and Processes

In the ALTER PROCESS command, the *process-type* parameter refers to one or more processes of a particular kind. For example, the TMFMON process type refers to all TMFMON processes running in the system—or stated another way, all instances of the TMFMON process. When you specify a configuration attribute for a process type, that

specification applies to all processes of this type. For instance, when you specify CPUS 1,2,5 for the BACKOUT process type, all backout processes can run in those processors only.

For each TMF process type, the description and start-up characteristics are listed next. Each description summarizes the process type's purpose, execution characteristics, and default processor assignment. As you will see, some types of processes start when the NonStop operating system is generated, others start when TMF is started, and still others run only when they are needed. Some persist continuously in the system and others do not. Details about processor assignment options appear in the discussion of the CPUS parameter on page [3-77](#).

- **AUDITDUMP**

**Description:** An audit dump process dumps audit trails to secondary storage, such as disk or tape. An audit dump process runs only when an audit trail is full and audit dumping is turned on for that audit trail, or when a manual rollover is needed. By default, audit dump processes are dynamically assigned to one or more processors as needed. You can override the default by adding or removing processors for this process type.

**Process start-up:** Audit dump processes start as they are needed.

- **AUDITRESTORE**

**Description:** An audit restore process restores audit dumps from secondary storage media such as disk or tape. An audit restore process runs only when recovery is necessary. By default, audit restore processes are dynamically assigned to one or more processors as needed. You can override the default by adding or removing processors for this process type.

**Process start-up:** Audit restore processes start as they are needed.

- **BACKOUT**

**Description:** A backout process undoes the effects of incomplete transactions. Four backout processes are always present in the system, and by default run on processors selected by the TMP. You cannot alter the number of backout processes, but you can override their processor assignments. For further information about processor assignment, see [CPU Option Considerations](#) on page 3-84.

**Process start-up:** The four backout processes start when TMF is started and terminate when TMF is stopped.

- **CATALOG**

**Description:** A catalog process is one member of the catalog process pair that maintains, queries, and manipulates entries in the TMF catalog in response to

requests from other TMF processes, TMFCOM commands dealing with tape and dump media, and requests from tape-handling processes. Up to two catalog processes (a primary and a backup process) can run at any time, but never in the same processor.

**Process start-up:** Catalog processes start when TMF is started, and terminate when TMF is stopped. If a catalog process stops or abends while TMF is running, the TMP automatically restarts it.

- **COPYAUDITOVFLOW**

**Description:** An audit overflow process writes audit-trail files to overflow-audit volumes. An audit overflow process runs only when the overflow threshold for an active-audit trail is reached. At most, there can be one audit overflow process running for each audit trail, for a maximum of 16 audit overflow processes. By default, audit overflow processes are dynamically assigned to processors by the TMP as needed. You can override the default by adding or removing processors for this process type.

**Process start-up:** Audit overflow processes start as they are needed.

- **FILERECOVERY**

**Description:** A file recovery process reconstructs unusable audited files during recovery operations. (A file can become unusable for several reasons, including disk failure, mistaken deletion of an audited volume, or incorrect changes to the database by an application program.) A file recovery process runs only during file recovery. By default, file recovery processes are dynamically assigned to one or more processors as needed. You can override the default by adding or removing processors for this process type.

**Process start-up:** File recovery processes start whenever a TMFCOM [RECOVER FILES](#) command is issued.

- **FILERECOVLIST**

**Description:** A file recovery list process translates file sets specified in a TMFCOM [RECOVER FILES](#) command using wild-card characters into a list of fully qualified database tables and file names. A file recovery list process runs only during this translation operation. By default, file recovery list processes are dynamically assigned to one or more processors as needed. You can override the default by adding or removing processors for this process type.

**Process start-up:** File recovery processes start whenever a TMFCOM RECOVER FILES command is issued.

- **ONLINEDUMP**

**Description:** An online dump process dumps audited database files to secondary storage, such as tape or disk. It runs only while the dump is in progress. By default, online dump processes are dynamically assigned to one or more processors as needed. You can override the default by adding or removing processors for this process type.

**Process start-up:** An online dump processes starts whenever the TMFCOM [DUMP FILES](#) command is issued.

- **ONLINERESTORE**

**Description:** An online restore process copies online dump files from archive media to disk, on behalf of a file recovery process. By default, online restore processes are dynamically assigned to one or more processors as needed. You can override the default by adding or removing processors for this process type.

**Process start-up:** An online restore process starts whenever a file recovery process requires that one or more online dumps be restored.

- **TMFMON**

**Description:** A TMFMON (transaction manager monitor) process is responsible for highest-priority, performance-critical operations on the processor. It is chiefly responsible for allocating the memory segments used by TMFLIB data structures and notifying the TMP when disk volumes come up and go down. A TMFMON process runs in every processor on the node. You cannot override this processor assignment.

**Process start-up:** The TMFMON processes start during processor cold-load, and persist continuously in the system.

- **TMFMON2**

**Description:** A TMFMON2 process is a surrogate process that creates the transaction management process (TMP) and allocates the memory segments used by TMFLIB. It also is responsible for resynchronizing TMFLIB data structures in their processors after processor failure and reload. It also cleans up TMFLIB and disk process data structures after a TMF crash or a STOP TMF, ABRUPT command. A TMFMON2 process runs in every processor on the node. You cannot override this processor assignment.

**Process start-up:** The TMFMON2 processes are started during processor cold-load by their associated TMFMON processes, and persist continuously in the system.

- **TMP**

**Description:** A TMP process is one of the transaction management process pairs that coordinates the TMF configuration and distributed transactions. It is in charge of all operations that require a single point of control, such as starting and stopping TMF, initiating the recovery processes, and implementing configuration changes. It manages the data volumes and audit trails, and is the final authority on whether transactions should be committed or aborted. It also creates other TMF processes, as needed. Up to two TMP processes (a primary and a backup process) can run at any time, but never in the same processor.

**Process start-up:** The TMP process pair is started by TMFMON2 whenever a processor configured for the TMP is loaded or reloaded. This process pair persists continuously in the system.

- **VOLUMERECOVERY**

**Description:** A volume recovery process recovers the database files to their most consistent state if they become inconsistent because of a disk volume or system failure. A volume recovery process runs only when an audited volume must be recovered. By default, volume recovery processes are dynamically assigned to one or more processors as needed. You can override the default by adding or removing processors for this process type.

**Process start-up:** Volume recovery processes start whenever TMF attempts to enable a data volume.

## When ALTER PROCESS Takes Effect

The ALTER PROCESS command takes effect the next time a process of the type specified is created. Processes currently running are not affected.

For a new TMP, BACKOUT, or CATALOG process type configuration to take effect, you manually stop and restart the processes as directed in [Table 3-2](#) on page 3-89.

For the TMFMON process type, only the EXTENDEDSEGSIZE parameter has effect; this parameter determines the size of the TMFLIB segment in each processor. A change to this TMFMON process attribute does not take effect until all TMFMON processes are restarted. However, the only way to stop and restart the TMFMON processes is to stop and reload all processors. To accomplish this, you can use either of the following approaches:

- Perform a full system load (cold-load).

- Stop and reload each processor individually

---

△ **Caution.** If you use the second of these approaches, do not stop, at the same time, both members of any processor pair attached to a critical resource such as the \$SYSTEM disk. Ensure that either the primary or backup processor remains running while you stop and reload the other member of the pair.

For more information about the effect of altering process attributes, see [Stopping and Restarting TMF Processes](#) on page 3-89.

---

Even when the CPUS, PRI, SWAP, and PROGRAMFILE attribute values are specified by the ALTER PROCESS command, they may be overridden by CMON (if it is running). As a result, the values displayed by the [INFO PROCESS](#) command may differ from the actual attribute values of the process type.

## Recommendations for Option Use

The CPUS, PRI, and SWAP options of this command are intended for use by operators. However, the MAKESAVEABEND, DEBUG, INSPECT, PROGRAMFILE, TERM, and EXTENDEDSEGSIZE options should be used by TMF experts only; use of these parameters requires a substantial understanding of lower-level aspects of the system software.

## Returning to Default Configuration Values

After you use the ALTER PROCESS command to change a process-type attribute value, the process type uses that value until it is changed or reset by another ALTER PROCESS command. To reset an attribute to its default value for all process types, use the command:

```
ALTER PROCESS *,RESET attribute-name
```

You can also reset an attribute value for a single process type by using the following command:

```
ALTER PROCESS process-type, RESET attribute-name
```

## CPU Option Considerations

When you specify a list of processors in the CPU parameter, TMF goes through them in order of ascending CPU number when starting processes.

When you specify two processors for the catalog process type or the TMP process type, the first processor is used for the primary process and the second processor is used for the backup process.

---

**Note.** For the TMFMON and TMFMON2 process types, the CPUS parameter has no effect.

---

On a given system, when multiple volume recovery processes run concurrently, the maximum limit is one volume recovery process per processor, for up to eight processors.

On NonStop servers, the TMP always runs four concurrent backout processes for each system, regardless of the number of processors on the system. These backout processes are created whenever the START TMF command is issued. You can change their processor assignment with the CPU parameter.

## PRI Option Considerations

For the PRI parameter, you can specify values ranging from 1 through 199, except for the TMP process type (which must be 204), the TMFMON process (which must be 211), and the TMFMON2 process (which must be 200). Applications must always be run at a priority less than the PRI for these three processes.

## EXTENDEDSEGSIZE Option Considerations

The EXTENDEDSEGSIZE option allows you to alter the initial size (*integer1*) and growth limit (*integer2*) for a process type's extended segment. For most process types, the values you enter for *integer1* and *integer2* can range from 1 through 128; exceptions are noted later in this discussion. The *integer2* value must be greater than or equal to the *integer1* value, and must be a multiple of *integer1*.

The EXTENDEDSEGSIZE default values are established for minimum (four-processor) system configurations. You should review these settings as they apply to your actual configuration, and assign new EXTENDEDSEGSIZE values accordingly. You must make these adjustments in either of these cases:

- If a TMF process fails repeatedly. Whenever a process fails because of insufficient memory space, TMF periodically tries to restart that process. An increased EXTENDEDSEGSIZE value usually resolves this situation.
- If you receive these two EMS messages during processor reload or in periods of high transaction activity:

```
*0277*  TMP Process:  Transactions DISABLED due to low
{TMP | TMFLIB} memory availability in CPU 0.
```

```
*0278*  TMP Process:  Transactions reenabled due to restored
memory availability in {TMP | TMFLIB} in CPU 0.
```

The condition reported by these messages is not fatal, but it can cause application programs to experience Error Code 0086 (transaction processing is disabled either by the operator or because one or more TMF limits have been reached).

If your TMF system services many volumes, you may need to adjust the EXTENDEDSEGSIZE values for the recovery process types: BACKOUT, VOLUMERECOVERY, and FILERECOVERY.

For the TMP and TMFMON process types, a precise calculation of the extended segment size for a given transaction load cannot be done. This requirement rests heavily on the size of the transaction control blocks (TCBs) needed to support transaction processing at your site, which in turn depends on the size of your NonStop



server and the applications that you run. As an example, the portion of the TMP's extended segment used to support data volumes, audit trails, and dependent processes depends on the configuration of your system. Larger configurations require more space for control tables. The remaining space is used for dynamic allocation of TCBs. TCB size depends on the number of nodes involved in distributed transactions, the number of messages queued for the TCB, and other factors that are hard to predict. For more details about the EXTENDEDSEGSIZE attribute for the TMP and TMFMON process types, and related differences in TMF behavior, see [EXTENDEDSEGSIZE Option Considerations for the TMP](#) on page 3-87 and [EXTENDEDSEGSIZE Option Considerations for the TMFMON](#) on page 3-88.

For other recommendations for specific process types, see *TMF Planning and Configuration Guide*.

The EXTENDEDSEGSIZE attribute values (*integer1* and *integer2*) have the default and minimum values listed for each process type in the following table:

| Process-type    | <i>integer1</i><br>Default<br>Value   | <i>integer1</i><br>Minimum<br>Value | <i>integer2</i><br>Default<br>Value | <i>integer2</i><br>Minimum<br>Value |
|-----------------|---|-------------------------------------|-------------------------------------|-------------------------------------|
| AUDITDUMP       | 0.5 MB  | 1 MB                                | 8 MB                                | 8 MB                                |
| AUDITRESTORE    | 0.5 MB  | 1 MB                                | 8 MB                                | 8 MB                                |
| ONLINEDUMP      | 0.5 MB  | 1 MB                                | 8 MB                                | 8 MB                                |
| ONLINERESTORE   | 0.5 MB  | 1 MB                                | 8 MB                                | 8 MB                                |
| COPYAUDITOVFLOW | 0.5 MB  | 1 MB                                | 8 MB                                | 8 MB                                |
| BACKOUT         | 5 MB  | 5 MB                                | 40 MB                               | 40 MB                               |
| VOLUMERECOVERY  | 40 MB   | 40 MB                               | 40 MB                               | 40 MB                               |
| FILERECOVERY    | 40 MB   | 40 MB                               | 40 MB                               | 40 MB                               |
| FILERECOVLIST   | 1 MB  | 1 MB                                | 1 MB                                | N/A                                 |
| TMFMON          | 12 MB   | 12 MB                               | N/A                                 | N/A                                 |
| TMFMON2         | 0.25 MB   | N/A                                 | N/A                                 | N/A                                 |
| TMP             | See discussion, <a href="#">EXTENDEDSEGSIZE Option Considerations for the TMP</a> on page 3-87. |                                     |                                     |                                     |
| CATALOG         | N/A   | N/A                                 | N/A                                 | N/A                                 |

For all process types except the TMP, if you specify 128 for *integer2*, TMF internally uses 127.5 MB as the segment growth limit and modifies the specified *integer1* value down so that the initial segment size becomes a multiple of 127.5. The initial segment size and growth limit displayed by the [INFO PROCESS](#) command, however, are the values specified for *integer1* and *integer2* in the ALTER PROCESS command, rather than the segment size and growth limit actually in use. For the TMP, the behavior for *integer1* and *integer2* differs as described under [EXTENDEDSEGSIZE Option Considerations for the TMP](#) on page 3-87.

The values shown in the “*integer1* Default Value” and “*integer2* Default Value” columns are, respectively, the values for *integer1* and *integer2* initially assigned to the process



and used until modified by the ALTER PROCESS command. The following ALTER PROCESS command for the BACKOUT process type, for example, changes the *integer1* value from 5 MB to 20 MB, and the *integer2* value from 40 MB to 80 MB:

```
ALTER PROCESS BACKOUT, EXTENDEDSEGSIZE (20,80)
```

In the “*integer2* Default Value” column, the notation N/A means that the segment size for this process type does not automatically increase; any value you specify for *integer2* in EXTENDEDSEGSIZE is ignored. You can, however, increase the segment size to an upper limit of 128 MB by entering the desired value as the *integer1* parameter. As an example, suppose that you enter the following command:

```
ALTER PROCESS TMFMON, EXTENDEDSEGSIZE (16,24)
```

As a result of this command, TMF assigns TMFMON the value of 16 for *integer1* and ignores the value of 24 specified for *integer2*.

---

**Note.** The TMFMON extended memory is locked and is not available for paging by the operating system.

For the TMFMON2 process type, if you specify an EXTENDEDSEGSIZE from 1 MB through 128 MB, TMF defaults to 0.25 MB. This size is enough for a TMFMON2 process, since the extended segment in this process is used mainly for message system control data structures. This size is not related to transaction rate or transaction load in any way.

---

The values shown in the “*integer1* Minimum Value” and the “*integer2* Minimum Value” columns are the minimum values for *integer1* and *integer2* that can be assigned through the ALTER PROCESS command.

The EXTENDEDSEGSIZE attribute does not apply to the CATALOG process type, which does not use extended segments.

## EXTENDEDSEGSIZE Option Considerations for the TMP

For the TMP process type, as with other process types, both *integer1* and *integer2* in EXTENDEDSEGSIZE must be less than or equal to 128, and *integer2* must be a multiple of *integer1*. However, for the TMP, other EXTENDEDSEGSIZE rules and behavior differ from those for other process types:

- TMF uses the value you specify for *integer2* as the initial segment size, and ignores the value you specify for *integer1*. (Nevertheless, you must still supply a valid value for *integer1*, or TMFCOM reports an error.)
- If you specify a value less than 40 for *integer2*, TMF uses 40 MB as the initial segment size.
- Even though you cannot supply a value greater than 128 for *integer1* or *integer2*, TMF allows the segment to grow as needed until it either:
  - reaches a maximum allowable size of 1024 MB (rounded down to an even multiple of the specified *integer2*), or
  - uses all the KMSF swap space available.

You cannot manually adjust the segment size beyond 128 MB; the growth to the 1024-MB upper limit is managed entirely by TMF.

## EXTENDEDSEGSIZE Option Considerations for the TMFMON

In addition to the considerations common to the TMP and TMFMON described under [EXTENDEDSEGSIZE Option Considerations](#) on page 3-85, the TMFMON's extended segment has a relatively fixed component for various TMF Library tables and the support of disk volumes attached to each CPU. Larger systems, of course, require more space for these tables. The remainder of the segment involves TCBs and message buffers in the TMF Library and other dynamic factors. TCB size depends on the number of files involved in a given transaction and can differ greatly from processor to processor, and message buffer use is very unpredictable.

Even though very precise estimates are difficult, a general operating guideline is possible for the TMFMON. First, estimate the number of active transactions in your system, as reflected by the TRANSCOUNTTHRESH value established by the ALTER BEGINTRANS command. Next, consult the following table and enter the corresponding EXTENDEDSEGSIZE value in the ALTER PROCESS command. Later, if this value proves insufficient, you can raise it further.

| TRANSCOUNTTHRESH | EXTENDEDSEGSIZE |
|------------------|-----------------|
| 3000             | 10              |
| 3200             | 12              |
| 4000             | 16              |
| 5000             | 16              |
| 6000             | 18              |
| 7000             | 20              |

---

**Note.** The values in this table are based on the TRANSPERCPU default value, 1024, which limits the number of transactions that can be active in a processor. Changing that value would alter the fixed portion of the TMF library's extended segment used. Therefore, if you increase the TRANSPERCPU value, you should also increase the EXTENDEDSEGSIZE value.

---

## Stopping and Restarting TMF Processes

Some types of TMF processes persist in the system but can be manually stopped and restarted, as described in [Table 3-2](#). Others are transient processes that start and stop automatically as needed, as also noted in this table.

**Table 3-2. Stopping and Restarting TMF Processes**

| Process Type       | How Stopped and Restarted   |
|--------------------|---|
| TMP                | <p>To manually stop and restart the TMP process pair:</p> <ol style="list-style-type: none"> <li>1. Issue a STOP TMF command to obtain a clean shutdown of TMF.</li> <li>2. Issue a STOP TMF, ABRUPT command to stop the TMP processes. (This operation is equivalent to a controlled TMF crash, which terminates both the primary and backup TMPs. A new TMP process pair is created automatically.)</li> <li>3. Issue a START TMF command to restart TMF.</li> </ol>  |
| BACKOUT<br>CATALOG | <p>To manually stop and restart all backout or catalog processes:</p> <ol style="list-style-type: none"> <li>1. Issue a STOP TMF command to obtain a clean shutdown of TMF. This action stops all processes of these types.</li> <li>2. Issue a START TMF command to restart TMF. This action restarts the processes.</li> </ol> <p>An alternative way to stop and restart the backout or catalog processes is by entering, through the TACL command interpreter, a STOP <i>process-name</i> command for each process. For the backout <i>process-name</i>, use \$XBK<i>n</i>, where <i>n</i> is a digit uniquely identifying the particular process; for the catalog process pair, use \$XCAT. After they are stopped in this way, the TMP automatically restarts the backout and catalog processes.</p> |
| TMFMON<br>TMFMON2  | <p>The TMFMON and TMFMON2 processes are started automatically during system cold load and processor reload, and run until the system is shut down or a processor fails. The only way to stop and restart these processes is to cold load the system or shut down and reload individual processors.</p>  |

**Table 3-2. Stopping and Restarting TMF Processes**

| Process Type    | How Stopped and Restarted  |
|-----------------|--|
| AUDITDUMP       | Processes of these types are transient in the system; they are started automatically as they are needed, and terminate automatically when they are no longer needed. |
| AUDITRESTORE    |  |
| COPYAUDITOVFLOW |  |
| FILERECOVERY    |  |
| FILERECOVLIST   |  |
| ONLINEDUMP      |  |
| ONLINERESTORE   |  |
| VOLUMERECOVERY  |  |

## Cold Loading and Reloading Considerations

All settings established by the ALTER PROCESS command persist across subsequent cold-load operations.

Altering the program file for the TMFMON2 process type does not require cold loading all processors. However, the ALTER PROCESS command does not take effect for a particular processor until that processor is reloaded. There is no way to stop an individual TMFMON2 process without stopping its processor. To avoid possible incompatibilities, you might wish to cold load the system after altering the TMFMON2 program file name.

If the TMFMON2 process type's SWAP, TERM, or PROGRAMFILE attributes are modified so that the swap volume, terminal, or program file is inaccessible when a processor is reloaded, the attempt to create the TMFMON2 process in that processor fails. Now, TMF tries again to create TMFMON2 in this processor, this time ignoring the configured value of the attribute causing the problem and using instead that attribute's default value. Although TMF starts this TMFMON2 process with one or more default values, the configuration attributes are not permanently modified. Thus, if the attribute values no longer cause errors when future TMFMON2 processes are created, these new processes run with the configured values rather than the defaults.

TMF fails if the SYSTEM\_TERMINAL option was not used in the system generation operation. When this failure occurs:

- The TMFMON2 process in the cold-load processor (named \$ZTMnn) may repeat the following EMS event message every 10 seconds:

```
TMF *0182* TMP Process #0: The TMF process TMP could not be
started or it failed. Completion Code: 4. Completion Info:
8 PROCESS_CREATE_error 8, 14 on file TMFTMP
```

- Attempts to reload any other processor will fail, and the target processor will halt with halt code %1002.

To correct this situation, you can use the TERM option in the ALTER PROCESS command as follows, with *hometerm* indicating a valid home terminal to be used by the TMP and TMFMON2 process types in TMF:

```
ALTER PROCESS (TMFMON2, TMP), TERM hometerm
```

## Examples

The following ALTER PROCESS command alters the configuration of the catalog process type so that processes of this type run on CPUs 0 and 1:

```
TMF 62> ALTER PROCESS CATALOG, CPUS (0,1)
```

The next command specifies that all backout processes run on CPU 1:

```
TMF 63> ALTER PROCESS BACKOUT, CPUS 1
```

---

**Note.** Altering the backout process' processor attribute limits the initiation of backout processes to the processors specified in the CPUS option list. On configurations with multiple processors, this action can cause multiple backout processes to run in a single processor. When the backout process CPU attribute is set to the default value, the allocation of processors for backout processes is dynamic. See the description of the CPUS parameter, and the discussion [CPU Option Considerations](#) on page 3-84, for further information.

---

The following command specifies that online dump processes should run at a priority of 160:

```
TMF 64> ALTER PROCESS ONLINEDUMP, PRI 160
```

The following command sets the initial size of the extended segment for each backout process to 6 MB, and the growth limit for this segment to 42 MB:

```
TMF 65> ALTER PROCESS BACKOUT, EXTENDEDSEGSIZE (6,42)
```

The next command enables the INSPECT debug program for volume recovery processes, causes debug mode to be enabled whenever a volume recovery process starts, and causes the process to generate a saveabend file when terminating abnormally:

```
TMF 66> ALTER PROCESS VOLUMERECOVERY, INSPECT ON, DEBUG ON, &  
>>>MAKESAVEABEND ON
```

The following command changes the home terminal for file recovery processes to \$TERM07:

```
TMF 67> ALTER PROCESS FILERECOVERY, TERM $TERM07
```

# ALTER TAPEMEDIA

The ALTER TAPEMEDIA command changes the configured attribute values of tape volumes.

```
ALTER TAPEMEDIA { tape-volume
                  { (tape-volume [ , tape-volume ] ... ) }

, STATUS { ASSIGNED }
         { BAD }
         { RELEASED }
         { SCRATCH }
```

*tape-volume*

identifies the tape volume whose entries in the catalog are to be changed. For this parameter, use the name magnetically encoded on the tape volume.

STATUS

specifies the status of the tape volume named in this command. STATUS is required and must include one of the following keywords:

ASSIGNED

indicates that the tape volume contains valid dumps. Any tape volume specified becomes available for file recovery, provided that the file dumps contained on that volume have not been removed from the catalog. For information about re-adding dumps to the catalog, see the [ADD DUMPS](#) command description.

BAD

indicates that the tape volume is defective and cannot be used later for dumping files or for file recovery.

RELEASED

indicates that the contents of the tape volume are no longer required for file recovery, and that the tape is not available for re-use in dumping or for file recovery. To use a tape older than that indicated by the RETAINDEPTH attribute for file recovery, you can change the status of tape volumes from RELEASED to ASSIGNED. (For information about RETAINDEPTH, see the [ALTER CATALOG](#) command description.)

SCRATCH

indicates that the tape volume can be used for dumping files.

## Security Restrictions

You can issue the ALTER TAPEMEDIA command only if you are a member of the super user group.

## TMF State Requirement

Before you can enter the ALTER TAPEMEDIA command, TMF must be started.

## Examples

The following ALTER TAPEMEDIA command changes the status of the tape volume TMF006 to indicate that this volume should not be used:

```
TMF 71> ALTER TAPEMEDIA TMF006, STATUS BAD
```

The INFO TAPEMEDIA command displays the new status:

```
TMF 72> INFO TAPEMEDIA
Media Name    Media Type    Media Status
-----
TMF001        tape         assigned
TMF002        tape         assigned
TMF003        tape         scratch
TMF004        tape         assigned
TMF005        tape         assigned
TMF006        tape         bad
```

# ALTER TMF

The ALTER TMF command changes the TMF configuration volume or switches the functions of the TMP's primary and backup processes.

```
ALTER TMF, {CONFIGVOL volume }  
           {SWITCHPROCESS TMP }
```

CONFIGVOL *volume*

specifies the name of the volume that contains the configuration files to be used in the next TMF startup. The volume must be accessible by the TMFMON2 process, which creates other TMF processes during TMF startup.

SWITCHPROCESS TMP

switches the function of the TMP's primary and backup processes, effectively changing the primary process to the backup and vice-versa.

## Security Restrictions

You can use the ALTER TMF command only if you are a member of the super user group.

## TMF State Requirement

Before you can enter the ALTER TMF command with the CONFIGVOL *volume* option, TMF must be stopped. However, you can enter the ALTER TMF command with the SWITCHPROCESS TMP option at any time, whether or not TMF is running.

## Usage Guidelines

With the ALTER TMF command, you can select either of two operations: changing the TMF configuration volume or switching the TMP's primary and backup processes. When entering an ALTER TMF command, you select one or the other of these operations, but not both.

## Changing the TMF Configuration Volume

With the CONFIGVOL option, the ALTER TMF command causes the TMP process to change the TMF configuration volume and then stop itself. The new TMP process, re-created automatically, uses the new configuration volume. If there are no configuration files located in the ZTMFCONF subvolume on that volume, the TMP creates new ones.



You can restart TMF with a [START TMF](#) command.

- 
- △ **Caution.** Be very careful when entering the ALTER TMF command with the CONFIGVOL option, because this command makes the previous TMF configuration inaccessible by changing the configuration volume. Because your TMF configuration is thereafter defined on the new configuration volume, it is a good precaution to ensure that this volume is a member of a mirrored disk pair.

To make the previous TMF configuration active again, you must enter another ALTER TMF command, giving the previous configuration volume name in *volume*.

---

If the configuration volume becomes inaccessible to the TMFMON2 process, the TMP fails. To recover from this situation, issue an ALTER TMF command that specifies the old configuration volume.

## Switching the Primary and Backup TMP Process

With the SWITCHPROCESS TMP option, the ALTER TMF command causes the primary TMP process to pass control to the backup TMP process. Thus, the primary process becomes the backup, and the backup process becomes the primary.

The SWITCHPROCESS TMP option only takes effect when a backup TMP process exists.

## Examples

This ALTER TMF command changes the configuration volume from \$SYSTEM to \$CAT01:

```
TMF 76> ALTER TMF, CONFIGVOL $CAT01
```

The following command switches the primary and backup TMP processes:

```
TMF 77> ALTER TMF, SWITCHPROCESS TMP
```

# CANCEL OPERATIONS

The CANCEL OPERATIONS command cancels a specific TMF file-recovery or dump-files operation.

`CANCEL OPERATION[S] operation-number`

*operation-number*

is an integer that identifies the operation to be canceled, as described under “Usage Guidelines.” The lowest operation number assigned is 1, and the highest is 100.

## Security Restrictions

You can issue the CANCEL OPERATIONS command only if you are a member of the super user group.

## TMF State Requirement

Before you can enter the CANCEL OPERATIONS command, TMF must be started.

## Usage Guidelines

The CANCEL OPERATIONS command affects only TMF file-recovery and dump-files operations. To remove other types of operations, you must use the STOP TMF, ABRUPT command.

TMF recognizes each specific instance of a particular kind of operation (such as a TMF file-recovery or dump-files operation) as a unique operation and assigns an operation serial number that distinguishes this operation from other recent operations of the same type. If you use the NOWAIT option in the command that generates an operation, TMFCOM displays this number at your terminal. You can subsequently cancel a file-recovery or dump-files operation by entering this number in the *operation-number* parameter of the CANCEL OPERATIONS command.

The operation numbers generated are re-used, to keep the number of digits low as an aid in remembering these numbers; after the highest number allowed (100) is assigned, TMF returns to the lowest possible number (1) to identify the next operation.

For more information about operation numbers and how they are assigned and displayed, see the discussion of the [DISPLAY OPERATIONS](#) command.

## Examples

The following CANCEL OPERATIONS command cancels the operation identified by *operation-number* 45:

```
TMF 81>CANCEL OPERATION 45  
TMF 82>
```

# CLOSE RESOURCEMANAGER

The CLOSE RESOURCEMANAGER command initiates closure of one or more recoverable resource managers that are currently open by a gateway process.

```
CLOSE {RESOURCEMANAGER | RM} { rm-name
                                { (rm-name [ , rm-name ] . . . ) } }
```

*rm-name*

is the name of the recoverable resource manager to be closed. This name can contain a maximum of 128 characters; the first character must be a letter, and the others can be alphanumeric or any of the following special characters:

\$ \_ - ^ @ &

Names that begin with ZZ-VOLATILE- are not allowed in this parameter. Names containing adjoining hyphens (--) are not allowed, although names using multiple hyphens separated by one or more intervening characters are legal. (TMFCOM interprets two consecutive hyphens as the start of a comment.) Thus, the name "ALPHA--1" is not permitted, but "ALPHA-N-1" is allowed.

## Security Restrictions

You can issue the CLOSE RESOURCEMANAGER command only if you are a member of the super user group.

## TMF State Requirement

Before you can enter the CLOSE RESOURCEMANAGER command, TMF must be started.

## Usage Guidelines

---

**Note.** Resource managers, to which the CLOSE RESOURCEMANAGER command applies, support heterogeneous transaction processing. Resource managers are used in products such as NonStop TUXEDO. Although TMFCOM and the TMFSERVE programmatic interface provide TMF commands and options for operating on resource managers, system managers who use them do so in the context of issues involving multiple software subsystems and inter-platform considerations. Complete discussion of heterogeneous transaction processing and the context in which resource-manager commands and options are used lies beyond the scope of the TMF manual library.

For more information about heterogeneous transaction processing, see the Open Group TRANSACTION PROCESSING Publications, available from X/Open Publications at the following Web location:

<http://www.opengroup.org/pubs/catalog/tp.htm>

These publications can also be ordered through many bookstores.

---

The CLOSE RESOURCEMANAGER command itself does not directly close a specified recoverable resource manager. However, when this command is executed, the gateway process that opened the resource manager receives a signal indicating that the operator requested closure of the resource manager. If the gateway process tries to use the file number associated with its open instance of the resource manager, that process encounters Error 66 (FEDEVDOWN). To continue processing transactions for this resource manager, the gateway process must explicitly close and then re-open the resource manager. To allow other gateway processes to use this resource manager, the gateway process must close it.

The CLOSE RESOURCEMANAGER command also forces any active transactions that have branches with a specified resource manager to abort. If this is a recoverable resource manager with unresolved transaction branches, the resource manager enters the crashed (rather than the closed) state.

If the specified resource manager is not currently open, an error message appears.

## Examples

The following CLOSE RESOURCEMANAGER command forces closure of a recoverable resource manager named TUX-A1\_UNIX@SUN:

```
TMF 86> CLOSE RESOURCEMANAGER TUX-A1_UNIX@SUN
```

The next command forces closure of a recoverable resource manager named TUX-A2\_NT@MICROSOFT:

```
TMF 87> CLOSE RM TUX-A2_NT@MICROSOFT
```

# CMDVOL

The CMDVOL command specifies the default volume and subvolume names for expanding unqualified file names other than those in OBEY commands.

|  |
|--|
| <pre>CMDVOL [ <i>volume</i> ]<br/>      [ [ <i>volume</i> . ] <i>subvolume</i> ]</pre> |
|--|

*volume*

is the name of the default disk volume.

*subvolume*

is the name of the default subvolume.

## Security Restrictions

Anyone can enter the CMDVOL command.

## TMF State Requirement

You can enter the CMDVOL command at any time, whether or not TMF has been started.

## Usage Guidelines

Through the CMDVOL command, you can specify default *volume* and *subvolume* names that are recorded for your session and that remain in effect until you issue another CMDVOL command.

If you do not issue a CMDVOL command, no default volume and subvolume are recorded for your session. In this case, unqualified file names are expanded according to the defaults in effect when you started TMFCOM.

If you enter the CMDVOL command without specifying either *volume* or *subvolume*, the default action depends on the parameters you omit:

- If you enter *volume* but omit *subvolume*, the default subvolume in effect when you started TMFCOM becomes the recorded default subvolume for your session.
- If you omit *volume* but enter *subvolume*, the default volume in effect when you started TMFCOM becomes the recorded default volume for your session.
- If you omit both *volume* and *subvolume*, the recorded default volume and subvolume are cleared. As a result, unqualified file names are expanded according to the defaults in effect when you started TMFCOM

At any time, you can display the currently recorded default volume and subvolume for your session, as well as the default volume and subvolume in effect at the beginning of your session, by entering the [ENV](#) command.

If you specify an invalid volume name in *volume*, or an invalid subvolume name in *subvolume*, an error message appears and the CMDVOL command is ignored.

The CMDVOL command applies to all unqualified file names encountered in TMFCOM commands during this session except file names used as [OBEY](#) command parameters. (However, the CMDVOL command does apply to file names that appear within OBEY command files themselves.)

The distinction between the CMDVOL command and the [OBEYVOL](#) and [VOLUME](#) commands, described later in this section, is as follows:

- The CMDVOL command applies to all unqualified file names except those used as [OBEY](#) command parameters.
- The [OBEYVOL](#) command applies only to unqualified file names used as [OBEY](#) command parameters.
- The [VOLUME](#) command applies to all unqualified file names encountered in all TMFCOM commands during the session, except those used as [CMDVOL](#) and [OBEYVOL](#) command parameters.

## Example

The following CMDVOL command specifies \$DATA and RTL9001 as the default volume and subvolume, respectively, for expanding unqualified file names used in interactive TMFCOM commands during this TMFCOM session (except those used in OBEY commands):

```
TMF 89> CMDVOL $DATA.RTL9001
```

The Command Volume entry in the output from the following ENV command reflects this change:

```
TMF 90> ENV
Default Volume      \TSII.$BOOKS1.STAFF
Volume              None Specified
Command Volume      $DATA.RTL9001
Obey Volume         None Specified
In File             \TSII.$ZTN1A.#PTGUJ0U
Out File            \TSII.$ZTN1A.#PTGUJ0U
Server Process      \TSII.$Z34H
```

# DELETE CATALOG

The DELETE CATALOG command clears from the TMF catalog all online dump, audit dump, and media entries, and resets to 64 the counter used for assigning dump serial numbers.

`DELETE CATALOG`

## Security Restrictions

You can enter the DELETE CATALOG command only if you are logged on under the super ID; the super ID is the person with the user ID 255,255—in effect, the group manager of the super user group

## TMF State Requirement

Before you can enter the DELETE CATALOG command, TMF must be started.

## Usage Guidelines

Because the DELETE CATALOG command clears all dump and tape-volume entries from the catalog, you will not be able to use online and audit dumps that were made before this command was issued.

- 
- △ **Caution.** Use extreme caution when entering the DELETE CATALOG command, because this command removes TMF's ability to recover your database from previously dumped files.
- 

The DELETE CATALOG command displays a prompt that allows you to confirm the operation.

After issuing the DELETE CATALOG command, be sure to make new online dumps to support future restore operations.

## Example

The following DELETE CATALOG command produces the normal prompt for confirmation:

```
TMF 90> DELETE CATALOG
```

```
WARNING: The Delete Catalog command will throw away all
         online dump audit dump and media entries. Use
         CAUTION when performing the Delete Catalog
         command.
```

```
Do you confirm the DELETE CATALOG? (Y/N)  y
.
.
.
```



# DELETE DATAVOLS

The DELETE DATAVOLS command removes one or more data volumes from the TMF configuration.

```
DELETE DATAVOL[S] { volume
                     { (volume [ , volume ] ... ) }
                     [ , ALLOWINCONSISTENCY ]
```

*volume*

identifies a data volume to be deleted.

ALLOWINCONSISTENCY

indicates that the data volumes specified are to be deleted even if it is uncertain that they contain only consistent data.

Normally, if you attempt to delete a data volume in the disabled-dirty state, the DELETE DATAVOLS command fails; however, specifying ALLOWINCONSISTENCY in this command overrides this protective mechanism and allows TMF to delete the volume despite potential inconsistencies.

## Security Restrictions

You can issue a DELETE DATAVOLS command only if you are a member of the super user group.

## TMF State Requirement

Before you can enter the DELETE DATAVOLS command, TMF must be running and the data volumes to be removed must be disabled with the [DISABLE DATAVOLS](#) command.

## Usage Guidelines

After each delete operation, the DELETE DATAVOLS command displays a message stating whether the deletion was completed successfully, and if not, the reason why.

After a data volume has been deleted from the TMF configuration, it cannot be restarted for transaction processing until it has been re-added. In addition, after a data volume is deleted, no volume recovery can be performed until it is added again to the configuration and started.

---

△ **Caution.** After a data volume has been deleted with ALLOWINCONSISTENCY selected, volume recovery cannot return it to a consistent state.

---

If file recovery is no longer needed for the files on a deleted data volume, issue a [DELETE DUMPS](#) command to remove any associated online dumps from the TMF

catalog; this allows the audit dumps associated with the deleted volume to be recycled. If any online dumps for the deleted volume remain in the catalog, however, audit dump media will not be recycled, and the performance of the catalog process will eventually degrade as the catalog database grows. Furthermore, if the data volume is later added back into the configuration, and associated with a different audit trail than before, any attempt at file recovery using an online dump made before the audit-trail change will fail.

## Examples

The following DELETE DATAVOLS command successfully deletes a disabled-clean volume:

```
TMF 91> DELETE DATAVOLS $DATA3
$DATA3 -- datavol deleted.
TMF 92>
```

The next DELETE DATAVOLS command attempts to remove a volume that was not disabled:

```
TMF 92> DELETE DATAVOLS ($PART, $SUPP)
$PART -- datavol not deleted (not disabled).
$SUPP -- datavol deleted.
TMF 93>
```

The following DELETE DATAVOLS command illustrates what happens when you attempt to delete a disabled-dirty data volume, first without specifying the ALLOWINCONSISTENCY option and then with the option supplied:

```
TMF 93> DELETE DATAVOLS $DATA7
$DATA7 -- datavol not deleted (dirty, not overridden).
TMF 94> DELETE DATAVOLS $DATA7, ALLOWINCONSISTENCY
$DATA7 -- datavol deleted.
TMF 95>
```

For more information on the disabled-dirty and other data volume states, see the [STATUS DATAVOLS](#) command description.

# DELETE DUMPS

The DELETE DUMPS command removes dump entries from the TMF catalog and attempts to delete the associated dump files from disk.

```
DELETE DUMPS {file-set } [!]  
              {(file-set [,file-set]...)} [!]  
  
              [,SERIAL serial-number]
```

*file-set* [!]

identifies the files in the online dump or the audit dump whose entries are to be deleted from the catalog.

For online dumps, enter *file-set* in the following format:

```
[[{volume | *}].]{subvolume | *}].]{file-id | *}
```

An asterisk (\*) indicates “all”; thus, \*. \* indicates all files in all subvolumes on the default volume. An asterisk used in *volume* could result in an inefficient and time-consuming operation; for example, if you have many disk volumes but only want to delete files from two of them, it might be more efficient to enter two separate DELETE DUMPS commands. An asterisk used in *subvolume* does not impact the operation so severely and can be used with less discretion.

For audit dumps, enter *file-set* in the following format:

```
$ZTMFAT.ZTMFAT.{file-id | *}
```

An asterisk (\*) indicates all files in the \$ZTMFAT.ZTMFAT subvolume. \$ZTMFAT and ZTMFAT are pseudo or dummy volume and subvolume names, respectively, used to support TMF's ability to spread audit-trail files across multiple disk volumes.

For both online dumps and audit dumps, in the *file-set* list, file names are expanded with default volume and subvolume names, and the space required for them is impacted by other overhead as well. All *file-set* information cannot exceed the size of the SPI message buffer that supports communication between TMFCOM and TMFSERVE, which is 28 KB. If this limit is exceeded, TMFCOM displays Error Message 1050.

If you specify the exclamation point (!), the catalog entry is deleted even if the purge of the disk dump fails. If the disk dump purge fails, a warning message appears at the home terminal and an event message is generated in the EMS log.

If you omit the exclamation point and an error occurs when purging the dump from disk, the entry is not deleted from the catalog and a warning message is generated to report the event.

`SERIAL serial-number`

directs this DELETE DUMPS command to remove the catalog entries only for dumps with the specified serial number. You can obtain information about dump serial numbers by using the [INFO DUMPS](#) command or by examining reports produced for an audit or online dump.

## Security Restriction

You can issue the DELETE DUMPS command only if you are a member of the super user group.

## TMF State Requirement

Before you can enter the DELETE DUMPS command, TMF must be started.

## Usage Guidelines

You can specify that entries for certain files in a dump having a particular serial number are deleted from the catalog. In this case, other entries for files having the same serial number remain in the catalog.

When you enter the DELETE DUMPS command, the status of each dump you select for deletion is changed from ASSIGNED (needed for recovery) to RELEASED (not needed for recovery) in the TMF catalog. The dump entries for a specific medium (tape reel or disk subvolume) remain in the catalog, however, until all dump files on that medium are marked either RELEASED or INVALID; then TMF removes all dumps on that medium from the catalog and performs the following operations on the medium itself:

- For a tape medium, TMF marks the appropriate tape reels as either RELEASED or SCRATCH, depending on the setting of the RELEASED catalog configuration option.
- For a disk medium, TMF purges the files containing the dumps.

As long as other valid (ASSIGNED) dumps exist on a medium, that medium retains its ASSIGNED status. To display the status of the dumps on an ASSIGNED medium, issue the INFO DUMPS command with the DETAIL option; the resulting display shows which dumps are ASSIGNED, RELEASED, or INVALID.

Specifying DELETE DUMPS \*.\* without the SERIAL option deletes all dump entries in the TMF catalog; therefore, when you enter this command, TMFCOM prompts you to verify that you indeed want to take this action. To confirm your intent, respond to the prompt by entering YES; TMF then deletes the dump entries. To prevent this deletion, enter NO; TMF prompts you for another command.

## Examples

The following DELETE DUMPS command deletes dumps for the audited file \$DATA3.TMFSN02F.F65289S from the TMF catalog:

```
TMF 6> DELETE DUMPS $DATA3.TMFSN02F.F65289S
```

The next DELETE DUMPS command deletes from the catalog all files that belong to dumps identified by serial number 1000, regardless of whether those dump files were purged from disk:

```
TMF 7> DELETE DUMPS *.*.* !, SERIAL 1000
```

TMF places an event message in the EMS log and issues a warning message for any file associated with serial number 1000 that cannot be purged; you must then delete each of these files manually.

The next DELETE DUMPS command removes from the catalog the files belonging to dump serial number 1000 only if those dump files were first purged from disk:

```
TMF 8> DELETE DUMPS *.*.*, SERIAL 1000
```

If any file belonging to dump serial number 1000 is not purged, TMF displays an appropriate warning message and places an event message in the EMS log. However, because the exclamation point is not specified in this command, you can retry the delete operation when the problem has been corrected.

# DELETE RESOURCEMANAGER

The DELETE RESOURCEMANAGER command deletes one or more recoverable resource managers from the resource manager directory.

```
DELETE {RESOURCEMANAGER | RM} {rm-name
                                {(rm-name [ , rm-name ] . . . )}}
```

*rm-name*

is the name of the recoverable resource manager to be deleted from the directory. This name can contain a maximum of 128 characters; the first character must be a letter, and the others can be alphanumeric or any of the following special characters:

\$ \_ - ^ @ &

Names that begin with ZZ-VOLATILE- are not allowed in this parameter. Names containing adjoining hyphens (--) are not allowed, although names using multiple hyphens separated by one or more intervening characters are legal. (TMFCOM interprets two consecutive hyphens as the start of a comment.) Thus, the name "ALPHA--1" is not permitted, but "ALPHA-N-1" is allowed.

## Security Restrictions

You can issue the DELETE RESOURCEMANAGER command only if you are a member of the super user group.

## TMF State Requirement

Before you can enter the DELETE RESOURCEMANAGER command, TMF must be started.

## Usage Guidelines

---

**Note.** Resource managers, to which the DELETE RESOURCEMANAGER command applies, support heterogeneous transaction processing. Resource managers are used in products such as NonStop TUXEDO. Although TMFCOM and the TMFSERVE programmatic interface provide TMF commands and options for operating on resource managers, system managers who use them do so in the context of issues involving multiple software subsystems and inter-platform considerations. Complete discussion of heterogeneous transaction processing and the context in which resource-manager commands and options are used lies beyond the scope of the TMF manual library.

For more information about heterogeneous transaction processing, see the Open Group TRANSACTION PROCESSING Publications, available from X/Open Publications at the following Web location:

<http://www.opengroup.org/pubs/catalog/tp.htm>

These publications can also be ordered through many bookstores.

---

Entries for recoverable resource managers can be deleted from the resource manager directory either by a gateway process that performs the deletion programmatically, or by a user who enters the DELETE RESOURCEMANAGER command.

The resource manager you specify must not have any unresolved transaction branches outstanding; that is, it must be in the closed state. If the resource manager is in the open, recovering, or crashed state, the DELETE RESOURCEMANAGER command displays an error message. You can determine the current state of a resource manager by entering the [STATUS RESOURCEMANAGER](#) command.

- 
- △ **Caution.** Deleting a recoverable resource manager without first preventing gateway processes from opening it can result in errors, either in the gateway process or in the execution of the DELETE RESOURCEMANAGER command.
- 

## Examples

The following DELETE RESOURCEMANAGER command deletes a resource manager named TUX-A1\_UNIX@SUN from the resource manager directory:

```
TMF 10> DELETE RESOURCEMANAGER TUX-A1_UNIX@SUN
```

The next command removes a resource manager named TUX-A2\_NT@MICROSOFT from the directory:

```
TMF 11> DELETE RM TUX-A2_NT@MICROSOFT
```

# DELETE TAPEMEDIA

The DELETE TAPEMEDIA command removes tape-volume entries from the TMF catalog and also deletes the entries for dumps stored on the removed tape volumes.

```
DELETE TAPEMEDIA { tape-volume
                   { (tape-volume [ , tape-volume ] ... ) } }
```

*tape-volume*

identifies the tape volume whose entries are to be deleted from the catalog. For this parameter, use the name magnetically encoded on the tape volume.

## Security Restrictions

You can issue the DELETE TAPEMEDIA command only if you are a member of the super user group.

## TMF State Requirement

Before you can enter the DELETE TAPEMEDIA command, TMF must be started.

## Examples

The following DELETE TAPEMEDIA command removes the tape volume TMF004 from the catalog:

```
TMF 14> DELETE TAPEMEDIA TMF004
```

If this volume currently has ASSIGNED status (indicating that it contains valid dumps), the dump file entries associated with the volume in the catalog are also removed.

The next DELETE TAPEMEDIA command deletes the tape volumes TMF009, TMF010, and TMF012 from the catalog:

```
TMF 15> DELETE TAPEMEDIA (TMF009, TMF010, TMF012)
```



# DELETE TMF

The DELETE TMF command purges the entire TMF configuration, including volume and file recovery information for the database. Only tape media names are retained in the TMF catalog.

This command, entered only when TMF is stopped, produces an empty configuration with no audit trails and with default values for other configuration information. DELETE TMF is typically used to make major configuration changes.

`DELETE TMF`

## Security Restrictions

You can enter the DELETE TMF command only if you are logged on under the super ID; the super ID is the person with the user ID 255,255—in effect, the group manager of the super user group.

## TMF State Requirement

You cannot use the DELETE TMF command while TMF is running; if you have started TMF, you must stop it with the [STOP TMF](#) command before entering DELETE TMF.

## Usage Guidelines

The DELETE TMF command is not allowed in either command files or the parameter line of the TMFCOM command that starts a noninteractive session. This command must only be entered interactively—that is, in response to the TMFCOM interactive prompt (TMF *line-number*>).

---

△ **Caution.** Contact the Global Customer Support Center (GCSC) or your service provider before issuing this command.

Use extreme caution when entering the DELETE TMF command, because this command removes TMF's ability to recover database files. This command can leave the database in an inconsistent or damaged state, potentially affecting both local and remote nodes on which shared distributed transactions exist.

TMF is shipped from the factory with the \$AUDIT disk volume configured as the active-audit volume and the \$DSMSCM disk volume configured as a TMF data volume. If you use the DELETE TMF command, you must subsequently respecify an active-audit volume (\$AUDIT or some other volume, through the ADD AUDITTRAIL command), and a TMF data volume (\$DSMSCM or another volume, through the ADD DATAVOLS command); if you do not do so, you cannot restart TMF.

---

After a DELETE TMF command, the dump entries remain in the TMF catalog until the next [START TMF](#) command is entered.

Each time TMF is deleted and subsequently restarted, TMF updates a value in the *tm-flags* portion of the transaction-identifier assigned to each transaction. If the *tm-*

*flags* value is not zero, it appears for each transaction listed by the [STATUS TRANSACTIONS](#) command.

## Example

The following DELETE TMF command purges the TMF configuration. Immediately following this command, TMFCOM displays a warning and prompts you to confirm the delete operation:

```
TMF >17 DELETE TMF
```

```
WARNING:  After a DELETE TMF command, neither volume recovery  
          nor file recovery can recover your files.  Use  
          CAUTION when performing a DELETE TMF operation in a  
          crisis situation.
```

```
          If this is a CRISIS situation, you should contact  
          your support representative NOW.  Attempting a DELETE  
          TMF operation might produce broken files that are not  
          recoverable.
```

```
Do you confirm this DELETE TMF? (Y/N)  y  
TMF 18>
```

# DELETE TRANSACTION

The DELETE TRANSACTION command forcibly removes a transaction from the system.

```
DELETE TRANS[ACTION] transaction-id

where transaction-id is:

    [node .                               ]
    [node-number .                         ]
    [node(tm-flags) .                     ] cpu.seq-num
    [node-number(tm-flags) .             ]
    [ (tm-flags) .                       ]
```

*transaction-id*

identifies the transaction to be deleted:

*node*

is the name of the node (system) on which the transaction originated, which is known as the home node for the transaction. The syntax rules for specifying a node name appear under [Disk File Names](#), earlier in this section. The default is the node on which the TMFSERVE process communicating with your TMFCOM process is running.

*node-number*

is the number of the home node. A node number specification consists of a backslash followed by the number assigned to the node. The default is the node on which the TMFSERVE process communicating with your TMFCOM process is running.

*tm-flags*

is a number representing flags used internally by TMF. If this number is zero, it does not appear in displays of the *transaction-id* presented by the STATUS TRANSACTIONS command. For more information about *tm-flags*, see the [DELETE TMF](#) command discussion.

---

**Note.** If the *tm-flags* value is a number other than zero, you must include this number in parentheses in the ABORT TRANSACTION command. For example:

---

**~DELETE TRANSACTION \SYSB(1).3.45678**

---

*cpu*

is the number of the processor on which the transaction originated.

*seq-num*

is a sequence number assigned by TMF to the transaction, which distinguishes it from other transactions originating in the same processor on the same home node under the current TMF configuration.

## Security Restrictions

You can enter the DELETE TRANSACTION command only if you are logged on under the super ID. (The super ID is the person with the user ID 255,255—in effect, the group manager of the super user group.)

## TMF State Requirement

Before you can enter the DELETE TRANSACTION command, TMF must be started.

## Usage Guidelines

In certain situations, transactions might remain in the system after you expect their removal, resulting in resources such as audit trails, locks, or data structures being held for a long time. When the availability of these resources is more important than data integrity, consider forcibly deleting the transaction.

- 
- △ **Caution.** DELETE TRANSACTION is a very dangerous command, because it can result in a corrupted database. Use this command only with extreme caution. Alternative and safer methods for forcibly deleting a transaction are provided by the RESOLVE TRANSACTION and ABORT TRANSACTION commands, as noted elsewhere in this section of the manual.
- 

If a transaction is in the hung or committed state, you can force its deletion by using the DELETE TRANSACTION command. A hung transaction is one that TMF attempted to abort but could not; a committed transaction is one that successfully completed its changes to the database. (More about these and other transaction states appears in the [STATUS TRANSACTIONS](#) command description.) As an example, you can use a DELETE TRANSACTION command to remove a transaction that was hung by the backout process because of audit-reading errors. In fact, if an audit-reading error is permanent, you can only eliminate this transaction by using DELETE TRANSACTION to forcibly remove it.

Usually, you will not manually alter the states of any transactions as the result of a communications link failure. Transactions in the committed state cause no problems: they do not hold database locks and they do not retain (pin) audit-trail files on disk. The only reason you might want to consider deleting a transaction from the committed state is if the child node is destroyed and will never come back online.

## Example

The following DELETE TRANSACTION command deletes the transaction identified as \TSII.2.1958911:

```
TMF 19> DELETE TRANSACTION \TSII.2.1958911  
Confirm: y
```

# DISABLE AUDITDUMP

The DISABLE AUDITDUMP command suspends an audit dump process, preventing the dumping of audit-trail files.

```
DISABLE AUDITDUMP {MASTER | MAT }
                  {AUX[ ILIARY ] nn}
                  [ , ABRUPT ]
```

MASTER | MAT  
AUX[ ILIARY ] nn

specifies the audit trail associated with the audit dump process that is to be suspended, thus identifying that process. MASTER or MAT specifies the master audit trail. AUXILIARY<sub>nn</sub> or AUX<sub>nn</sub> specifies an auxiliary audit trail. *nn* is a two-digit number ranging from 01 to 15 that specifies one of the 15 potential auxiliary audit trails in the configuration.

ABRUPT

requests that, if an audit dump process is currently running, TMF abort that process and discard the dump it is preparing. If you omit this option and an audit dump process is running, that process completes the dump.

## Security Restrictions

You can issue the DISABLE AUDITDUMP command only if you are a member of the super user group.

## TMF State Requirement

You can enter the DISABLE AUDITDUMP command at any time, whether or not TMF has been started. However, the audit trail associated with the dump must be configured for dumping (in other words, its AUDITDUMP attribute must be set to ON).

## Usage Guidelines

The audit dump process identified in the DISABLE AUDITDUMP command remains disabled until you explicitly issue an [ENABLE AUDITDUMP](#) command. The process remains disabled even if you stop and restart TMF; you must issue [ENABLE AUDITDUMP](#) to resume the dump.

If you specify ABRUPT in the DISABLE AUDITDUMP command, the file whose dump is aborted is the first file to be dumped when dumping is reenabled.

The ABRUPT option offers a quick method for suspending an audit dump in progress. Without this option, you would need to take the following steps to suspend that dump:

1. Issue a [STATUS AUDITDUMP](#) command to display the name of the current audit dump process.
2. Issue a DISABLE AUDITDUMP command, identifying the process by its associated audit trail, to disable the process. (At this point, remember, the dump currently in progress still continues.)
3. Issue an [EXIT](#) command to terminate TMFCOM.
4. From TACL, enter a STOP command, using the audit dump process name, to abort the audit dump process.

If ABRUPT is not specified, audit dumps in progress are not affected by the DISABLE AUDITDUMP command. Of course, the audit dump process cannot dump any further audit-trail files after you issue this command.

---

△ **Caution.** Leaving audit dumps disabled for long durations can cause audit-trail files to accumulate. This accumulation, in turn, can potentially cause TMF to reach the OVERFLOWTHRESHOLD and BEGINTRANSDISABLED limits.

---

## Examples

The following DISABLE AUDITDUMP command disables the audit dump process associated with the master audit trail and immediately suspends dumping in progress:

```
TMF 21> DISABLE AUDITDUMP MAT, ABRUPT
```

The next DISABLE AUDITDUMP command disables the audit dump process associated with the auxiliary audit trail AUX02, but allows it to complete the current dump:

```
TMF 22> DISABLE AUDITDUMP AUX02
```

# DISABLE BEGINTRANS

The DISABLE BEGINTRANS command suspends transaction processing throughout the TMF system.

`DISABLE BEGINTRANS`

## Security Restrictions

You can enter the DISABLE BEGINTRANS command only if you are a member of the super user group.

## TMF State Requirement

Before you can enter the DISABLE BEGINTRANS command, TMF must be started.

## Usage Guidelines

The DISABLE BEGINTRANS command suspends transaction processing by requesting TMF to disallow new calls to the BEGINTRANSACTION system library procedure.

- 
- △ **Caution.** Suspending transaction processing can cause your transaction processing applications to fail; thus, you may have to stop your applications before issuing the DISABLE BEGINTRANS command. If you must suspend transaction processing, choose a time when the interruption will affect system users the least.
- 

Transaction processing remains disabled until you issue the [ENABLE BEGINTRANS](#) command. It remains disabled even if you stop and restart TMF; to reenabling transaction processing, you must explicitly issue [ENABLE BEGINTRANS](#).

If transaction processing is already disabled when you enter DISABLE BEGINTRANS, TMFCOM displays an error message.

The [START TMF](#) command also contains an option called DISABLE BEGINTRANS that disables transaction processing.

## Example

The following DISABLE BEGINTRANS command suspends transaction processing:

```
TMF 24> DISABLE BEGINTRANS
```



# DISABLE DATAVOLS

The DISABLE DATAVOLS command shuts down transaction processing (audited updates) for the specified data volume and places this volume in the disabled state so that it will not be automatically restarted by TMF.

```
DISABLE DATAVOL[S] {volume
                    {(volume [,volume]...)}}
                    [,ABRUPT]
```

*volume*

identifies a data volume for which transaction processing is to be disabled.

ABRUPT

specifies that TMF should immediately begin rejecting audited requests for the data volume specified, even those on behalf of existing transactions that have already updated this volume.

If ABRUPT is not specified, then transactions that have already acted on the volume may continue to update it, and TMF will finish disabling the volume after all such transactions have completed.

## Security Restrictions

You can issue a DISABLE DATAVOLS command only if you are a member of the super user group.

## TMF State Requirement

Before you can enter the DISABLE DATAVOLS command, TMF must be started.

## Usage Guidelines

The DISABLE DATAVOLS command is typically issued in preparation for deleting the volume from the TMF configuration or to get ready for hardware servicing.

After you enter the DISABLE DATAVOLS command, the resulting disabled data volume remains disabled and audited work on this volume is suspended until you explicitly enable the volume again with the [ENABLE DATAVOLS](#) command.

The DISABLE DATAVOLS command succeeds only if the volume specified is in a startup state (that is, startup begun, waiting for transaction resolution, queued for recovery, recovering, or startup completing) or in the started, disabling, or down state. Furthermore, if the volume is in the starting state, the command succeeds only if ABRUPT is specified. These and other data volume states are explained in the [STATUS DATAVOLS](#) command description.

If you want to disallow audited updates on a volume that is already being processed by the DISABLE DATAVOLS command (entered without the ABRUPT option), issue a second DISABLE DATAVOLS command with ABRUPT specified. This second command overrides the first and prevents further audited activity on the volume.

Much work on behalf of the DISABLE DATAVOLS command is done by the operating system's disk process. When the disk process successfully completes, in an orderly way, all work that should be done during a transaction processing shutdown, the volume specified enters the disabled-clean state. If this orderly shutdown does not occur, however, the volume is placed in the disabled-dirty state. Specifying the ABRUPT option is likely to place a volume in the disabled-dirty state, but this result is not guaranteed. Whether the state is disabled-clean or disabled-dirty, the result is the same: transaction processing is disabled for the specified volume. Regardless of whether the state is disabled-clean or disabled-dirty, volume recovery will return the volume to a consistent state after it is again enabled.

---

△ **Caution.** Leaving data volumes disabled for long durations can cause audit-trail files to accumulate. This accumulation, in turn, can potentially cause TMF to reach the OVERFLOWTHRESHOLD and BEGINTRANSDISABLE limits.

---

## Examples

The following DISABLE DATAVOLS command shuts down transaction processing for the volume named \$DATA2, but allows transactions in progress on the volume to complete:

```
TMF 27> DISABLE DATAVOLS $DATA2
      $DATA2 -- disabling datavol.
TMF 28>
```

The next DISABLE DATAVOLS command shuts down transaction processing on two data volumes, \$DATA3 and \$DATA4, and abruptly terminates any such processing now in progress on those volumes:

```
TMF 28> DISABLE DATAVOLS ($DATA3, $DATA4), ABRUPT
      $DATA3 -- disabling datavol.
      $DATA4 -- disabling datavol.
TMF 29>
```

# DISPLAY OPERATIONS

The DISPLAY OPERATIONS command displays the Event Management Service (EMS) messages recorded for a specific TMF operation.

```
DISPLAY OPERATION[S] operation-number

    [,LONG                               ]

    [,BEGINTIME date-time [GMT]]

    [,LOGFILE filename                 ]
    [,COLLECTOR process-name          ]
```

*operation-number*

is an integer that identifies the operation for which EMS messages are displayed, as described under “Usage Guidelines.” The lowest operation number assigned is 1, and the highest is 100.

LONG

displays EMS messages in the long form, including the time that the operation occurred, the name of the process that generated the operation, and the identity of the subsystem (such as TMF.G07) that produced the operation.

In the long form, message display lines often exceed the typical screen width of 80 characters and are therefore difficult to read. If you omit LONG, the default displays messages in the short form, which excludes the event time, process name, and subsystem identity.

BEGINTIME *date-time*

specifies the approximate time that the operation for which you want information was generated, as described under “Usage Guidelines.” Specify *date-time* in the following format:

{*month*[-]*day*[-]*year*, *hr*:*min*[:*sec*] }

{*day*[-]*month*[-]*year*, *hr*:*min*[:*sec*] }

where:

*month*

is a three-character abbreviation for the name of the month:

{JAN|FEB|MAR|APR|MAY|JUN|JUL|AUG|SEP|OCT|NOV|DEC}

*day*

is a two-digit integer for the day of the month.

*year*

is a four-digit integer for the calendar year.

---

**Note.** You can use either dashes (-) or spaces, or both, as delimiters to separate the *month*, *day*, and *year* values from one another, but you must use at least one of these delimiters.

---

*hr*

is an integer for the hour, from 0 through 23.

*min*

is an integer for the minute, from 00 through 59.

*sec*

is an integer for the second, from 00 through 59.

GMT

indicates that the BEGINTIME value is specified in Greenwich mean time. If GMT is omitted, the default is the local time at your site.

LOGFILE *filename*

identifies the EMS log file on which all EMS messages are recorded, entered in the following format:

[*volume.*][*subvolume.*]*file-id*

The default is the current EMS log file for TMF, used by the standard EMS collector process (named \$0).

COLLECTOR *process-name*

specifies the process name of the EMS collector that is the source of the EMS messages. The default is the standard EMS collector process (named \$0).

---

**Note.** You can specify either the LOGFILE or the COLLECTOR option, but not both.

---

## Security Restrictions

You can issue the DISPLAY OPERATIONS command only if you have read-access to the EMS log files.

## TMF State Requirement

You can enter the DISPLAY OPERATIONS command at any time, whether or not TMF has been started.

## Usage Guidelines

The DISPLAY OPERATIONS command returns information for one operation only.

TMF recognizes each specific instance of a particular kind of operation (such as a TMF start, a TMF stop, or a file recovery) as a unique operation and assigns an operation serial number that distinguishes this operation from other recent operations of the same type. If you use the NOWAIT option in the command that generates an operation, TMFCOM displays this number at your terminal. You can then retrieve EMS messages about the operation by entering this number in the *operation-number* parameter of the DISPLAY OPERATIONS command.

TMF maintains a pool of 100 operation numbers, assigning each new operation the next available number. TMF re-uses the numbers generated; after TMF assigns operation number 100, it next assigns the lowest number that is not presently in use. For example, if TMF has assigned operation number 100 and the number 1 is still in use when a new operation begins, TMF uses the number 2 for the new operation. By keeping the number of digits used for operation numbers low, TMF provides values that are easy to remember.

TMF keeps a history of recent operations, and you normally only need to include *operation-number* in the DISPLAY OPERATIONS command to uniquely identify the operation you want. However, if an operation occurred some time ago, the *operation-number* list might have wrapped-around to its beginning and the number of the operation you want to display might subsequently have been assigned to another operation as well. Use the BEGINTIME *date-time* option to distinguish between these multiple operations that share the same *operation-number*. The *date-time* of an operation, like the *operation-number*, is displayed when you use the NOWAIT option in the TMFCOM command that generates the operation.

If you specify an *operation-number* only, TMF displays the EMS messages associated with the most recent occurrence of *operation-number*. If the TMP no longer has a record of that *operation-number*, however, no data is displayed. If no data appears, take these steps to retrieve the information needed:

1. Enter a [STATUS OPERATIONS](#) command to determine whether the TMP has a record of the desired operation.
2. If this record exists, enter the DISPLAY OPERATIONS command, specifying the operation-number only. If the record does not exist, however, enter the DISPLAY OPERATIONS command with an approximate BEGINTIME value; in this case, TMF does not consult the TMP and simply starts its search at the specified time.

If you specify the BEGINTIME option, TMF displays the EMS messages associated with the next occurrence of *operation-number* after the time indicated by BEGINTIME *date-time*.

If you specify the LOGFILE option, TMF searches for *operation-number* in the log file indicated by LOGFILE *filename* rather than the current log file.

If you enter the DISPLAY OPERATIONS command but do not have read-access to the EMS log files, TMFCOM displays Error Messages 951, 952, and 953 and aborts the command. To correct this situation, you must either log on with a super user group ID which automatically grants you read-access to the EMS log files, or have the EMS log files secured to permit you read-access.

## Examples

The following DISPLAY OPERATIONS command presents, in the short form, the EMS messages relating to the operation identified by *operation-number* 50:

```
TMF 30> DISPLAY OPERATIONS 50
Event Messages:
NonStop TMF on \TSII *0155* DUMP FILES [50] TMP Process: Begin operation.
NonStop TMF on \TSII *0217* DUMP FILES [50] OnlineDump Process #1
OnlineDumpMgmt: Dump serial: 2108 File-set-list:
$ALTVOL.*.*, $BLAST.*.*, $SIGN.*.*, $SPOOL.*.*, $MAIL03.*.*, $MAIL04.*.*,
$MAIL06.*.*, $MAIL09.*.*
NonStop TMF on \TSII *0218* DUMP FILES [50] OnlineDump Process #1
OnlineDumpMgmt: Options: MEDIUM TAPE, TAPEMEDIA ()
NonStop TMF on \TSII *0219* DUMP FILES [50] OnlineDump Process #1
OnlineDumpMgmt: Options: BLOCKSIZE 28, COPIES 1, SYSTEM (\tess), VERIFYTAPE
OFF
NonStop TMF on \TSII *0222* DUMP FILES [50] OnlineDump Process #1
OnlineDumpMgmt: *First tape* Tape #1, volume ID: T01373, drive \TESS.$SILO0
NonStop TMF on \TSII *0227* DUMP FILES [50] OnlineDump Process #1
OnlineDumpMgmt: $ALTVOL.FOO.DICTALT Initial EOF: 12288, End Time:
3 Aug 2002, 18:22:27, Master: 599, Data: 599, Type: K
NonStop TMF on \TSII *0227* DUMP FILES [50] OnlineDump Process #1
OnlineDumpMgmt: $ALTVOL.FOO.DICTCDF Initial EOF: 0, End Time:
3 Aug 2002, 18:22:27, Master: 599, Data: 599, Type: K
.
.
.
NonStop TMF on \TSII *0227* DUMP FILES [50] OnlineDump Process #1
OnlineDumpMgmt: $SPOOL.USER.VERSIONS Initial EOF: 1536, End Time:
4 Aug 2002, 6:14:44, Master: 599, Data: 599, Type: K Ta
NonStop TMF on \TSII *0227* DUMP FILES [50] OnlineDump Process #1
OnlineDumpMgmt: $SPOOL.USER.VIEWS Initial EOF: 0, End Time:
4 Aug 2002, 6:14:45, Master: 599, Data: 599, Type: K Ta
NonStop TMF on \TSII *0223* DUMP FILES [50] OnlineDump Process #1
OnlineDumpMgmt: Tape volume IDs used : T01373, T01581, T01605, T01625, T01637
NonStop TMF on \TSII *0211* DUMP FILES [50] OnlineDump Process #1
OnlineDumpMgmt: Completed Online Dump to tape. Dump serial number: 2108.
NonStop TMF on \TSII *0228* DUMP FILES [50] OnlineDump Process #1
OnlineDumpMgmt: Files dumped = 473 Files not dumped = 0
NonStop TMF on \TSII *0156* DUMP FILES [50] TMP Process: End operation.
TMF 31>
```

# DUMP FILES

The DUMP FILES command copies audited files to either tape or disk, producing online dumps. These online dumps might be used later if file recovery is initiated through a [RECOVER FILES](#) command.

```
DUMP FILES {file-set
            {(file-set [,file-set]...)}

    [,BLOCKSIZE { 4 | 8 | 12 | 16 | 20 | 24 | 28 | 32 | 36
                  40 | 44 | 48 | 52}]

    [,COPIES { 1 | 2 } [PARALLEL]

    [,DISKMEDIA {disk
                 {(disk [,disk]...)}

    [,NOT {file-set
           {(file-set [,file-set]...)}

    [,START start-file

    [,TAPEMEDIA {tape-volume
                 {(tape-volume [,tape-volume]...)}

    [,SYSTEM {node
              {(node, node)}

    [,UNLOAD { ON | OFF }

    [,VERIFYTAPE { ON | OFF }
```

## *file-set*

identifies the audited files to be dumped to tape or disk. You must specifically name alternate-key files and partitions of partitioned files in *file-set* if you want them to be dumped. Nonaudited files are not dumped. If you specify the fully qualified file name of a nonaudited file, an error message stating the file was not dumped is displayed. Enter *file-set* in the following format:

```
[ [volume.] subvolume.] file-id
```

---

**Note.** If you plan to refer to SQL/MX objects in a *file-set* list, you must use the Guardian names of the underlying files in all TMFCOM commands. You can run the MXGNAMES utility to convert one or more objects' ANSI names to their underlying Guardian file names. You can then use the Guardian file names in the *file-set* list. For convenience, MXGNAMES provides an option that generates names formatted as TMF file-sets for direct use in TMFCOM commands. For information about using MXGNAMES, see the *SQL/MX Reference Manual*.

---

In the *file-set* list, file names are expanded with default volume and subvolume names, and the space required for them is impacted by other overhead

as well. All *file-set* information cannot exceed the size of the SPI message buffer that supports communication between TMFCOM and TMFSERVE, which is 28 KB. If this limit is exceeded, TMFCOM displays Error Message 1050.

You can specify a *file-set* as a file-name pattern. This pattern resembles a file name but designates a set of entities (that is, a set of disk volumes, subvolumes, or files) through the use of pattern-matching (wild-card) characters. The wild-card characters are:

- \* An asterisk matches zero or more letters, digits, or a combination of these.
- ? A question mark matches exactly one letter or digit.

The file-name pattern syntax is:

```
[$pattern.]pattern.pattern
```

In this syntax, *pattern* consists of one or more characters. Allowable characters are letters, digits, asterisks (\*), and question marks (?). The maximum length of a pattern is 8 characters, including wild-card characters. Wild-card characters can appear in any portion of a name, for as many times as there can be characters in that portion.

Examples:

- \* matches all files in the current subvolume
- \*z\* matches all files in the current subvolume that have IDs containing the letter "z."
- \$DISK?? matches all disk volumes at the current node that have seven-character names starting with "\$DISK," such as \$DISK01" and "\$DISK26."
- P\*.\* matches all disk files on the current volume that are in subvolumes whose names begin with the letter "P."

```
BLOCKSIZE { 4 | 8 | 12 | 16 | 20 | 24 | 28 | 32 | 36
              | 40 | 44 | 48 | 52 }
```

for dumps to tape, sets the tape data block length, when multiplied by 1024 bytes. It is more efficient to use large data block lengths, but this is limited by the density in bits per inch (bpi) at which the tape drive writes tapes. The default is 8.

The following are recommended values for BLOCKSIZE:

|               |     |      |      |
|---------------|-----|------|------|
| Density (bpi) | 800 | 1600 | 6250 |
| BLOCKSIZE     | 4   | 8    | 28   |

If you specify a BLOCKSIZE value larger than 28, but the tape drive selected cannot handle data blocks this long, TMF uses a block size of 28 KB for the dump and generates a warning message.



COPIES { 1 | 2 } [PARALLEL]

for dumps to tape, specifies the number of copies made of each online dump. If you set COPIES to 1, only one tape drive is required and the copy is directed to that drive. If you set COPIES to 2, two tape drives are required and the copies are always produced in parallel, not in serial order, regardless of whether you enter the PARALLEL option. When producing two copies:

- If you specify only one node name in the SYSTEM parameter, TMF produces the two copies in parallel, using two tape drives on that node.
- If you specify two node names in the SYSTEM parameter, TMF produces the two copies in parallel, using one tape drive on each node.

COPIES 2 is recommended in case one tape cannot be read because of a tape media failure or if each copy of the dump requires multiple tape volumes.

The default for this parameter is COPIES 1.

#### DISKMEDIA

for dumps to disk, identifies the disk volume on which the online dumps are recorded. At least one *disk* must be specified. The maximum number of disks allowed is 16, each specified in any of the following formats:

```
remote-node.volume.prefix
local-node.volume.prefix
local-node.volume
volume
volume.prefix
```

*prefix* is a sequence of two alphabetic characters, to which TMF appends a 6-digit integer to name the online dump.

If you specify a list of disks, TMF tries to send the online dump to the first disk listed. If that disk cannot be accessed or is out of space, TMF tries the next disk, and so forth.

The optional and required syntax items for *disk* and the naming conventions for *prefix* depend on whether the specified *disk* is on a remote or local node.

When the disk resides on a remote node:

- The name specified for *disk* must be fully qualified.
- *volume* must not exceed seven characters, counting the dollar sign (\$).
- You cannot specify ZT as the prefix.

When the disk resides on a local node:

- *node* and *prefix* are optional.
- If *node* is not specified, the default is the node on which your TMSERVE process is running.

- Any value for *prefix* can be specified.
- If *prefix* is not specified, the local node default ZT is automatically assigned as the prefix.

Optical disks are not supported for dump to disk operations.

If you specify DISKMEDIA, you must omit the following options: BLOCKSIZE, COPIES, SYSTEM, TAPEMEDIA, and VERIFYTAPE.

#### NOT *file-set*

identifies one or more files that are not to be dumped. If you specify the same file in this option and in *file-set*, the file is not dumped. The NOT *file-set* list allows the same pattern-matching (wild-card) characters described earlier in the discussion of the *file-set* option for this command. The NOT *file-set* information cannot exceed the size of the SPI message buffer that supports communication between TMFCOM and TMFSERVE, which is 28 KB. If this limit is exceeded, TMFCOM displays Error Message 1050.

---

**Note.** If you plan to refer to SQL/MX objects in a *file-set* list or a NOT *file-set* list, you must use the Guardian names of the underlying files in all TMFCOM commands. You can run the MXGNAMES utility to convert one or more objects' ANSI names to their underlying Guardian file names. You can then use the Guardian file names in the *file-set* list or NOT *file-set* list. For convenience, MXGNAMES provides an option that generates names formatted as TMF file-sets for direct use in TMFCOM commands. For information about using MXGNAMES, see the *SQL/MX Reference Manual*.

---

#### START *start-file*

indicates the file at which the online dump begins in the specified volume and subvolume. TMF dumps this file and all subsequent ones (based on alphabetical ordering) in this volume and subvolume. Enter *start-file* in the following format:

```
{[[volume.]subvolume.]{file-id | *}}
{[volume.]*.*}
```

You can use an asterisk (\*) in the *file-id* or *subvolume* field of *start-file*, but not in the *volume* field. If you specify an asterisk in the *file-id* field, the dump starts at the beginning of the subvolume. If you specify an asterisk in the *subvolume* field, the dump starts at the beginning of the volume. To avoid dumping the entire database, you should specify *start-file* as a fully qualified file name.

If you do not specify the volume and subvolume in *start-file*, the current default volume and subvolume are used. If the resulting file cannot be found, the START parameter is ignored and TMF dumps the entire file set.

**TAPEMEDIA** *tape-volume*

for dumps to tape, identifies the tape volume to which the online dumps are written. *tape-volume* is the identifier magnetically encoded in the tape volume label and registered in the TMF catalog. The maximum number of tape volumes allowed by this command is 16.

If the TAPEMEDIA parameter is specified in the DUMP FILES command, TMF attempts to use the tape volume specified by this parameter, as described under [Online Dumps to Tape](#) later in this discussion.

If no medium is specified (that is, if neither TAPEMEDIA nor DISKMEDIA is specified), tape media is assumed by default and a scratch tape is selected from among those listed in the TMF catalog.

If you specify TAPEMEDIA, you must not also specify the DISKMEDIA option.

**SYSTEM** *node*

specifies the node (system) or nodes on which all copies are made. The SYSTEM parameter applies only to dumps directed to tape. (When directing a dump to a disk volume on a remote node, include the node name within the disk file name.) The syntax rules for specifying a node name appear under [Disk File Names](#), earlier in this section. The default is the local node.

If you specify a remote node for dumps to tape, Guardian labeled-tape processing must be enabled on that node and must be running on the NonStop operating system. TMF, however, need not be configured or running on the remote node.

**UNLOAD** {ON | OFF}

for dumps to tape, determines whether the tape volume is unloaded after the dump process finishes writing to the tape. ON unloads the tape volume. OFF suppresses unloading. If the UNLOAD option is completely omitted, the default is OFF.

**VERIFYTAPE** {ON | OFF}

for dumps to tape, determines whether tape verification is performed after a tape is written. ON performs tape verification. After each tape volume is written, it is rewound, read completely, and finally rewound and unloaded. The tape-record checksums, tape-record sequence numbers, and tape format are verified. The checks are the same as those made during file recovery; however, tape verification does not include comparing the tape data to the audited files on disk.

If tape verification fails, an error message appears, the online dump terminates, and the online dump copy is marked as invalid in the catalog. Tape verification approximately doubles the time needed for a dump.

OFF suppresses verification of tape volumes.

If the VERIFYTAPE option is completely omitted, the default is VERIFYTAPE OFF.

## Security Restrictions

You can issue the DUMP FILES command only if you are a member of the super user group.

## TMF State Requirement

Before you can enter the DUMP FILES command, TMF must be started.

## Usage Guidelines

The DUMP FILES command dumps the specified audited files to disk if the DISKMEDIA option is specified; otherwise, this command dumps them to tape.

---

**Note.** For both online dumps and audit dumps, Expand links to remote disk volumes must support 56 KB input/output.

---

The more frequently you make online dumps, the less time file recovery requires.

Either of the following strategies, or some combination of both, effectively protects your databases:

- To protect specifically against media failures, request online dumps by volume.
- To protect specifically against damage to a selected application, request online dumps of the database that belongs to that application.

If an online dump fails before all of the specified audited files are dumped, the amount of the dump that has completed successfully is saved. In most cases, the TMP automatically determines the last file that was successfully dumped and implicitly reissues the DUMP FILES command with the START option specifying the file at which the dump is to resume. In other cases, however, you might have to perform this restart yourself, as follows:

- Use the [INFO DUMPS](#) command to determine the last file that was successfully dumped.
- Issue a new DUMP FILES command with the START option, specifying the file on which the dump is to resume.

To avoid dumping your entire database by mistake, use a fully qualified file name for *start-file* in the START parameter of the DUMP FILES command.

- 
- △ **Caution.** Error messages generated by TMF and other subsystems during online dump operations can sometimes have a vital impact that is not always obvious. For example, a disk or tape parity or checksum error can prevent a file from being dumped. For this reason, you should check the individual messages and summary information displayed by the DUMP FILES command to ensure that you are aware of all such errors. In addition, you can use the [INFO DUMPS](#) command to verify that all the intended files were dumped.
-

## Online Dumps to Disk

Online dumps can span disk volumes. These dumps are recorded in large files, called “disk dump files,” on the disks that are specified with the DISKMEDIA option. There must be enough cumulative space available on all of the specified disks to hold the online dumps.

For maximum protection, store your online dumps on disks other than those containing the files being dumped.

Store different generations of online dumps on different sets of disks.

When the disk specified with the DISKMEDIA option is on a remote node, *volume* must not exceed seven characters, counting the leading dollar sign (\$).

If online dumps are recorded on nonmirrored disks, the RETAINDEPTH option of the [ALTER CATALOG](#) command should be set greater than 1 so that file recovery is still possible even if one of the disks cannot be read because of a media failure.

An online dump to disk fails if the SAFEGUARD subsystem’s DEFAULT-PROTECTION option is configured for your user ID. To avoid this situation, request the online dump under a user ID for which DEFAULT-PROTECTION is not configured.

## Online Dumps to Tape

All tapes used in a dump operation must be labeled tapes defined in the TMF catalog. Each tape is identified by the name previously assigned in the *tape-volume* parameter of the [ADD TAPEMEDIA](#) command, which inserts the name into the magnetically encoded label on the tape volume and enters this name for the volume in the catalog. For example, if tmf011 is specified in the *tape-volume* parameter, tmf011 is written in the label and recorded in the catalog as the tape-volume name.

In the DUMP FILES command, the TAPEMEDIA parameter allows you to specify, by volume name, a particular tape or set of tapes to be used as the dump destination. (For instance, you can specify the volume named tmf011 for this purpose.)

- If the TAPEMEDIA parameter is included in the DUMP FILES command, the following actions take place:
  - If the tape volume specified in TAPEMEDIA is presently mounted, TMF uses this tape for the dump.
  - If the tape volume specified in TAPEMEDIA is not presently mounted, the tape-handling software determines whether any other TMF scratch tape volumes defined in the TMF catalog are mounted. If a scratch tape volume is found, TMF uses that volume. However, if no TMF scratch tape volumes are mounted, TMF requests the operator to mount the volume specified in TAPEMEDIA; if the operator responds by mounting that volume or any other volume defined as a scratch tape in the TMF catalog, the volume mounted will be used.

- If no dump medium is specified (that is, if neither TAPEMEDIA nor DISKMEDIA is included) in the DUMP FILES command, tape media is assumed by default; TMF searches the TMF catalog for the first tape volume listed with SCRATCH status and requests the operator to mount that volume. The operator can respond by mounting that volume, or can select another TMF scratch tape volume defined in the TMF catalog.

If you specify TAPEMEDIA in the DUMP FILES command, you must not also specify the DISKMEDIA option.

The more frequently you make online dumps, the fewer the number of tape volumes needed to store audit dumps.

Specifying two copies of an online dump is recommended for dumping to tape. By using the SYSTEM (*node, node*) and COPIES 2 PARALLEL parameter constructs, you can produce parallel dumps on two nodes (one copy per node). Both nodes can be remote nodes, or one can be a remote node and the other your local node. All reels of a specific copy (Copy 1 or Copy 2), however, must be made on the same node.

If you specify COPIES 1, specifying RETAINDEPTH 3 (or higher) in the ALTER CATALOG command is recommended. If you specify COPIES 2, specifying RETAINDEPTH 2 (or higher) in the [ALTER CATALOG](#) command is recommended.

Even if none of the copies of the online dump can be read, you can use an earlier dump generation for file recovery.

If the first tape mounted in response to a tape-mount request is write-protected, TMF does not invoke the automatic volume-recognition (AVR) feature with successive tapes. This is particularly troublesome when multiple TMF scratch tapes have been pre-mounted on a multi-tape device; in such a case, all remaining scratch tapes will be rejected and unloaded. Always verify that pre-mounted TMF scratch tapes are not write-protected.

If TMF encounters an error when writing a dump to tape, the tape's status in the TMF catalog is changed from SCRATCH to RELEASED regardless of whether the RELEASED parameter for the catalog is set to ON or OFF. RELEASED status indicates that something may be wrong with the tape and helps prevent operators from repeatedly mounting a bad tape.

## Output Displayed

When the DUMP FILES command is executed, TMFCOM displays an online report that includes the items listed in [Table 3-3](#).

**Table 3-3. DUMP FILES Output**

| Heading or Item                             | Description  |
|---|--|
| Dump Files operation [ <i>nn</i> ]<br>begun | Notification that the dump operation has been started; <i>nn</i> is the operation serial number that distinguishes this operation from other recent operations of the same type.   |
| Operation TimeStamp                         | The date and time that the operation began.  |
| Event Messages                              | One or more EMS event messages indicating the progress of the dump operation, including for each file: the time and date the dump began, the process performing the dump, the name of the file dumped, the initial EOF for the file, the date and time the dump ended (End Time), the master audit trail sequence number of the dump (Master), the auxiliary audit trail sequence number of the dump (Data), and the type of file dumped (Type). |

## Examples

The following DUMP FILES command dumps the files in the subvolume \$DATA3.ZTMFDATA to the volume \$DATA4 on the local system \COMMIT, assigning YC as the prefix in the online dump name:

```
TMF 35> DUMP FILES $DATA3.ZTMFDATA.*, DISKMEDIA $DATA4.YC
```

```
Dump Files operation [1] begun.
Operation TimeStamp: 20-Jan-2002 17:01:54
Event Messages:
17:01 20JAN02 243,00,049 TMF *0217* DUMP FILES [1] OnlineDump Process #1
OnlineDumpMgmt: Dump serial: 65 File-set-list: $DATA3.ZTMFDATA.*
17:01 20JAN02 243,00,049 TMF *0218* DUMP FILES [1] OnlineDump Process #1
OnlineDumpMgmt: Options: MEDIUM DISK, DISKMEDIA (\COMMIT.$DATA4.YC)
17:02 20JAN02 243,00,049 TMF *0224* DUMP FILES [1] OnlineDump Process #1
OnlineDumpMgmt: Begin dumping to disk dump file #1
17:02 20JAN02 243,00,049 TMF *0227* DUMP FILES [1] OnlineDump Process #1
OnlineDumpMgmt: $DATA3.ZTMFDATA.CUSTOMER Initial EOF: 0, End Time:
20 JAN 2002, 17:02:00G, Master: 1, Data: 1, Type: K
17:02 20JAN02 243,00,049 TMF *0227* DUMP FILES [1] OnlineDump Process #1
OnlineDumpMgmt: $DATA3.ZTMFDATA.MASTER Initial EOF: 20480, End Time:
20 JAN 2002, 17:02:01G, Master: 1, Data: 1, Type: K
17:02 20JAN02 243,00,049 TMF *0227* DUMP FILES [1] OnlineDump Process #1
OnlineDumpMgmt: $DATA3.ZTMFDATA.REC1K000 Initial EOF: 0, End Time:
20 JAN 2002, 17:02:01G, Master: 1, Data: 1, Type: K
17:02 20JAN02 243,00,049 TMF *0227* DUMP FILES [1] OnlineDump Process #1
OnlineDumpMgmt: $DATA3.ZTMFDATA.REC1K001 Initial EOF: 0, End Time:
20 JAN 2002, 17:02:02G, Master: 1, Data: 1, Type: K
17:02 20JAN02 243,00,049 TMF *0227* DUMP FILES [1] OnlineDump Process #1
OnlineDumpMgmt: $DATA3.ZTMFDATA.REC1K002 Initial EOF: 0, End Time:
20 JAN 2002, 17:02:02G, Master: 1, Data: 1, Type: K
.
.
17:02 20JAN02 243,00,049 TMF *0226* DUMP FILES [1] OnLineDump Process #1
OnlineDumpMgmt: *Finished* Disk dump file #1 ($DATA4.YC000065.F0000001)
17:02 20JAN02 243,00,049 TMF *0212* DUMP FILES [1] OnLineDump Process #1
OnlineDumpMgmt: Completed Online Dump to disk. Dump serial number: 65.
17:02 20JAN02 243,00,049 TMF *0228* DUMP FILES [1] OnLineDump Process #1
OnlineDumpMgmt: Files dumps = 32 Files not dumped = 0
```

17:02 20JAN02 243,00,049 TMF \*0156\* DUMP FILES [1] TMP Process: End operation.

The next DUMP FILES command dumps all audited files residing on the disk file \$DATA to tape. Because the TAPEMEDIA option is not specified in this command, TMFCOM prompts the operator to mount scratch tapes for the online dump to tape operation:

TMF 36> **DUMP FILES \$DATA.\*.\***

The following DUMP FILES command includes the START option to request dumping files from the disk volume \$DATA, starting with those files in the subvolume \$DATA.MELLOW and any files in subvolumes following \$DATA.MELLOW:

TMF 37> **DUMP FILES \$DATA.\*.\*, START \$DATA.MELLOW.\***

The next DUMP FILES command dumps the file named F4 and all subsequent files in the subvolume \$VA.S2, and all files in the subvolume \$VA.S3. The files that precede F4 in \$VA.S2 are not dumped.

TMF 37> **DUMP FILES (\$VA.S2.\*, \$VA.S3.\*), START \$VA.S2.F4**

(If the parameter list in the above example had specified a subvolume that preceded \$VA.S2 alphabetically, that subvolume would be ignored and dumping would still begin with file F4 in subvolume \$VA.S2.)

The next DUMP FILES command dumps all audited files residing on the local data volume \$DATA07 to a magnetic tape drive on the remote system \NEWARK:

TMF 38> **DUMP FILES \$DATA07.\*.\*, SYSTEM \NEWARK**

The following DUMP FILES command directs two copies of all audited files residing on the data volume \$DATA12 to magnetic tape drives on the remote system \LOOP, specifies a blocksize of 20, and requests tape verification:

TMF 39> **DUMP FILES \$DATA12.\*.\*, SYSTEM \LOOP, COPIES 2 &  
>>>PARALLEL, BLOCKSIZE 20, VERIFYTAPE ON**

The DUMP FILE command shown next directs a copy of all audited files on the data volume \$DATA03 to the disk volume \$ODUMP:

TMF 40> **DUMP FILES \$DATA03.\*.\*, DISKMEDIA \$ODUMP**

In the next DUMP FILES command, all audited files residing on the disk volume \$LADY are dumped to disk. The disks specified in the DISKMEDIA list are \$TRAMP and \$SQLOFOX on the system \CAL. When \$TRAMP becomes full, the remainder of the dump is directed to \$SQLOFOX.

TMF 41> **DUMP FILES \$LADY.\*.\*, DISKMEDIA (\CAL.\$TRAMP,  
\CAL.\$SQLOFOX)**



# ENABLE AUDITDUMP

The ENABLE AUDITDUMP command permits an audit dump process to resume execution after it has been suspended by the [DISABLE AUDITDUMP](#) command, allowing the process to dump audit-trail files once again.

|   |
|---|
| <pre>ENABLE AUDITDUMP {MASTER   MAT }<br/>                  {AUX[ ILIARY ] nn }</pre> |
|---|

MASTER | MAT  
AUX[ ILIARY ] nn

specifies the audit trail associated with the audit dump process to resume execution, thus identifying that process. MASTER or MAT specifies the master audit trail. AUXILIARY<sub>nn</sub> or AUX<sub>nn</sub> specifies an auxiliary audit trail. *nn* is a two-digit number ranging from 01 to 15 that specifies one of the 15 potential auxiliary audit trails in the configuration.

## Security Restrictions

You can issue the ENABLE AUDITDUMP command only if you are a member of the super user group.

## TMF State Requirement

You can enter the ENABLE AUDITDUMP command at any time, whether or not TMF has been started. However, the audit trail associated with the dump must be configured for dumping (in other words, its AUDITDUMP attribute must be set to ON).

## Examples

The following ENABLE AUDITDUMP command reenables the audit dump process associated with the master audit trail, and that process resumes dumping:

```
TMF 61> ENABLE AUDITDUMP MAT
```

The next ENABLE AUDITDUMP command directs the audit dump process associated with the auxiliary audit trail AUX02 to resume dumping:

```
TMF 62> ENABLE AUDITDUMP AUX02
```

# ENABLE BEGINTRANS

The ENABLE BEGINTRANS command resumes transaction processing after it has been suspended by the DISABLE BEGINTRANS command.

`ENABLE BEGINTRANS`

## Security Restrictions

You can enter the ENABLE BEGINTRANS command only if you are a member of the super user group.

## TMF State Requirement

Before you can enter the ENABLE BEGINTRANS command, TMF must be started.

## Usage Guidelines

The ENABLE BEGINTRANS command resumes transaction processing by requesting TMF to again allow calls to the BEGINTRANSACTION system library procedure.

Although transaction processing is sometimes explicitly suspended through the [DISABLE BEGINTRANS](#) command, this suspension can also result from other events:

- The active capacity of one or more audit trails reaches its configured threshold.
- The number of concurrent transactions in the system exceeds its configured threshold.
- Memory available for the Transaction Manager library on one or more processors runs low.
- Memory available for either TMP process runs low.

If any of these conditions exist when you enter the ENABLE BEGINTRANS command, TMF generates an EMS event message. Before the ENABLE BEGINTRANS command takes effect, these conditions must all be corrected.

If transaction processing is not disabled when you enter ENABLE BEGINTRANS, TMFCOM displays an error message.

## Example

The following ENABLE BEGINTRANS command resumes transaction processing:

```
TMF 65> ENABLE BEGINTRANS
```

# ENABLE DATAVOLS

The ENABLE DATAVOLS command reenables transaction processing (audited updates) for the specified data volumes, and makes them subject to automatic restart by TMF. If the data volumes specified are accessible, they are restarted immediately.

```
ENABLE DATAVOL[S] { volume
                     { (volume [, volume] , ...)
                     * [, NOT volume]
                     * [, NOT (volume [, volume] ... ) ] }
```

*volume*

identifies a disabled data volume for which transaction processing is to be reenabled.

\*

reenables transaction processing for all disabled data volumes in the TMF configuration. Exceptions are the volumes listed in the NOT clause, if any.

NOT *volume*

used with the asterisk (\*) indicator to specify one or more disk volumes for which transaction processing is not to be reenabled; in other words, this option excludes specific volumes from being reenabled.

## Security Restrictions

You can issue an ENABLE DATAVOLS command only if you are a member of the super user group.

## TMF State Requirement

Before you can enter the ENABLE DATAVOLS command, TMF must be started.

## Usage Guidelines

The ENABLE DATAVOLS command places one or more data volumes in a state where they can be automatically restarted and recovered, if they can be accessed by TMF.

The ENABLE DATAVOLS command succeeds only if the volumes listed are in the disabled state; that is, they must be either disabled-clean or disabled-dirty.

If the disk volume is currently up when the ENABLE DATAVOLS command is issued, TMF immediately restarts it. If the volume is inaccessible when the ENABLE DATAVOLS command is issued, TMF records that it is down and restarts it immediately the next time it becomes accessible.

## Examples

The following ENABLE DATAVOLS command reenables transaction processing for the volume named \$DATA2:

```
TMF 67> ENABLE DATAVOLS $DATA2  
$DATA2 -- enabled.  
TMF 68>
```

The next ENABLE DATAVOLS command reenables transaction processing on all disabled data volumes except \$DATA4 and \$DATA5:

```
TMF 68> ENABLE DATAVOLS *, NOT ($DATA4, $DATA5)  
$DATA1 -- enabled.  
$DATA3 -- enabled.  
$DATA6 -- enabled.  
$DATA7 -- enabled.  
.  
.  
.  
TMF 69>
```

# ENV

The ENV command displays the current operating environment for your TMFCOM session.

ENV

## Security Restrictions

Anyone can enter the ENV command.

## TMF State Requirement

You can enter the ENV command at any time, even if TMF has been started.

## Usage Guidelines

Unqualified file names that appear in TMFCOM commands are expanded according to the default node, volume, and subvolumes currently in effect. In TMFCOM, you can alter these defaults through the [VOLUME](#), [CMDVOL](#), and [OBEYVOL](#) commands. You can also specify a different input file for your TMFCOM commands and an alternate destination file for your TMFCOM output through the IN and OUT parameters, respectively, in the TMFCOM command that begins your session. In addition, you can change the TMFSERVE process with which your TMFCOM session communicates through the [OPEN SERVER](#) command.

By using the ENV command, you can display the current settings for all of these defaults.

If a NonStop node (system) name has been explicitly specified in the most recent [VOLUME](#), [CMDVOL](#), or [OBEYVOL](#) command, the ENV command also displays this node name.

## Output Displayed

The ENV command presents the current default settings for your session environment. The items in the display are listed in [Table 3-4](#).

---

**Table 3-4. ENV Output** (page 1 of 2)

| Heading or Item | Description   |
|-----------------|---|
| Default Volume  | Default volume assigned at the start of this TMFCOM session.        |
| Volume          | Default volume assigned by <a href="#">VOLUME</a> command, if any.  |
| Command Volume  | Default volume assigned by <a href="#">CMDVOL</a> command, if any.  |
| Obey Volume     | Default volume assigned by <a href="#">OBEYVOL</a> command, if any. |

---

**Table 3-4. ENV Output** (page 2 of 2)

| Heading or Item | Description  |
|-----------------|--|
| In File         | File (device) used for TMFCOM input (typically, your workstation or terminal). |
| Out File        | File (device) used for TMFCOM output (typically, your workstation or terminal) |
| Server Process  | Name of TMFSERVE process with which your TMFCOM is communicating.              |

## Example

The ENV command displays the TMFCOM operating environment for the current session, including the default volume and subvolume names (\$TMF10 and BRTADA1) as specified in the preceding VOLUME command:

```

TMF 71> VOLUME $TMF10.BRTADA1
TMF 72> ENV
      Default Volume      \FOXII.$TMF10.DMTADA0
      Volume              $TMF10.BRTADA1
      Command Volume      None Specified
      Obey Volume         None Specified
      In File              \FOXII.$ZTN1A.#PTPWJD6
      Out File             \FOXII.$ZTN1A.#PTPWJD6
      Server Process       \NEPAL.$Z1KR
TMF 73>

```

# EXIT

The EXIT command ends your current TMFCOM session.

`EXIT`

## Security Restrictions

Anyone can enter the EXIT command.

## TMF State Requirement

You can enter the EXIT command at any time, whether or not TMF has been started.

## Usage Guidelines

If you enter the EXIT command in your current TMFCOM session, TMFCOM terminates the session and returns control to the operating system.

If the EXIT command appears in a command file, TMFCOM stops reading the command file and ignores any commands in that file that follow the EXIT command. Furthermore:

- If the command file name was supplied in the IN option of the command that runs TMFCOM, TMFCOM passes control to the operating system.
- If the command file name was supplied in an OBEY command issued during an interactive TMFCOM session, TMFCOM resumes control and prompts you for a new command.

You can also end the current TMFCOM session by pressing the Control and Y keys at the same time (Ctrl/Y), which is equivalent to issuing the EXIT command.

## Example

When the EXIT command is issued in an interactive session, control of the terminal returns to TACL:

```
TMF 50> EXIT
2>
```

# FC

The FC command permits you to display and repeat or modify a command previously entered in the current TMFCOM session, as recorded in the TMFCOM command history buffer:

|   |
|---|
| <code>FC [-linenum   linenum   string   "quoted-string"]</code> |
|---|

*-linenum*

identifies the command to be displayed relative to the current command line. For instance, FC -3 displays the third command before the current one and allows you to repeat or modify it.

*linenum*

identifies the command to be displayed by command line number. For instance, FC 2 displays the second command of the current session, and allows you to repeat or modify it.

*string*

identifies the command to be displayed by a text string that specifies the first character or group of characters in that command. For instance, FC ADD displays the most recent ADD command and allows you to repeat or modify it.

*"quoted-string"*

identifies the command to be displayed by a text string, enclosed in double or single quotes, that specifies a character or group of characters located anywhere in that command. TMFCOM searches every character in the history buffer until it finds this string, and then displays the command that contains this string, allowing you to repeat or modify the command. Notice that the string can appear anywhere in a command, not just as the first character or first character grouping. As an example, FC "\$DATA1" displays the most recent command that referred to a disk volume named "\$DATA1."

## Security Restrictions

Anyone can enter the FC command.

## TMF State Requirement

You can enter the FC command at any time, whether or not TMF has been started.

## Usage Guidelines

Use the FC command to display, re-execute, or modify and re-execute, a specific command entered previously in the current TMFCOM session. If you use the FC



command without any parameter, the last TMFCOM command you entered is displayed.

You can use the [HISTORY](#) command to display the line numbers associated with the most recently entered TMFCOM commands. Only commands currently stored in the history buffer are displayed, and only those can be selected for action by the FC command. When you issue an FC command, the command you specify is displayed and a subcommand prompt ( . . ) appears. At the prompt, you can either

- Press the Return key to repeat the command, or
- Use the subcommands R, I, or D to replace, insert, or delete characters in the command line, respectively. As a simpler alternative to the R subcommand, you can just enter the replacement character directly under the character you want to replace.

---

**Note.** Use the space bar and the backspace keys on your keyboard to position the cursor under the text you want to change. Do not use the arrow keys for this purpose.

---

The FC command is common to the TACL command interpreter and several other subsystems in addition to TMFCOM. For complete information about its use, see the *TACL Reference Manual*.

## Output Displayed

The FC command displays the command to be repeated or modified as it was originally entered, and prompts you for subcommands to correct the command.

## Examples

Suppose that you enter the INFO AUDITDUMP command, but mistype the fourth character in the command name. TMFCOM displays an error message and issues another input prompt:

```
TMF 81> INFP AUDITDUMP *
INFP AUDITDUMP
^
**ERROR-7043** Expecting one of ABORT, ADD, ALTER, CANCEL,
    CMDVOL, DELETE, DISABLE, DISPLAY, DUMP, ENABLE, ENV, EXIT,
    FC, HELP, INFO, NEXT, OBEY, OBEYVOL, OPEN, RECOVER, RELOCATE,
    RESET, RESOLVE, SET, SHOW, START, STATUS, STOP, or VOLUME.
TMF 82>
```

You can correct this mistake by entering the FC command, which redisplay the INFO AUDITDUMP command as you erroneously entered it, followed by a prompt ( . . ) for your R, I, or D subcommand or replacement character:

```
TMF 82>FC
TMF 82>INFP AUDITDUMP *
TMF 82..
```

To correct the entry, following the subcommand prompt, enter the replacement character O under the incorrect letter (the first P) that appears on the previous line, then press the Return key (or its equivalent) to transmit the correction:

```
TMF 82>INFP AUDITDUMP *
TMF 82..  O
```

TMFCOM now displays the corrected INFO AUDITDUMP command followed by another subcommand prompt that asks for any further corrections. Because you have no further changes, you press the Return key after the subcommand prompt. Now, TMFCOM processes the INFO AUDITDUMP command, this time successfully:

```
TMF 82>INFO AUDITDUMP *
TMF 82..
Auditdump Configuration:
  Master:  enabled, medium tape, Copies 1, verifytape Off,
          blocksize 28, system \NY
TMF 83>
```

On the next line, suppose you inadvertently enter an extraneous character in the DELETE DATAVOLS command:

```
TMF 83> DELEETE DATAVOLS $BOOKS1
**ERROR-7043** Expecting one of ABORT, ADD, ALTER, CANCEL,
          CMDVOL, DELETE, DISABLE, DISPLAY, DUMP, ENABLE, ENV, EXIT,
          FC, HELP, INFO, NEXT, OBEY, OBEYVOL, OPEN, RECOVER, RELOCATE,
          RESET, RESOLVE, SET, SHOW, START, STATUS, STOP, or VOLUME.
TMF 84>
```

You correct this entry by entering the FC command followed by the D (for delete) subcommand, and then, after the command is redisplayed properly, pressing Return:

```
TMF 84> FC
TMF 84> DELEETE DATAVOLS $BOOKS1
TMF 84..  D
TMF 84> DELETE DATAVOLS $BOOKS1
TMF 84..
$BOOKS1 -- datavol deleted.
TMF 85>
```

The following example shows how to use the FC command to determine the last ADD TAPEMEDIA command issued in the current TMFCOM session, and to re-execute that command using a different tape specification:

```
TMF 85> FC ADD TAPEMEDIA
TMF 85> ADD TAPEMEDIA TT04 LABEL ON! DRIVE $TAPE
TMF 85..  5
TMF 85> ADD TAPEMEDIA TT05 LABEL ON! DRIVE $TAPE
TMF 85..
TAPE VOLUME NAME INITIALIZED.
TMF 86>
```

# HELP

The HELP command displays syntax information for TMFCOM commands.

```
HELP [command ]  
      [[command] object]  
      [* | ALL ]
```

*command*

specifies the TMFCOM command for which information is to be displayed.

*object*

specifies a TMF object for which information is to be displayed.

\* | ALL

displays a list of all TMFCOM commands.

## Security Restrictions

Anyone can enter the HELP command.

## TMF State Requirement

You can enter the HELP command at any time, whether or not TMF has been started.

## Usage Guidelines

If you enter the *command* option only, TMFCOM lists the objects that can be operated upon by that command. If you enter the *object* option only, TMFCOM lists the commands that can be used with that object. If you enter both *command* and *object*, TMFCOM displays the syntax for the command when used with the specified object.

If you omit all options, TMFCOM displays the syntax for the HELP command.

## Output Displayed

The HELP command displays the command or keyword syntax information requested.

## Examples

The following HELP command displays information about the TMFCOM command RESET, showing the objects upon which the command can operate:

**~HELP RESET**

```
-----
| RESET AUDITTRAIL
| RESET DATAVOLS
| RESET DUMPS
| RESET TAPEMEDIA
|
|-----
```

The next HELP command displays the purpose and syntax of the TMFCOM command RESET TAPEMEDIA. (The command name is RESET, and the object name is TAPEMEDIA.)

**~HELP RESET TAPEMEDIA**

```
-----
| RESET TAPEMEDIA Command
|
| Removes certain tape volume attributes, or resets
| others to their defaults, canceling the effects of
| previous SET TAPEMEDIA commands.
|
| Security:  any user.
|
|-----
```

```
RESET TAPEMEDIA {attribute [,attribute]...}
                  { * }
```

where:

attribute is:

```
[ DENSITY ]
[ DRIVE   ]
[ LABEL   ]
[ STATUS  ]
[ UNLOAD  ]
```

# HISTORY

The HISTORY command displays a history of commands entered during the current TMFCOM session, as maintained in the TMFCOM command history buffer.

|                          |
|--------------------------|
| HISTORY [ <i>lines</i> ] |
| [ RESET LAST ]           |
| [ RESET ALL ]            |

*lines*

specifies the number of command lines to display, counting from the most recently entered line. If *lines* is greater than the number of lines in the history buffer, all lines in the buffer are displayed. If *lines* is omitted, the last 20 lines are displayed.

RESET LAST

removes the most recently entered line from the command history buffer and changes the current command line number to the number of the cleared command. The HISTORY RESET LAST command is also cleared from the history buffer.

RESET ALL

removes all commands, including the HISTORY RESET ALL command, from the command history buffer and resets the command line counter to 1.

## Security Restrictions

Anyone can enter the HISTORY command.

## TMF State Requirement

You can enter the HISTORY command at any time, whether or not TMF has been started.

## Usage Guidelines

TMFCOM maintains a history buffer that contains the commands you enter during your current TMFCOM session. This buffer can hold up to 29,000 command characters (bytes). For each command, the buffer retains up to the first 2 KB of text. Once the buffer is filled, the oldest commands are deleted to make room for new commands. The HISTORY command displays the commands in this buffer, as follows:

- To display the contents of the history buffer, up to the last 20 commands stored there, enter the HISTORY command with no parameters.
- To display a specified number of command lines, starting with the line most recently entered in the buffer, enter the HISTORY command with the *lines* option.

- To clear the most recently entered command line in the buffer, and change the current line number to that of the cleared command line, enter the HISTORY command with the RESET LAST option.
- To clear all commands in the buffer and reset the command line counter to 1, enter the HISTORY command with the RESET ALL option.

In the display, all non-blank lines appear, whether or not they contained valid, executable commands. For example, if a syntax error in a command prevented its execution, HISTORY still displays the command.

The [FC](#), [!](#), and [?](#) commands also operate on commands in the history buffer. Each line that an [FC](#), [!](#), or [?](#) command causes to be repeated will appear as a new line in the HISTORY display, just as if you had entered the command once more. The [FC](#), [!](#), or [?](#) command itself, however, will not appear.

## Output Displayed

The HISTORY command lists the command lines entered in the current TMFCOM session, preceded by their line numbers. The items in the display are listed in [Table 3-5](#).

---

**Table 3-5. HISTORY Output**

| Heading or Item | Description   |
|-----------------|---|
| Line Number     | Line number of the command line.  |
| Command         | The command text entered, whether or not it constituted a valid TMFCOM command. The HISTORY command that requested the current display is shown last. |

---

## Examples

The following HISTORY command, the fifth command entered in the current session, displays all TMFCOM commands entered thus far:

```
TMF 5> HISTORY
1> OPEN SERVER \NY
2> INFO TMF
3> STATUS TMF
4> STATUS DATAVOLS
5> HISTORY
TMF 6>
```

More commands are entered in the session, and the next HISTORY command displays the four most recent commands:

```
TMF 9>HISTORY 4
6> INFO AUDITDUMP *
7> ADD DATAVOL $DATA1
8> INFO DATAVOLS
9> HISTORY 4
10>
```

The next HISTORY command clears all command lines in the buffer, and resets the current line number to 1:

```
TMF 10> HISTORY RESET ALL
TMF 1>  INFO CATALOG
Catalog Configuration:
retaindepth 2, released on
TMF 2> HISTORY
1>INFO CATALOG
2> HISTORY
TMF 3>
```

# INFO AUDITDUMP

The INFO AUDITDUMP command displays the configured attribute values of audit dump processes.

```
INFO AUDITDUMP [MASTER | MAT
                [AUX[ILIARY]nn
                [(MASTER | MAT [ ,AUX[ILIARY]nn ] ) ]
                [(AUX[ILIARY]nn [ ,MASTER | MAT ] ) ]
                [(AUX[ILIARY]nn [ ,AUX[ILIARY]nn ] ) ]
                [ *
                ]

[ , OBEYFORM]
```

MASTER | MAT  
AUX[ILIARY]nn

specifies the audit trail associated with the audit dump process whose attributes are to be displayed, thus identifying that process. MASTER or MAT specifies the master audit trail. AUXILIARYnn or AUXnn specifies an auxiliary audit trail. nn is a two-digit number ranging from 01 through 15 that specifies one of the 15 potential auxiliary audit trails in the configuration. (When the value of nn is 09 or less, you must specify the leading zero.)

\*

displays information for all audit dump processes. If you do not specify an audit trail, this is the default.

OBEYFORM

formats INFO AUDITDUMP output as command file text that can be executed by TMFCOM. TMF saves this output in the file designated by the OUT option available with all commands. Subsequently executing this command file produces a new audit dump configuration that exactly matches the existing configuration. OUT option use and command file execution are described in [Directing Output to an Alternate Destination](#) on page 2-8.

## Security Restrictions

Anyone can enter the INFO AUDITDUMP command.

## TMF State Requirement

You can enter the INFO AUDITDUMP command at any time, whether or not TMF has been started.



## Usage Guideline

The INFO AUDITDUMP command displays a subset of the information presented by the [INFO TMF](#) command.

## Output Displayed

The INFO AUDITDUMP command displays the attributes configured for the audit dump process by [ALTER AUDITDUMP](#) commands or their defaults. The primary items in the display are listed in [Table 3-6](#).

---

**Table 3-6. INFO AUDITDUMP Output**

| Heading or Item          | Description   |
|--------------------------|---|
| Audit Dump Configuration | Name of the audit trail associated with this dump:<br>MASTER is the master audit trail, and AUXILIARY <sub>nn</sub> is an auxiliary audit trail.            |
| Status                   | Status of audit dump process, either enabled or disabled.   |
| Medium                   | Medium to which the dump is directed, either disk or tape.  |
| Disk Media               | For disk dumps, fully qualified name of volume to which dump is directed.   |
| Tape Information         | For tape dumps, settings for the COPIES, VERIFYTAPE, BLOCKSIZE, UNLOAD, and SYSTEM options specified in the ALTER AUDITDUMP command or assigned by default. |

---

## Example

The following INFO AUDITDUMP command displays information for all audit dump processes:

```
TMF 76> INFO AUDITDUMP *
Auditdump Configuration:
  Master:  enabled, medium tape, copies 1, verifytape off,
          blocksize 28, unload on, system \LA

Auxiliary01:  The AUXILIARY01 audit trail is not configured
for dumping

Auxiliary02:  DISABLED, medium tape, copies 2 parallel,
              verifytape off, blocksize 28, unload on, system (\SF,\LA)
```

# INFO AUDITTRAIL

The INFO AUDITTRAIL command displays the configured attribute values of audit trails.

```
INFO AUDITTRAIL [MASTER | MAT
                 [AUX[ILIARY]nn
                 [(MASTER | MAT [ ,AUX[ILIARY]nn ] ) ]
                 [(AUX[ILIARY]nn [ ,MASTER | MAT ] ) ]
                 [(AUX[ILIARY]nn [ ,AUX[ILIARY]nn ] ) ]
                 [ *
                 ]

[ , OBEYFORM]
```

MASTER | MAT  
AUX[ILIARY]nn

identifies the audit trail for which attributes are to be displayed. MASTER or MAT specifies the master audit trail. AUXILIARYnn or AUXnn specifies an auxiliary audit trail. nn is a two-digit number ranging from 01 through 15 that specifies one of the 15 potential auxiliary audit trails in the configuration. (When the value of nn is 09 or less, you must specify the leading zero.)

\*

displays information for all audit trails. If you do not specify an audit trail, this is the default.

OBEYFORM

formats INFO AUDITTRAIL output as command file text that can be executed by TMFCOM. TMF saves this output in the file designated by the OUT option available with all commands. Subsequently executing this command file produces a new audit trail configuration that exactly matches the existing configuration. OUT option use and command file execution are described in [Directing Output to an Alternate Destination](#) on page 2-8.

---

**Note.** When you specify OBEYFORM, TMF does not write the [MaxRetainedATFiles](#) field to the INFO AUDITTRAIL output because this value is only used temporarily during volume recovery or file recovery.

---

## Security Restrictions

Anyone can enter the INFO AUDITTRAIL command.

## TMF State Requirement

You can enter the INFO AUDITTRAIL command at any time, whether or not TMF has been started.

## Usage Guideline

The INFO AUDITTRAIL command displays a subset of the information presented by the [INFO TMF](#) command.

## Output Displayed

The INFO AUDITTRAIL command displays the attributes configured for the audit trails by [ADD AUDITTRAIL](#) and [ALTER AUDITTRAIL](#) commands entered previously. The primary items in the display are listed in [Table 3-7](#).

---

**Table 3-7. INFO AUDITTRAIL Output**

| Heading or Item                        | Description  |
|--|--|
| Master/Aux <sub>nn</sub> Configuration | Audit trail for which information is displayed.  |
| ActiveVols                             | Active-audit volumes on which active-audit trail is stored.  |
| Subvolume                              | Subvolume in which audit trail is stored (always ZTMFAT).  |
| Prefix                                 | Prefix used in audit trail identifier; TMF assigns the prefix AA to MAT files and the prefixes BB through PP to files in each of the auxiliary audit trails. |
| filesize                               | Size of the audit-trail files, in megabytes.   |
| filespervolume                         | Number of files in audit trail that reside on each active-audit volume configured.   |
| auditdump on/off                       | Indication of whether audit dumping is performed for audit trail.  |
| overflowthreshold                      | Percentage of active-audit trail capacity above which TMF begins overflow copying.   |
| begintransdisable                      | Percentage of active-audit trail capacity that can be consumed before new transactions are disabled.   |
| MaxRetainedATFiles                     | Maximum number of audit-trail files to be retained on restore-audit volumes indefinitely after recovery processes no longer need them.                       |
| OverflowVols                           | Volumes used for overflow copying.   |
| RestoreVols                            | Volumes used when restoring audit dumps with RECOVER FILES command.  |

---

## Examples

The following INFO AUDITTRAIL command displays information for the master audit trail:

```
TMF >77 INFO AUDITTRAIL MASTER
AuditTrail Configuration:
Master
  ActiveVols: $MAT1, $MAT2
  Subvolume ZTMFAT, Prefix AA, filesize 20, filespervolume 3,
  auditdump on, overflowthreshold 75%, begintransdisable 90%
  MaxRetainedATFiles 0
  OverflowVols: $DATA1
  RestoreVols: $DATA2
```

The next INFO AUDITTRAIL command formats the configuration of the master audit trail as command file text, directing this output to the file COMFL1 on the default volume and subvolume:

```
TMF 78> INFO /OUT COMFL1/ AUDITTRAIL MASTER, OBEYFORM
```

This text, which can be submitted from COMFL1 as a command to TMFCOM, is as follows. (As noted earlier in this discussion, TMF does not write the MaxRetainedATFiles field to the INFO AUDITTRAIL output when the OBEYFORM option is used.)

```
SET AUDITTRAIL auditdump on
SET AUDITTRAIL filesize 20
SET AUDITTRAIL filespervolume 3
SET AUDITTRAIL overthrowthreshold 75
SET AUDITTRAIL begintransdisable 90
SET AUDITTRAIL Activevols ( &
$MAT1 &
,$MAT2 &
)
SET AUDITTRAIL OverflowVols ( &
$DATA1 &
)
SET AUDITTRAIL RestoreVols ( &
$DATA2 &
)
ADD AUDITTRAIL Master
```

The following INFO AUDITTRAIL command displays information for all audit trails:

```
TMF 79> INFO AUDITTRAIL *
```

```
AuditTrail Configuration:
```

```
Master
```

```
ActiveVols: $MAT
```

```
Subvolume ZTMFAT, Prefix AA, filesize 20, filespervolume 5
```

```
auditdump on, overflowthreshold 80%, begintransdisable 90%
```

```
MaxRetainedATFiles 0
```

```
OverflowVols: None
```

```
RestoreVols: $SPARE
```

```
Auxiliary01:
```

```
ActiveVols: $AUX
```

```
Subvolume ZTMFAT, Prefix BB, filesize 20, filespervolume 3
```

```
auditdump off, overflowthreshold 90%, begintransdisable 90%
```

```
MaxRetainedATFiles 0
```

```
OverflowVols: None
```

```
RestoreVols: None
```

# INFO BEGINTRANS

The INFO BEGINTRANS command displays the configured attribute values for the BEGINTRANS object that disable and reenables transaction processing and that apply limits on resource managers and transaction branches used in heterogeneous transaction processing.

```
INFO BEGINTRANS  
[ , OBEYFORM ]
```

## OBEYFORM

formats the INFO BEGINTRANS output as command file text that can be executed by TMFCOM. TMF saves this output in the file designated by the OUT option available with all commands. Subsequently executing this command file produces a new BEGINTRANS configuration that exactly matches the existing configuration. OUT option use and command file execution are described in [Directing Output to an Alternate Destination](#) on page 2-8. If OBEYFORM is not specified, normal formatting is used.

## Security Restrictions

Anyone can enter the INFO BEGINTRANS command.

## TMF State Requirement

Before you can enter the INFO BEGINTRANS command, TMF must be started.

## Usage Guideline

The INFO BEGINTRANS command displays a subset of the information presented by the [INFO TMF](#) command.

## Output Displayed

The INFO BEGINTRANS command displays the attributes configured for the BEGINTRANS object by [ALTER BEGINTRANS](#) commands entered previously or their defaults. The primary items in the display are listed in [Table 3-8](#).

Because newly configured values for TRANSPERCPU, RECRMOUNT, RMOPENPERCPU, and BRANCHESPERRM do not take effect until the next time TMF is stopped and then restarted, both the currently effective and the configured values are displayed for these parameters.

**Table 3-8. INFO BEGINTRANS Output**

| Heading or Item  | Description   |
|------------------|---|
| TRANSCOUNTTHRESH | Upper and lower limits at which number of active transactions in the system automatically disables or reenables new transactions.         |
| TMFLIBMEMTHRESH  | Upper and lower limits at which percentage of processor memory allocated to TMFLIB automatically disables or reenables new transactions.  |
| TMPMEMTHRESH     | Upper and lower limits at which percentage of processor memory allocated to the TMP automatically disables or reenables new transactions. |
| TRANSPERCPU      | Number of transactions that can run in all processors.  |
| AUTOABORT        | Duration after which all transactions are aborted automatically.  |
| RECRMOUNT        | Maximum number of recoverable resource managers that can be registered for the entire TMF system (spanning a single EXPAND node).         |
| RMOPENPERCPU     | Maximum number of resource managers, both recoverable and volatile, that can be open for each processor.                                  |
| BRANCHESPERRM    | Maximum number of transaction branches that can be concurrently imported or exported by each recoverable or volatile resource manager.    |

**Note.** The RECRMOUNT, RMOPENPERCPU, and BRANCHESPERRM values apply to resource managers, which support heterogeneous transaction processing. Resource managers are used in products such as NonStop TUXEDO. Although TMFCOM and the TMFSERVE programmatic interface provide TMF commands and options for operating on resource managers, system managers who use them do so in the context of issues involving multiple software subsystems and inter-platform considerations. Complete discussion of heterogeneous transaction processing and the context in which resource-manager commands and options are used lies beyond the scope of the TMF manual library.

For more information about heterogeneous transaction processing, see the Open Group TRANSACTION PROCESSING Publications, available from X/Open Publications at the following Web location:

<http://www.opengroup.org/pubs/catalog/tp.htm>

These publications can also be ordered through many bookstores.

## Examples

The following INFO BEGINTRANS command displays the BEGINTRANS object attributes:

```
TMF 81> INFO BEGINTRANS
BeginTrans Configuration:
  TransCountThresh (1500, 1600)
  TmLibMemThresh (85, 90)
  TmpMemThresh (90, 95)
  TransPerCpu (1024, 1024)
  AutoAbort 7200 SECONDS
  RecRMCount 256
  RMOpenPerCpu 128
  BranchesPerRM 128
```

After the next START TMF, the BeginTrans Configuration will be:

```
  TransPerCpu (1024, 1024)
  RecRMCount 128
  RMOpenPerCpu 32
  BranchesPerRM 128
```

(The parameter values displayed in this example are the default values assigned by TMF. For information about setting these values to suit your particular processing environment, see the [ALTER BEGINTRANS](#) command description in this manual and the guidelines in the *TMF Operations and Recovery Guide*.)

The next INFO BEGINTRANS command uses the OBEYFORM option to format output as command file text in the file BTFIL on the default volume and subvolume:

```
TMF 82> INFO /OUT BTFIL/ BEGINTRANS, OBEYFORM
```

This text, which can be submitted from BTFIL as a command to TMFCOM, is as follows:

```
Alter BeginTrans, &
transcountfresh (1500, 1600), &
tmplibmemthresh (90, 95), &
tmpmemthresh (90, 95), &
transpercpu (1024, 1024), &
autoabort 7200 SECONDS, &
recount 128, &
rmopenpercpu 32, &
branchesperrm 128
```



# INFO CATALOG

The INFO CATALOG command displays the configured attribute values for the catalog process.

```
INFO CATALOG  
[ , OBEYFORM]
```

## OBEYFORM

formats the INFO CATALOG output as command file text that can be executed by TMFCOM. TMF saves this output in the file designated by the OUT option available with all commands. Subsequently executing this command file produces a new data volume configuration that exactly matches the existing configuration. OUT option use and command file execution are described in [Directing Output to an Alternate Destination](#) on page 2-8. If OBEYFORM is not specified, normal formatting is used.

## Security Restrictions

Anyone can enter the INFO CATALOG command.

## TMF State Requirement

You can enter the INFO CATALOG command at any time, whether or not TMF has been started.

## Usage Guideline

The INFO CATALOG command displays a subset of the information presented by the INFO TMF command.

## Output Displayed

The INFO CATALOG command displays the attributes configured for the catalog process by [ALTER CATALOG](#) commands entered previously or their defaults. The primary items in the display are listed in [Table 3-9](#).

**Table 3-9. INFO CATALOG Output**

| Heading or Item | Description   |
|-----------------|---|
| Retaindepth     | Number of online dump generations retained in the catalog.  |
| Released        | Status to which online or audit tape volumes are set after all dumps on the volumes are marked for deletion. If ON, status becomes RELEASED; if OFF, status becomes SCRATCH.  |
| Roundrobin      | ROUNDROBIN attribute setting. If ON, the catalog is searched in round-robin fashion, so that all tapes in the catalog are used before any are re-used. If OFF, the catalog is searched in the standard way, so that the first available tape after the beginning of the tape list is used.  |
| Last Reel       | <p>Name (identifier) of tape reel, if TMF has been started and a tape has been assigned. Otherwise, this field displays the following information:</p> <ul style="list-style-type: none"> <li>● “None” if TMF has been started but no tape has been assigned.</li> <li>● “TMF not started” if TMF is stopping or stopped.</li> </ul> <p>This field only appears if ROUNDROBIN is ON, and is not generated when the OBEYFORM option is used in the INFO CATALOG command.</p> |

## Example

The following INFO CATALOG command displays the current catalog process attributes, showing that RETAINDEPTH is set to 3, RELEASED is OFF, and ROUNDROBIN is OFF:

```
TMF 85> INFO CATALOG
Catalog Configuration:
retaindepth 3, released Off, roundrobin Off
```

The following command shows the current catalog-process attributes when ROUNDROBIN is ON:

```
TMF 86> INFO CATALOG
Catalog Configuration:
retaindepth 3, released Off, roundrobin On
Last Reel: None.
```

# INFO DATAVOLS

The INFO DATAVOLS command displays the configured attribute values of data volumes, and can also filter and then display volumes based on selected configuration attributes.

```
INFO DATAVOL[S] [volume
                  [(volume [,volume]...)]
                  [*]

                  [,config-attrib [,config-attrib]...]

                  [,OBEYFORM

                  [, SORT {BYATID | BYNAME} ]
```

*volume*

identifies the data volume whose attribute values are to be displayed.

\*

displays the attribute values of all configured data volumes. If you do not specify a data volume, this action is the default.

*config-attrib*

filters the volumes selected for display, and can be specified in either of two ways:

```
AUDITTRAIL {MASTER | MAT }
            {AUX[ILINARY]nn}
```

displays information only for data volumes that send their audit records to the specified audit trail. MASTER or MAT specifies the master audit trail.

AUXILIARY<sub>nn</sub> or AUX<sub>nn</sub> specifies an auxiliary audit trail. *nn* is a two-digit number ranging from 01 through 15 that specifies one of the 15 potential auxiliary audit trails in the configuration. If this attribute is not specified, information about all data volumes auditing to any audit trail is displayed.

```
RECOVERYMODE {ONLINE | ARCHIVE}
```

displays information only for data volumes that have been assigned the recovery mode (either online or archive) that you specify in this parameter.

OBEYFORM

formats the INFO DATAVOLS output as command file text that can be executed by TMFCOM. TMF saves this output in the file designated by the OUT option available with all commands. Subsequently executing this command file produces a new data volume configuration that exactly matches the existing configuration. OUT option use and command file execution are described in [Directing Output to](#)

[an Alternate Destination](#) on page 2-8. If OBEYFORM is not specified, normal formatting is used.

```
SORT {BYATID | BYNAME}
```

displays the data volume information in the order indicated:

BYATID

lists data volumes first by audit-trail ID and then alphabetically by data volume name.

BYNAME

lists the data volumes by name, in alphabetical order. If the SORT parameter is omitted, this option is the default.

## Security Restrictions

Anyone can enter the INFO DATAVOLS command.

## TMF State Requirement

Before you can enter the INFO DATAVOLS command, TMF must be started.

## Usage Guidelines

You can control the volumes displayed by using the optional *config-attrib* filters. These filters can be used with one another to determine even more selectively the volumes displayed.

## Output Displayed

The INFO DATAVOLS command displays the attributes configured for data volumes by [ADD DATAVOLS](#) and [ALTER DATAVOLS](#) commands entered previously. The primary items in the display are listed in [Table 3-10](#).

---

**Table 3-10. INFO DATAVOLS Output**

| Heading or Item | Description  |
|-----------------|--|
| Volume          | Name of the data volume.   |
| Audit Trail     | Audit trail to which this data volume generates audit information. |
| Recovery Mode   | Recovery mode setting (either online or archive).                  |

---

## Examples

The following INFO DATAVOLS command, issued without any volume names or other options, displays configuration attributes for all data volumes:

TMF 90> **INFO DATAVOLS**

| Volume  | Audit Trail | Recovery Mode |
|---------|-------------|---------------|
| -----   |             |               |
| \$DATA1 | Master      | Online        |
| \$DATA2 | Master      | Online        |
| \$DATA3 | Master      | Online        |
| \$DATA4 | Master      | Online        |
| \$DATA5 | Master      | Archive       |
| \$DATA6 | Auxiliary01 | Online        |
| \$DATA7 | Auxiliary01 | Online        |
| \$DATA8 | Auxiliary01 | Archive       |
| \$DATA9 | Auxiliary01 | Online        |

The next INFO DATAVOLS command displays configuration information for all data volumes whose recovery mode is archive:

TMF 91> **INFO DATAVOLS, RECOVERYMODE ARCHIVE**

| Volume  | Audit Trail | Recovery Mode |
|---------|-------------|---------------|
| -----   |             |               |
| \$DATA5 | Master      | Archive       |
| \$DATA8 | Auxiliary01 | Archive       |

The following INFO DATAVOLS command determines if any of three specified volumes generate audit records to the AUX01 audit trail. The resulting display shows that only volumes \$DATA6 and \$DATA7 send audit records to AUX01:

TMF 92> **INFO DATAVOLS (\$DATA5,\$DATA6,\$DATA7),AUDITTRAIL AUX01**

| Volume  | Audit Trail | Recovery Mode |
|---------|-------------|---------------|
| -----   |             |               |
| \$DATA6 | Auxiliary01 | Online        |
| \$DATA7 | Auxiliary01 | Online        |

The next INFO DATAVOLS command, using the OBEYFORM parameter, formats output as command file text in the file COMFL2 on the default volume and subvolume:

TMF 93> **INFO /OUT COMFL2/ DATAVOLS, OBEYFORM**

This text, which can be submitted from COMFL2 as a series of commands to TMFCOM, is as follows:

```
Add DataVols $DATA1, AuditTrail MAT, RecoveryMode Online
Add DataVols $DATA2, AuditTrail MAT, RecoveryMode Online
Add DataVols $DATA3, AuditTrail MAT, RecoveryMode Online
Add DataVols $DATA4, AuditTrail MAT, RecoveryMode Online
Add DataVols $DATA5, AuditTrail MAT, RecoveryMode Archive
Add DataVols $DATA6, AuditTrail AUX01, RecoveryMode Online
Add DataVols $DATA7, AuditTrail AUX01, RecoveryMode Online
Add DataVols $DATA8, AuditTrail AUX01, RecoveryMode Archive
Add DataVols $DATA9, AuditTrail AUX01, RecoveryMode Online
```

Remember that a data volume cannot be added to a new TMF configuration if it was associated with a previous TMF configuration and thus might be inconsistent. Therefore, before submitting the CONFL2 command file for execution, you must take one of the following actions:

- Edit the [ALTER DATAVOLS](#) commands in COMFL2 to include the IGNOREPREVIOUSCONFIG option, which allows the data volumes to be added in spite of possible inconsistencies.
- Explicitly delete the data volumes referenced in COMFL2 from the existing TMF configuration.

If you do not take one of these actions, the [ALTER DATAVOLS](#) commands will fail when the command file is executed.

This final INFO DATAVOLS command displays configuration attributes for all data volumes (in another TMF configuration) sorted by audit-trail ID:

TMF 96> **INFO DATAVOLS \*, SORT BYATID**

| Volume   | Audit Trail | Recovery Mode |
|----------|-------------|---------------|
| -----    |             |               |
| \$DATAF1 | Master      | Online        |
| \$DATAM1 | Master      | Online        |
| \$DATAP2 | Master      | Online        |
|          | .           |               |
|          | .           |               |
|          | .           |               |
| \$DATAF2 | Auxiliary01 | Online        |
| \$DATAG1 | Auxiliary01 | Archive       |
|          | .           |               |
|          | .           |               |
|          | .           |               |
| \$DATAF3 | Auxiliary02 | Online        |
| \$DATAG2 | Auxiliary02 | Online        |
| \$DATAH1 | Auxiliary02 | Online        |
| \$DATAJ1 | Auxiliary02 | Online        |
|          | .           |               |
|          | .           |               |
|          | .           |               |

# INFO DUMPS

The INFO DUMPS command displays online and audit dump entries in the TMF catalog.

```
INFO DUMPS [file-set
            [(file-set [,file-set]...)]

    [,BRIEF | DETAIL | OBEYFORM      ]
    [,MEDIUM {TAPE | DISK}          ]
    [,SERIAL serial-number          ]
    [,SORT {BYFILE | BYSERIAL}       ]
    [,STATUS {ASSIGNED}
             {INVALID}
             {RELEASED}              ]
    [,TYPE {AUDITDUMP | ONLINEDUMP}  ]
    [,CATALOG [node.]volume.subvolume]
    [,NOT {file-set
           [(file-set [,file-set]...)]}
    [,START start-file              ]
```

## *file-set*

identifies the dumped files for which you want to retrieve online or audit dump information. If you specify a dump that is not listed in the catalog, TMF displays an error message.

For online dumps, enter *file-set* in the following format:

```
[[volume.]subvolume.]file-id
```

For audit dumps, enter *file-set* in the following format:

```
$ZTMFAT.ZTMFAT.file-id
```

\$ZTMFAT and ZTMFAT are pseudo or dummy volume and subvolume names, respectively, used to support TMF's ability to spread audit-trail files across multiple disk volumes.

---

**Note.** If you plan to refer to SQL/MX objects in a *file-set* list, you must use the Guardian names of the underlying files in all TMFCOM commands. You can run the MXGNAMES utility to convert one or more objects' ANSI names to their underlying Guardian file names. You can then use the Guardian file names in the *file-set* list. For convenience, MXGNAMES provides an option that generates names formatted as TMF file-sets for direct use in TMFCOM commands. For information about using MXGNAMES, see the *SQL/MX Reference Manual*.

---

For both online dumps and audit dumps, in the *file-set* list, file names are expanded with default volume and subvolume names, and the space required for them is impacted by other overhead as well. All *file-set* information cannot exceed the size of the SPI message buffer that supports communication between TMFCOM and TMFSERVE, which is 28 KB. If this limit is exceeded, TMFCOM displays Error Message 1050.

You can specify a *file-set* as a file-name pattern, using the asterisk (\*) and question mark (?) as wild-cards in pattern-matching. For full information about specifying a file-name pattern, see the *file-set* parameter discussion in the [DUMP FILES](#) command description.

If the *file-set* parameter is omitted, all dumped files are selected, subject to the additional filtering criteria specified in the other INFO DUMPS parameters.

BRIEF | DETAIL | OBEYFORM

specifies the size and content of the report to be produced, and its destination:

#### BRIEF

produces a short form of the report and is normally used when directing output to devices that handle 80 columns or fewer per line (terminals, for example). The report includes the name of the dumped file, the tape volume identifier, the serial number of the dump, the date and time of the dump (to the minute), and the type of the dump. If you omit both BRIEF and DETAIL, BRIEF is assigned by default.

#### DETAIL

produces a long form of the report and is normally used when directing output to devices that handle up to 132 columns per line (printers, for example). In addition to the items listed with the BRIEF option, the detailed report includes the master audit trail sequence number, the part number of the dump, the copy number of the dump, and the date and time (to the second) that the dump was taken.

#### OBEYFORM

formats the INFO DUMPS output as command file text that can be executed by TMFCOM. TMF saves this output in the file designated by the OUT option



available with all commands. Subsequently executing this command file produces new dump entries that exactly match the existing entries for the dumps you select in INFO DUMPS. Use of the OUT option and command file execution are described in [Directing Output to an Alternate Destination](#) on page 2-8. If OBEYFORM is not specified, normal formatting is used.

---

**Note.** If you use the OBEYFORM option, you must also use the SERIAL option and specify the file-set as “\*. \*.\*”. Furthermore, in this case, you cannot use the STATUS, NOT, START, or MEDIUM options. The OBEYFORM option creates a command file that will rebuild all the dump entries for a single dump serial number only. You can use this file after a DELETE DUMPS command has incorrectly deleted needed dump entries from the catalog.

---

MEDIUM {TAPE | DISK}

selects, by medium, the kind of dumps for which information is listed:

TAPE

selects only dumps to tape.

DISK

selects only dumps to disk.

If the MEDIUM parameter is not specified, the report covers both tape and disk dumps.

SERIAL *serial-number*

selects only dumps identified by *serial-number*. If you omit this option, the report includes all dumps by default.

SORT {BYFILE | BYSERIAL}

displays dump information in the order indicated:

BYFILE

lists the selected dumps alphabetically by file name. If the SORT option is omitted, this option is the default.

BYSERIAL

lists the selected dumps in descending numeric order by serial number.

## STATUS

selects the dumps for which information is listed, according to their status:

## ASSIGNED

selects only those dumps whose status is ASSIGNED, meaning that the dump file is available for use in file recovery.

## INVALID

selects only those dumps whose status is INVALID, meaning that the dump file is marked in the catalog as unreadable and cannot be used for file recovery.

## RELEASED

selects only those dumps whose status is RELEASED, meaning that the dump has been marked in the catalog as damaged or deleted and cannot be used for file recovery.

## TYPE {AUDITDUMP | ONLINEDUMP}

selects the dumps for which information is listed, according to their type:

## AUDITDUMP

selects only audit dumps.

## ONLINEDUMP

selects only online dumps.

CATALOG [*node.*]*volume.subvolume*

specifies the subvolume containing the TMF catalog from which the dump information is retrieved. If you omit this option, the default is the current TMF catalog on the node where your TMFSERVE process is running.

NOT *file-set*

identifies one or more files for which dump entries are not to be displayed. If you specify the same file in this option and in *file-set*, the dump for this file is not displayed. The NOT *file-set* list allows the same pattern-matching (wild-card) characters described in the discussion of the [file-set](#) option for this command on page [3-165](#) and the *file-set* option for the [DUMP FILES](#) command on page [3-125](#). The NOT *file-set* information cannot exceed the size of the SPI

message buffer that supports communication between TMFCOM and TMFSERVE, which is 28 KB. If this limit is exceeded, TMFCOM displays Error Message 1050.

---

**Note.** If you plan to refer to SQL/MX objects in a *file-set* list or a NOT *file-set* list, you must use the Guardian names of the underlying files in all TMFCOM commands. You can run the MXGNAMES utility to convert one or more objects' ANSI names to their underlying Guardian file names. You can then use the Guardian file names in the *file-set* list or NOT *file-set* list. For convenience, MXGNAMES provides an option that generates names formatted as TMF file-sets for direct use in TMFCOM commands. For information about using MXGNAMES, see the *SQL/MX Reference Manual*.

---

START *start-file*

specifies the first file in *file-set* for which the information is retrieved. The volume and subvolume names cannot contain wild-card characters. If you omit this option, the first file actually encountered in *file-set* is used. Enter *start-file* in the following format:

```
{[[volume.]subvolume.]{file-id | *}}
{[volume.]*.*}
```

## Security Restrictions

Anyone can enter the INFO DUMPS command.

## TMF State Requirement

You can enter the INFO DUMPS command at any time, whether or not TMF has been started.

## Usage Guidelines

If a TMF process is currently updating the catalog, the INFO DUMPS command might not report the most recent dump information. When the updating is completed, this command then reports the most recent information.

To provide hardcopy information for rebuilding the catalog if the catalog is destroyed, use the OBEYFORM option or DETAIL option of the INFO DUMPS command, with output directed to a printer.

See the *TMF Operations and Recovery Guide* for more on obtaining information about dumps.

## Output Displayed

The INFO DUMPS command displays the attributes for dumps selected by the filtering parameters of this command. The primary items in the display are listed in [Table 3-11](#).

**Table 3-11. INFO DUMPS Output**

| Heading or Item                 | Description  |
|---------------------------------|--|
| System                          | System on which report is produced.  |
| Current Date-Time               | Date and time report is produced.  |
| FileName                        | Name of dumped file.   |
| Dump Serial                     | Serial number of dump.   |
| Date-Time                       | Date and time of dump (to the minute).   |
| Dump Type                       | Type of dump (online or audit).  |
| Media Type                      | Medium on which dump is generated (disk or tape).  |
| Media Name                      | Name of device on which dumped file resides.   |
| Date-Time (DETAIL form only)    | Date and time of dump (to the second).   |
| Master (DETAIL form only)       | Master audit trail sequence number of dump.  |
| Data (DETAIL form only)         | Auxiliary audit trail sequence number of dump. If only master audit trail is configured, the master audit trail sequence number appears in both the Master and Data columns of the report. |
| Dump Status (DETAIL form only)  | Status of dump file (assigned, invalid, or released).  |
| Media Status (DETAIL form only) | Status of medium on which dump file resides (assigned, invalid, or released).  |
| Part Copy (DETAIL form only)    | Part number of the dump.   |

## Examples

The following INFO DUMPS command displays configuration information for all online and audit dumps recorded in the catalog, using the brief form of the report:

```

TMF 5> INFO DUMPS
System      \SAB      Current Date-Time      25-Feb-2002  09:56      Page      1

      FileName      Dump      Date-Time      Dump      Media
      -----      Serial      -----      Type      Type      Media Name
      -----      -----      -----      -----      ----      -----
$ZTMFAT.ZTMFAT

AA000001      66      25-Feb-2002  09:52      audit      disk      \SAB.$DEV2.
                                           ZT000066.F0000001
AA000002      66      25-Feb-2002  09:52      audit      disk      \SAB.$DEV2.
                                           ZT000066.F0000001
AA000003      66      25-Feb-2002  09:52      audit      disk      \SAB.$DEV2.
                                           ZT000066.F0000001
AA000004      66      25-Feb-2002  09:52      audit      disk      \SAB.$DEV2.
                                           ZT000066.F0000001
AA000005      66      25-Feb-2002  09:52      audit      disk      \SAB.$DEV2.
                                           ZT000066.F0000001

```

|          |    |             |       |       |      |                                   |
|----------|----|-------------|-------|-------|------|-----------------------------------|
| BB000001 | 66 | 25-Feb-2002 | 09:52 | audit | disk | \SAB.\$DEV2.<br>ZT000066.F0000001 |
| BB000002 | 66 | 25-Feb-2002 | 09:52 | audit | disk | \SAB.\$DEV2.<br>ZT000066.F0000001 |
| BB000003 | 66 | 25-Feb-2002 | 09:52 | audit | disk | \SAB.\$DEV2.<br>ZT000066.F0000001 |
| BB000004 | 66 | 25-Feb-2002 | 09:52 | audit | disk | \SAB.\$DEV2.<br>ZT000066.F0000001 |
| BB000005 | 66 | 25-Feb-2002 | 09:52 | audit | disk | \SAB.\$DEV2.<br>ZT000066.F0000001 |

The next INFO DUMPS command displays configuration information for all online and audit dumps directed to disk, using the detail form of the report. (The first two columns of the report, File Name and Dump Serial, are not shown in this example to conserve display space.)

```
TMF 6> INFO DUMPS, DETAIL, MEDIUM DISK
Current Date-Time      24-Feb-2002  14:20:11
Page 1
```

| Date - Time          | Dump Type | Master | Data | Dump Status | Media Type | Media Status | Media Name                    | Part | Copy |
|----------------------|-----------|--------|------|-------------|------------|--------------|-------------------------------|------|------|
| 24-Feb-2002 14:18:14 | online    | 1      | 1    | assigned    | disk       | assigned     | \SAB.\$DEV2.OD000067.F0000001 |      |      |
| 24-Feb-2002 14:18:15 | online    | 1      | 1    | assigned    | disk       | assigned     | \SAB.\$DEV2.OD000067.F0000001 |      |      |
| 24-Feb-2002 14:18:15 | online    | 1      | 1    | assigned    | disk       | assigned     | \SAB.\$DEV2.OD000067.F0000001 |      |      |
| 24-Feb-2002 14:18:15 | online    | 1      | 1    | assigned    | disk       | assigned     | \SAB.\$DEV2.OD000067.F0000001 |      |      |
| 24-Feb-2002 14:18:16 | online    | 1      | 1    | assigned    | disk       | assigned     | \SAB.\$DEV2.OD000067.F0000001 |      |      |
| 24-Feb-2002 14:18:16 | online    | 1      | 1    | assigned    | disk       | assigned     | \SAB.\$DEV2.OD000067.F0000001 |      |      |
| 24-Feb-2002 14:18:17 | online    | 1      | 1    | assigned    | disk       | assigned     | \SAB.\$DEV2.OD000067.F0000001 |      |      |
| 24-Feb-2002 14:18:17 | online    | 1      | 1    | assigned    | disk       | assigned     | \SAB.\$DEV2.OD000067.F0000001 |      |      |
| 24-Feb-2002 14:18:18 | online    | 1      | 1    | assigned    | disk       | assigned     | \SAB.\$DEV2.OD000067.F0000001 |      |      |
| 24-Feb-2002 14:18:18 | onlinr    | 1      | 1    | assigned    | disk       | assigned     | \SAB.\$DEV2.OD000067.F0000001 |      |      |

The following INFO DUMPS command, using the OBEYFORM option, formats output as command file text in the file DMPINFO on the default volume and subvolume:

```
TMF 7> INFO /OUT DMPINFO/ DUMPS *.*.*, SERIAL 140, OBEYFORM
```

The output text can be submitted from DMPINFO as a series of commands to TMFCOM to re-add the dump entries back into the catalog if they are accidentally deleted. An example of the output from an INFO DUMPS, OBEYFORM command appears under the [ADD DUMPS](#) command discussion.

# INFO PROCESS

The INFO PROCESS command displays the configured attribute values for the various TMF process types.

```
INFO PROCESS [process-type
              [(process-type [,process-type]...)]
              [ *
              ]
              [ ,OBEYFORM]
```

## *process-type*

identifies the type of TMF process whose configuration is to be displayed.

*process-type* may be any of the following names:

| Process-type    | Process-type Description   |
|-----------------|--|
| AUDITDUMP       | Audit dump process, which dumps audit-trail files to disk or tape.   |
| AUDITRESTORE    | Audit restore process, which restores audit dumps from disk or tape.   |
| BACKOUT         | Backout process, which undoes the effects of incomplete transactions.  |
| CATALOG         | Catalog process, which manages the TMF catalog.  |
| COPYAUDITOVFLOW | Overflow process, which writes audit-trail files to overflow-audit volumes.  |
| FILERECOVERY    | File recovery process, which reconstructs unusable audited files during recovery operations.   |
| FILERECOVLIST   | File recovery list process, which supports the file recovery process.  |
| ONLINEDUMP      | Online dump process, which dumps audited database files to tape or disk.   |
| ONLINERESTORE   | Online restore process, which restores online dumps from tape or disk.   |
| TMFMON          | TMF monitor process, which is responsible for highest-priority, performance-critical operations.   |
| TMFMON2         | Surrogate process that provides a processing environment for TMFLIB operations.  |
| TMP             | Transaction management process pair that coordinates distributed transactions and TMF configuration.   |
| VOLUMERECOVERY  | Volume recovery process that recovers database files to their most recent consistent state if they become inconsistent because of a disk volume or system failure. |

---

**Note.** The *process-type* parameter refers to all processes of a specific kind. For example, the AUDITDUMP process type identifies all audit dump processes in the system.

---

\*

displays the configured attributes of all *process-types*.

#### OBEYFORM

formats INFO PROCESS output as command file text that can be executed by TMFCOM. TMF saves this output in the file designated by the OUT option available with all commands. Subsequently executing this command file produces a new process configuration that exactly matches the existing configuration. OUT option use and command file execution are described in [Directing Output to an Alternate Destination](#) on page 2-8.

## Security Restrictions

Anyone can enter the INFO PROCESS command.

## TMF State Requirement

You can enter the INFO PROCESS command at any time, whether or not TMF has been started.

## Output Displayed

The INFO PROCESS command displays the same attributes for all process types. The primary items in the display are listed in [Table 3-12](#).

---

**Table 3-12. INFO PROCESS Output**

| Heading or Item  | Description   |
|------------------|---|
| Process Type     | One of the types of TMF processes for which configuration information is available.                             |
| Pri              | Execution priority of process, expressed as a value ranging from 1 (low priority) to 199 (high priority).       |
| CPUS             | Processor in which process type runs.   |
| MakeSaveAbend    | Indication of whether INSPECT program save-abend file is created if process of this type terminates abnormally. |
| Debug            | Indicates whether process of this type enters DEBUG or INSPECT debugger soon after it is started.               |
| Inspect          | Indicates whether INSPECT is used if process of this type falls into debugger after it is started.              |
| Extended SegSize | Size of extended segment allocated for process type <a href="#">ALTER PROCESS</a> .                             |
| PFSSize          | Size of data segment allocated for process type.  |

---

**Table 3-12. INFO PROCESS Output**

| Heading or Item | Description   |
|-----------------|---|
| ProgramFile     | Name of program file used by process type.  |
| Term            | Name of terminal used by process type.  |
| Swap            | Name of swap file specified for process type, followed by the notation "KSMF is Used." For more information about SWAP file assignment and this notation, see the <a href="#">ALTER PROCESS</a> command discussion. |

For most attributes, when the standard default value is in effect, the INFO PROCESS command displays the default value followed by the notation "- Default Value." In a few cases, however, the command does not retrieve the actual default value and instead displays only the notation "(Default)" next to the attribute name. For more information about default values, see the [ALTER PROCESS](#) command description, earlier in this section.

In the INFO PROCESS display, attribute values that cannot be altered for a process type are indicated by the notation "- Not Alterable."

## Examples

The following INFO PROCESS command displays the attributes common to all backout processes. (In the display, the SYS<sub>nn</sub> subvolume shown for the ProgramFile attribute indicates the operating system cold-load SYS<sub>nn</sub>, used by default.)

```
TMF 11> INFO PROCESS BACKOUT
```

```
Process Configuration:
```

```
BackOut:
```

|                 |                        |                 |
|-----------------|------------------------|-----------------|
| Pri             | 149                    | - Default Value |
| CPUS            | ( Default )            |                 |
| MakeSaveAbend   | On                     | - Default Value |
| Debug           | Off                    | - Default Value |
| Inspect         | On                     | - Default Value |
| ExtendedSegSize | ( 5 , 30 )             | - Default Value |
| PFSSize         | 1048576                | - Not Alterable |
| ProgramFile     | \$SYSTEM.SYSnn.TMFBOUT | - Default Value |
| TERM            | \$YMIOP.#CLCI          | - Default Value |
| Swap            | ( Default )            | - KMSF Is Used  |



The next INFO PROCESS command displays the attributes common to all catalog and volume recovery processes:

```
TMF 12> INFO PROCESS (CATALOG, VOLUMERECOVERY)
Process Configuration:
Catalog:
    .
    .
    .
VolumeRecovery:
    .
    .
    .
```

The subsequent INFO PROCESS command displays the attributes of all major TMF process types:

```
TMF 13> INFO PROCESS *
Process Configuration:
AuditDump:
    .
    .
    .
AuditRestore:
    .
    .
    .
BackOut:
    .
    .
    .
```

The last INFO PROCESS command formats output as command file text, directing this output to the file COMFL3 on the default volume and subvolume:

```
TMF 14> INFO /OUT COMFL3/ PROCESS *, OBEYFORM
```

# INFO RESOURCEMANAGER

The INFO RESOURCEMANAGER command displays the entries for recoverable resource managers currently registered in the resource manager directory.

|                             |   |   |
|-----------------------------|---|---|
| INFO {RESOURCEMANAGER   RM} | [ <i>rm-name</i>                                | ] |
|                             | [ ( <i>rm-name</i> [ , <i>rm-name</i> ] . . . ) | ] |
|                             | [ *   | ] |
| [ , OBEYFORM                |   | ] |

*rm-name*

is the name of the recoverable resource manager whose entry is to be displayed. This name can contain a maximum of 128 characters; the first character must be a letter, and the others can be alphanumeric or any of the following special characters:

\$ \_ - ^ @ &

Names that begin with ZZ-VOLATILE- are not allowed in this parameter. Names containing adjoining hyphens (--) are not allowed, although names using multiple hyphens separated by one or more intervening characters are legal. (TMFCOM interprets two consecutive hyphens as the start of a comment.) Thus, the name "ALPHA--1" is not permitted, but "ALPHA-N-1" is allowed.

\*

displays the entries for all recoverable resource managers registered in the resource manager directory. If you do not specify a resource manager name, this action is the default.

OBEYFORM

formats INFO RESOURCEMANAGER output as command file text that can be executed by TMFCOM. TMF saves this output in the file designated by the OUT option available with all commands. Subsequently executing this command file produces a new recoverable resource manager configuration that exactly matches the existing configuration. OUT option use and command file execution are described under [Directing Output to an Alternate Destination](#) on page 2-8.

## Security Restrictions

Anyone can enter the INFO RESOURCEMANAGER command.

## TMF State Requirement

Before you can enter the INFO RESOURCEMANAGER command, TMF must be started.

## Usage Guidelines

**Note.** Resource managers, to which the INFO RESOURCEMANAGER command applies, support heterogeneous transaction processing. Resource managers are used in products such as NonStop TUXEDO. Although TMFCOM and the TMFSERVE programmatic interface provide TMF commands and options for operating on resource managers, system managers who use them do so in the context of issues involving multiple software subsystems and inter-platform considerations. Complete discussion of heterogeneous transaction processing and the context in which resource-manager commands and options are used lies beyond the scope of the TMF manual library.

For more information about heterogeneous transaction processing, see the Open Group TRANSACTION PROCESSING Publications, available from X/Open Publications at the following Web location:

<http://www.opengroup.org/pubs/catalog/tp.htm>

These publications can also be ordered through many bookstores.

The INFO RESOURCEMAGER command displays information about recoverable resource managers only; volatile resource managers are not registered in the resource manager directory.

## Output Displayed

The INFO RESOURCEMANAGER command displays the resource manager directory entries for the resource managers you specify. The items in the display are listed in [Table 3-13](#).

**Table 3-13. INFO RESOURCEMANAGER Output**

| Heading or Item        | Description                          |
|------------------------|--------------------------------------|
| Resource Manager Names | Name assigned to resource manager.   |
| Owner                  | User ID of resource manager's owner. |

## Examples

The following INFO RESOURCEMANAGER command displays all currently configured resource manager entries.

```
TMF 20> INFO RESOURCEMANAGER
Resource Manager Names      Owner
-----
TUX-A1_UNIX@SUN            ( 255,147 )
TUXA2_NT@MICROSOFT         ( 255,151 )
TUX-HP@FOXII                ( 255,255 )
TUX-HP@ESSG                 ( 255,255 )
TUX-HP-TEMP@TSII           ( 255,255 )
```

The next INFO RESOURCEMANAGER command, using the OBEYFORM option, formats the output for a single resource manager as command file text in the file RECMGR on the default volume and subvolume:

```
TMF 21> INFO /OUT RECMGR/ RM TUX-HP@ESSG, OBEYFORM
```

The resulting output, which can be submitted as a command to TMFCOM, is as follows:

```
Add RM TUX-HP@ESSG, OWNER (255,255)
```

# INFO TAPEMEDIA

The INFO TAPEMEDIA command displays the configured attribute values of tape volumes.

```
INFO TAPEMEDIA [tape-volume
                [(tape-volume [,tape-volume]...)]
                [ *
                [ ,STATUS {ASSIGNED}
                        {BAD}
                        {RELEASED}
                        {SCRATCH }
                ]
                [ ,OBEYFORM ]
```

*tape-volume*

identifies the tape volume whose entries in the catalog are to be displayed. For this parameter, use the name magnetically encoded on the tape volume.

\*

displays all tape volume entries in the catalog. This is the default.

STATUS

displays tape volumes according to their status:

ASSIGNED

selects only those volumes whose status is ASSIGNED, meaning that they contain valid dumps available for file recovery.

BAD

selects only those volumes whose status is BAD, meaning that they are defective and not available for dumping or file recovery.

RELEASED

selects only those volumes whose status is RELEASED, meaning that they are no longer required for file recovery.

SCRATCH

selects only those volumes whose status is SCRATCH, meaning that they can be used for dumping files.

OBEYFORM

formats INFO TAPEMEDIA output as command file text that can be executed by TMFCOM. TMF saves this output in the file designated by the OUT option

available with all commands. Subsequently executing this command file produces tape media entries in the catalog that exactly match the existing entries. (The ADD TAPEMEDIA commands created by the OBEYFORM option specify STATUS SCRATCH regardless of the actual tape status, so that all tapes specified can be added back into the catalog.) The use of the OUT option and command file execution are described under [Directing Output to an Alternate Destination](#) on page 2-8. If OBEYFORM is not specified, normal formatting is used.

## Security Restrictions

Anyone can enter the INFO TAPEMEDIA command.

## TMF State Requirement

Before you can enter the INFO TAPEMEDIA command, TMF must be started.

## Output Displayed

The INFO TAPEMEDIA command displays entries in the catalog recorded by [ADD TAPEMEDIA](#) and [ALTER TAPEMEDIA](#) commands entered previously. The primary items in the display are listed in [Table 3-14](#).

**Table 3-14. INFO TAPEMEDIA Output**

| Heading or Item | Description   |
|-----------------|---|
| Media Name      | Name (identifier) of tape volume.                           |
| Media Type      | Type of media (always “tape”).                              |
| Media Status    | Status of tape volume: ASSIGNED, BAD, RELEASED, or SCRATCH. |

## Example

The following INFO TAPEMEDIA command shows the type and status of all tape volumes in the catalog:

```
TMF 30> INFO TAPEMEDIA
Media Name      Media Type      Media Status
-----
PLU000          tape           assigned
PLU001          tape           assigned
PLU002          tape           assigned
PLU003          tape           assigned
PLU004          tape           assigned
PLU005          tape           assigned
PLU006          tape           scratch
PLU007          tape           scratch
PLU008          tape           scratch
```

This INFO TAPEMEDIA command, using the OBEYFORM option, formats the output for all tape volumes as command file text in the file TPINFO on the default volume and subvolume:

```
TMF 35> INFO /OUT TPINFO/ TAPEMEDIA *, OBEYFORM
```

The resulting output, which can be submitted from TPINFO as commands to TMFCOM, is:

```
ADD TAPEMEDIA TF0250, LABEL OFF, STATUS SCRATCH
ADD TAPEMEDIA TF0251, LABEL OFF, STATUS SCRATCH
ADD TAPEMEDIA TF0252, LABEL OFF, STATUS SCRATCH
ADD TAPEMEDIA TF0253, LABEL OFF, STATUS SCRATCH
ADD TAPEMEDIA TF0254, LABEL OFF, STATUS SCRATCH
ADD TAPEMEDIA TF0255, LABEL OFF, STATUS SCRATCH
.
.
.
```

# INFO TMF

The INFO TMF command displays the configured attribute values of TMF.

`INFO TMF [ , {CNFGFORM | OBEYFORM} ]`

## CNFGFORM

formats the INFO TMF output as command file text that can be executed by TMFCOM. TMF saves this output in the file designated by the OUT option available with all commands. Subsequently executing this command file produces a running system that exactly matches your present TMF configuration. The use of the OUT option and command file execution are described under [Directing Output to an Alternate Destination](#) on page 2-8.

---

**Note.** When you specify CNFGFORM, TMF does not write the [MaxRetainedATFiles](#) field to the INFO TMF output because this value is only used temporarily during volume recovery or file recovery.

---

Before deleting your current TMF configuration, use the CNFGFORM option to capture this configuration in the command file. Then, you can confidently re-create that configuration at any time thereafter. The command file will contain all TMFCOM commands needed to reconfigure and start TMF, including the appropriate [STOP TMF](#) and [START TMF](#) commands.

## OBEYFORM

formats INFO TMF output as command file text that can be executed by TMFCOM. TMF saves this output in the file designated by the OUT option available with all commands. After also directing INFO DATAVOLS, OBEYFORM and INFO PROCESS, OBEYFORM output to this command file, subsequently executing the file produces a new TMF configuration that exactly matches the existing configuration. OUT option use and command file execution are described in [Directing Output to an Alternate Destination](#) on page 2-8.

---

**Note.** When you specify OBEYFORM, TMF does not write the [MaxRetainedATFiles](#) field to the INFO TMF output because this value is only used temporarily during volume recovery or file recovery.

---

In the INFO TMF command, the OBEYFORM option is an older feature than the CNFGFORM option. CNFGFORM provides a faster and easier alternative to the OBEYFORM option. Currently, OBEYFORM is most commonly used to provide permanent copies of existing configurations for informative purposes.

## Security Restrictions

Anyone can enter the INFO TMF command.



## TMF State Requirement

You can enter the INFO TMF command at any time, whether or not TMF has been started.

## Usage Guidelines

The INFO TMF command displays the current TMF configuration volume and combines the information presented by the INFO commands for the following objects: AUDITDUMP, AUDITTRAIL, BEGINTRANS (only when TMF is running), and CATALOG. Except for when the CNFGFORM option is used, information from the INFO commands for other objects (DATAVOLS, DUMPS, PROCESS, RESOURCEMANAGER, and TAPEMEDIA) does not appear in the INFO TMF display.

Sometimes, you will want to make basic changes to your TMF configuration that demand that you first use the DELETE TMF command to completely delete that configuration. Before issuing the DELETE TMF command, use the INFO TMF command with the CNFGFORM option to capture your present configuration in its entirety. The resulting command file will include all commands required to re-create and start the TMF system, including the appropriate [STOP TMF](#) and [START TMF](#) commands. When TMFCOM encounters the STOP TMF, ABRUPT command within this file, TMFCOM executes this command and then delays for about 10 seconds to allow the TMP to restart.

---

△ **Caution.** Do not use the CNFGFORM option output with versions of TMFCOM earlier than the G06.17 RVU; in such versions, the presence of the STOP TMF, ABRUPT command in the command file may result in errors.

---

The OBEYFORM option is most often used to list the current TMF configuration for informative purposes. The CNFGFORM option is the easier way to capture and reproduce a fully runnable copy of the configuration. In fact, if you use the OBEYFORM option, the output from this option alone cannot reproduce the complete TMF configuration. For the complete configuration:

- If you issue INFO TMF, OBEYFORM while TMF is running, you also need the output from an INFO DATAVOLS, OBEYFORM and an INFO PROCESS, OBEYFORM command.
- If you issue INFO TMF, OBEYFORM while TMF is stopped, you also need the output from an INFO BEGINTRANS, OBEYFORM, an INFO DATAVOLS, OBEYFORM, and an INFO PROCESS, OBEYFORM command. (The INFO BEGINTRANS, OBEYFORM and INFO DATAVOLS, OBEYFORM commands can only be issued when TMF is restarted.)

In a command file produced by the OBEYFORM option, ALTER commands that

require TMF to be running will not be executed unless you modify the file by adding a START TMF command preceding these commands.

---

**Note.** To augment the information captured through the INFO TMF, CNFG command, you can capture current information about TMF online and audit dumps by using the INFO DUMPS, OBEYFORM command. For more information about this step, see [INFO DUMPS](#) on page 3-165.

---

## Output Displayed

The primary items presented by the INFO TMF display are listed in the command descriptions for the following commands:

- INFO AUDITTRAIL ([Table 3-7](#))
- INFO AUDITDUMP ([Table 3-6](#))
- INFO CATALOG ([Table 3-9](#))
- INFO BEGINTRANS (if TMF is started) ([Table 3-8](#))

## Examples

The following INFO TMF command presents the combined attribute display:

```
TMF 40> INFO TMF
Configuration Volume:
    $SYSTEM
AuditTrail Configuration:
    Master
        ActiveVols: $AUDIT
        Subvolume ZTMFAT, Prefix AA, filesize 600, filespervolume 3,
        auditdump On, overflowthreshold 80%, begintransdisable 90%
        MaxRetainedATFiles 0
        OverflowVols: $MEZRPT, $OPER
        Restorevols: $MEZRPT
Auditdump Configuration:
    Master: enabled, medium tape, Copies 1, verifytape Off,
    blocksize 28, system \CYCLONE
Catalog Configuration:
    retaindepth 3, released off, roundrobin off
BeginTrans Configuration:
    TransCountThresh (1500,1600)
    TmfLibMemThresh (90,95)
    TmpMemThresh (90,95)
    TransPerCpu (1024,1024)
    Autoabort 7200 Seconds
    RecRMCount 256
    RMOpenPerCpu 96
    BranchesPerRM 128
After the next START TMF, the BeginTrans Configuration will be:
    TransPerCpu (1024, 1024)
    RecRMCount 256
    RMOpenPerCpu 96
```

BranchesPerRM 128

The following INFO TMF command generates CNFGFORM output:

TMF 41> **INFO /OUT CNFILE/ TMF, CNFGFORM**

Alter TMF, ConfigVol \$SYSTEM

Alter Process AuditDump, &  
     Pri 149, &  
     RESET CPUS, &  
     MakeSaveAbend       On , &  
     Debug                Off , &  
     Inspect              On , &  
     Reset ExtendedSegSize, &  
     PFSSize              1048576, &  
     Reset ProgramFile, &  
     Reset Term, &  
     Reset Swap

Alter Process AuditRestore, &  
     Pri 149, &  
     RESET CPUS, &  
     MakeSaveAbend       On , &  
     Debug                Off , &  
     Inspect              On , &  
     Reset ExtendedSegSize, &  
     PFSSize              1048576, &  
     Reset ProgramFile, &  
     Reset Term, &  
     Reset Swap

Alter Process BackOut, &

.  
.  
.

Alter Process Catalog, &

.  
.  
.

Alter Process CopyAuditOverflow, &

.  
.  
.

Alter Process FileRecovery, &

.  
.  
.

Alter Process OnLineDump, &

.  
.  
.

Alter Process OnLineRestore, &

.  
.  
.

```

Alter Process TMP, &
    .
    .
    .
Alter Process VolumeRecovery, &
    .
    .
    .
Alter Process TmfMon2, &
    .
    .
    .
Alter Process FileRecovList, &
    .
    .
    .
Alter Process TmfMon, &
    .
    .
    .

```

```
STOP TMF,ABRUPT
```

```

SET AUDITTRAIL AUDITDUMP On
SET AUDITTRAIL FILESIZE 600
SET AUDITTRAIL FILESPERVOLUME 3
SET AUDITTRAIL OVERFLOWTHRESHOLD 80
SET AUDITTRAIL BEGINTRANSDISABLE 90
SET AUDITTRAIL ACTIVEVOL ( &
$AUDIT &
)
SET AUDITTRAIL OVERFLOWVOL ( &
$MEZRPT &
, $OPER &
)
SET AUDITTRAIL RESTOREVOL ( &
$MEZRPT &
)
ADD AUDITTRAIL Master
ALTER AUDITDUMP Master, medium tape
ALTER AUDITDUMP Master, copies 1
ALTER AUDITDUMP Master, verifytape Off
ALTER AUDITDUMP Master, blocksize 28
ALTER AUDITDUMP Master, system \CYCLONE

ALTER CATALOG, &
retaindepth 3, &
released Off, &
roundrobin Off

START TMF, DISABLE BEGINTRANS

ALTER BEGINTRANS, &
transcountthresh (1500,1600) ,&
tmflibmemthresh (90,95) , &

```

```

tmpmemthresh (90,95) , &
transpercpu (1024,1024) , &
autoabort 7200 Seconds , &
recrmcount 256 , &
rmopenpercpu 96 , &
branchesperm 128

```

STOP TMF

START TMF, DISABLE BEGINTRANS

```

Add Datavol $AEGEAN, AuditTrail Mat, RecoveryMode Online, &
  IGNOREPREVIOUSCONFIG
Add Datavol $ALGAE, AuditTrail Mat, RecoveryMode Online, &
  IGNOREPREVIOUSCONFIG
Add Datavol $ATHENS, AuditTrail Mat, RecoveryMode Online, &
  IGNOREPERVIOUSCONFIG
Add Datavol $ATLIS, AuditTrail Mat, RecoveryMode Online, &
  IGNOREPREVIOUSCONFIG
Add Datavol $BASE, AuditTrail Mat, RecoveryMode Online, &
  IGNOREPREVIOUSCONFIG
Add Datavol $BAX, AuditTrail Mat, RecoveryMode Online, &
  IGNOREPREVIOUSCONFIG
Add Datavol $BFS001, AuditTrail Mat, RecoveryMode Online, &
  IGNOREPREVIOUSCONFIG
.
.
.

```

ENABLE BEGINTRANS

```

Add ResourceManager &
  IMPRM &
  , Owner (1,209)
Add ResourceManager &
  NEWONE &
  , Owner (1,195)
Add ResourceManager &
  RMEXP &
  , Owner (1,209)

```

This INFO TMF command produces OBEYFORM output:

```
TMF 42> INFO /OUT OBFIL/ TMF, OBEYFORM
```

```
Alter TMF, ConfigVol $SYSTEM
```

```

SET AUDITTRAIL AUDITDUMP On
SET AUDITTRAIL FILESIZE 600
SET AUDITTRAIL FILESPERVOLUME 3
SET AUDITTRAIL OVERFLOWTHRESHOLD 80
SET AUDITTRAIL BEGINTRANSDISABLE 90
SET AUDITTRAIL ACTIVEVOL ( &
$AUDIT &
)
SET AUDITTRAIL OVERFLOWVOL ( &
$MEZRPT &

```

```

    , $OPR &
)
SET AUDITTRAIL RESTOREVOL ( &
$MEZRPT &
)
ADD AUDITTRAIL Master
ALTER AUDITDUMP Master, medium tape
ALTER AUDITDUMP Master, copies 1
ALTER AUDITDUMP Master, verifytape Off
ALTER AUDITDUMP Master, blocksize 28
ALTER AUDITDUMP Master, system \CYCLONE
ALTER CATALOG, &
retaindepth 3, &
released off, &
roundrobin off
ALTER BEGINTRANS, &
transcountthresh (1500,1600), &
tmflibmemthresh (90,95), &
tmpmemthresh (90,95), &
transpercpu (1024,1024), &
autoabort 7200 Seconds, &
recremcount 256, &
rmopenpercpu 96, &
branchesperrm 128

```

Remember that the ALTER BEGINTRANS command in this OBEYFORM command file cannot be executed unless TMF is running. Therefore, before you execute the file, you must add a START TMF command before the ALTER BEGINTRANS command in the file. Also recall, as noted earlier in this discussion, that TMF does not write the MaxRetainedATFiles field to the INFO TMF output when the CNFGFORM or OBEYFORM option is used.

# NEXT AUDITTRAIL

The NEXT AUDITTRAIL command changes audit-trail files for a specified audit trail; it stops new audit records from being added to the current audit-trail file and establishes the next file in the audit trail sequence as the current audit-trail file.

|  |
|--|
| <pre>NEXT AUDITTRAIL {MASTER   MAT }                   {AUX[ ILIARY ] nn }</pre> |
|--|

MASTER | MAT  
AUX[ ILIARY ] *nn*

identifies the audit trail for which you want to change audit-trail files. MASTER or MAT specifies the master audit trail. AUXILIARY<sub>*nn*</sub> or AUX<sub>*nn*</sub> specifies an auxiliary audit trail. *nn* is a two-digit number ranging from 01 through 15 that specifies one of the 15 potential auxiliary audit trails in the configuration. (When the value of *nn* is 09 or less, you must specify the leading zero in this value.)

## Security Restrictions

You can issue the NEXT AUDITTRAIL command only if you are a member of the super user group.

## TMF State Requirement

Before you can enter the NEXT AUDITTRAIL command, TMF must be started.

## Examples

The following NEXT AUDITTRAIL command closes the current audit-trail file for the master audit trail, and assigns the next audit-trail file in the sequence as the new audit-trail file:

```
TMF 45> NEXT AUDITTRAIL MAT
```

The subsequent NEXT AUDITTRAIL command similarly changes the audit-trail file for the auxiliary audit trail AUX02:

```
TMF 46> NEXT AUDITTRAIL AUX02
```

# OBEY

The OBEY command executes a series of commands entered in a command file.

`OBEY [node.][volume.][subvolume.]file-id`

*node*

identifies the node (system) on which the command file is stored.

*volume*

identifies the disk volume on which the command file is stored.

*subvolume*

identifies the subvolume on which the command file is stored.

*file-id*

identifies the command file, which contains one or more valid TMFCOM commands.

## Security Restrictions

Anyone can enter the OBEY command.

## TMF State Requirement

You can enter the OBEY command at any time, whether or not TMF has been started.

## Usage Guidelines

If you omit *node*, *volume*, or *subvolume*, TMFCOM uses the default names supplied by the [OBEYVOL](#) or [VOLUME](#) commands. If you do not issue an [OBEYVOL](#) or [VOLUME](#) command, TMFCOM uses the defaults in effect when TMFCOM was started. If you issue both [OBEYVOL](#) and [VOLUME](#) in this session, [OBEYVOL](#) takes precedence over [VOLUME](#).

A command file can contain other OBEY commands, nested up to four levels deep.

TMFCOM reads the commands in the command file until it reaches an [EXIT](#) command or the end of the file. Then, TMFCOM closes the command file and reads its next command from the file from which it read the OBEY command.

If the command file contains [VOLUME](#) or [OBEYVOL](#) commands, the defaults are not reset after the command file is executed.

TMFCOM returns an error message if any part of the file specification is not valid or if the file does not exist. After this message appears, the current source for command input regains control.



If TMFCOM encounters an error while executing a command from a command file, TMFCOM terminates command file execution and ignores the remaining commands in the command file. TMFCOM then reads its next command from the file from which it read the OBEY command.

## Example

The following OBEY command reads and processes TMFCOM commands from the command file named TMFSTATS:

```
TMF 46> OBEY TMFSTATS
```

# OBEYVOL

The OBEYVOL command specifies the default node, volume, and subvolume names for expanding an unqualified command file name used in an [OBEY](#) command.

```
OBEYVOL [ [ node . ] volume . ] subvolume ]
```

*node*

is the name of the default node (system).

*volume*

is the name of the default disk volume.

*subvolume*

is the name of the default subvolume.

## Security Restrictions

Anyone can enter the OBEYVOL command.

## TMF State Requirement

You can enter the OBEYVOL command at any time, whether or not TMF has been started.

## Usage Guidelines

The OBEYVOL command applies to the file name referenced as the *file-id* parameter in the [OBEY](#) command and not to file names that appear in the TMFCOM commands contained in the command file (except for file names in [OBEY](#) commands nested within the command file). In other words, the OBEYVOL command affects the command file's name, not its contents (except for nested [OBEY](#) commands).

If you do not issue an OBEYVOL command, no default OBEY command volume and subvolume are recorded for your session. In this case, unqualified command file names used in [OBEY](#) commands are expanded according to the defaults in effect when you started TMFCOM.

If you enter the OBEYVOL command but omit one or more parameters, the default action depends on the parameters you omit:

- If you supply *node* or *volume* but omit *subvolume*, the default subvolume in effect when you started TMFCOM becomes the recorded subvolume for your session.
- If you supply *node* or *subvolume* but omit *volume*, the default volume in effect when you started TMFCOM becomes the recorded volume for your session.

- If you omit *node*, no default node is recorded for your session; the node on which your TMFSERVE process is running is assigned.
- If you omit *node*, *volume*, and *subvolume*, the recorded default node, volume, and subvolume for the OBEY command are cleared. As a result, unqualified file names are expanded according to the defaults in effect when you started TMFCOM.

At any time, you can display the currently recorded OBEY command default volume and subvolume for your session by entering the [ENV](#) command.

When the OBEYVOL command is issued from a command file, and you omit both *volume* and *subvolume* from this command, the following actions result:

- If the command file is a disk file, unqualified command file names are expanded according to the defaults for the current command input file.
- If the command file is not a disk file, an error occurs. TMFCOM displays an error message and ignores the OBEYVOL command.

If you specify an invalid volume or subvolume name in *volume* or *subvolume*, respectively, TMFCOM returns an error message and ignores the OBEYVOL command.

The distinction between the OBEYVOL command and the CMDVOL and VOLUME commands, described elsewhere in this section, is as follows:

- The OBEYVOL command applies only to unqualified file names used as OBEY command parameters.
- The CMDVOL command applies to all unqualified file names except those used as OBEY command parameters.
- The VOLUME command applies to all unqualified file names encountered in all TMFCOM commands during the session, except those used as [CMDVOL](#) and [OBEYVOL](#) command parameters.

## Example

The following OBEYVOL command specifies that OBEY command file names whose disk volume names are not qualified are expanded using the volume name \$HELLO:

```
TMF 49> OBEYVOL $HELLO
```

The output from the following ENV command reflects this change:

```
TMF 50> ENV
Default Volume      \TSII.$BOOKS1.STAFF
Volume             None Specified
Command Volume      None Specified
Obey Volume         $HELLO.STAFF
In File             \TSII.$ZTN1A.#PTGUJ0U
Out File            \TSII.$ZTN1A.#PTGUJ0U
Server Process      \TSII.$Z34H
```

# OPEN SERVER

The OPEN SERVER command specifies the TMFSERVE process with which your TMFCOM process communicates to control TMF.

```
OPEN SERVER {node
             { [node.] server }

             [,CPU cpu-num      ]
             [,PRI integer      ]
             [,SWAP volume      ]
             [,TERM [node.] term]
```

## *node*

identifies the node (system) on which the TMFSERVE process runs. The syntax rules for specifying a node name appear under [Disk File Names](#), earlier in this section. The default is the node on which your TMFCOM process is running.

## *server*

identifies the TMFSERVE process with which you want to communicate. If the process specified in *server* does not exist, TMF automatically creates a TMFSERVE process with that name. The name you specify in *server* can contain a dollar sign (\$) followed by one to five alphanumeric characters. If you omit *server*, TMF automatically creates a process with a randomly selected name.

## CPU *cpu-num*

identifies the processor on which the TMFSERVE process runs. The values for *cpu-num* can range from 0 through 15. If you omit this option, the primary processor for the TMFSERVE process is randomly selected when the command executes. This option cannot be specified for an existing process.

## PRI *integer*

specifies the execution priority of the TMFSERVE process. This value can range from 1 (for lowest priority) through 199 (for highest priority). If you omit this option, the default is the priority of the TMFCOM process when the command executes. This option cannot be specified for an existing process.

## SWAP *volume*

identifies the swap volume for the TMFSERVE process' data and extended segments. This option must indicate a local volume name. If you omit this parameter, the default assignment is the same swap volume used by TMFCOM. This option cannot be specified for an existing process.

TERM [*node.*] *term*

identifies the home terminal for the TMFSERVE process. This is the terminal where system start-up error messages are displayed. As a general rule, this option should not be used. Without this option, the default is the home terminal currently used by TMFCOM.

## Security Restrictions

Anyone can enter the OPEN SERVER command.

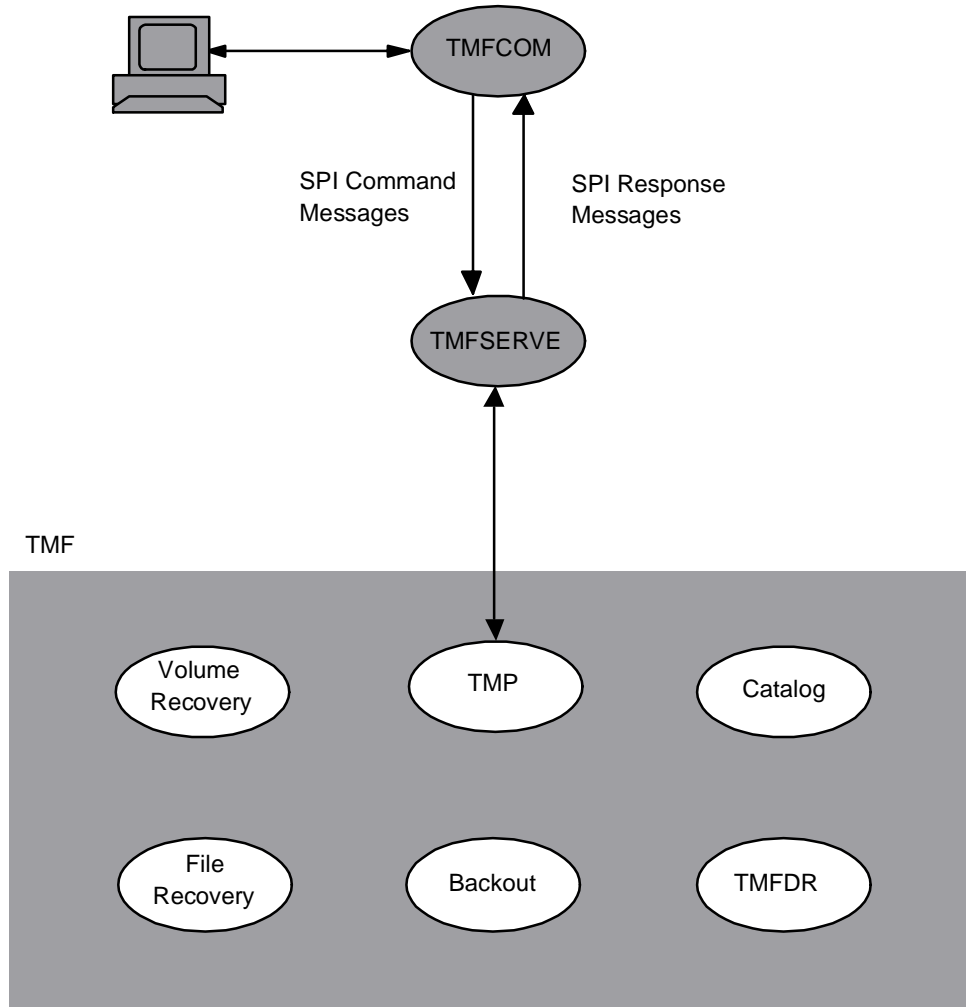
## TMF State Requirement

You can enter the OPEN SERVER command at any time, whether or not TMF has been started.

## Usage Guidelines

To process your commands, TMFCOM communicates with TMF through the TMFSERVE process, as illustrated in [Figure 3-1](#). In this interaction, TMFCOM is the requester process and TMFSERVE is the server process. TMFSERVE passes your requests to the transaction management process (TMP), which in turn typically distributes them to other TMF processes (such as the volume recovery process or the backout process) for servicing. These other processes communicate with each other as well as the TMP in servicing requests.

---

**Figure 3-1. TMFCOM and TMFSERVE Processes**


VST003.vsd

In general, you issue an OPEN SERVER command to communicate with an existing TMFSERVE process or to create and communicate with a new TMFSERVE process.

## Accessing TMFSERVE on Your Local Node

When you start a session by entering the TMFCOM keyword, no TMFSERVE process is running on behalf of your session. When you issue your first command in the session, however, TMFCOM automatically creates a TMFSERVE process with a randomly selected name on your local node. From this point on, this TMFSERVE process handles the subsequent TMFCOM commands.

With the OPEN SERVER command, however, you can create a new TMFSERVE process with attributes tailored to your own needs. You can also use this command to access a specific existing TMFSERVE process. With the OPEN SERVER command, for instance, you and another user could explicitly open and share access to the same

TMFSERVE process. Up to 10 user processes can share the same TMFSERVE process.

When you enter the OPEN SERVER command, one of the following actions takes place:

- If the TMFSERVE process you specify already exists, TMF establishes communication between this TMFSERVE process and your TMFCOM process. (For an existing process, the OPEN SERVER command can specify the TERM option but not the CPU, PRI, and SWAP options.)
- If the TMFSERVE process you specify does not exist, TMF starts a new TMFSERVE process with the options specified, and establishes communication between this new process and your TMFCOM process. (For a new process, the OPEN SERVER command can include any or all of the options: CPU, PRI, SWAP, and TERM. You must be certain, however, to use an object file named "TMFSERVE" for the new process.)

After you establish communication with a TMFSERVE process through the OPEN SERVER command, TMFCOM directs all subsequent commands to that TMFSERVE process.

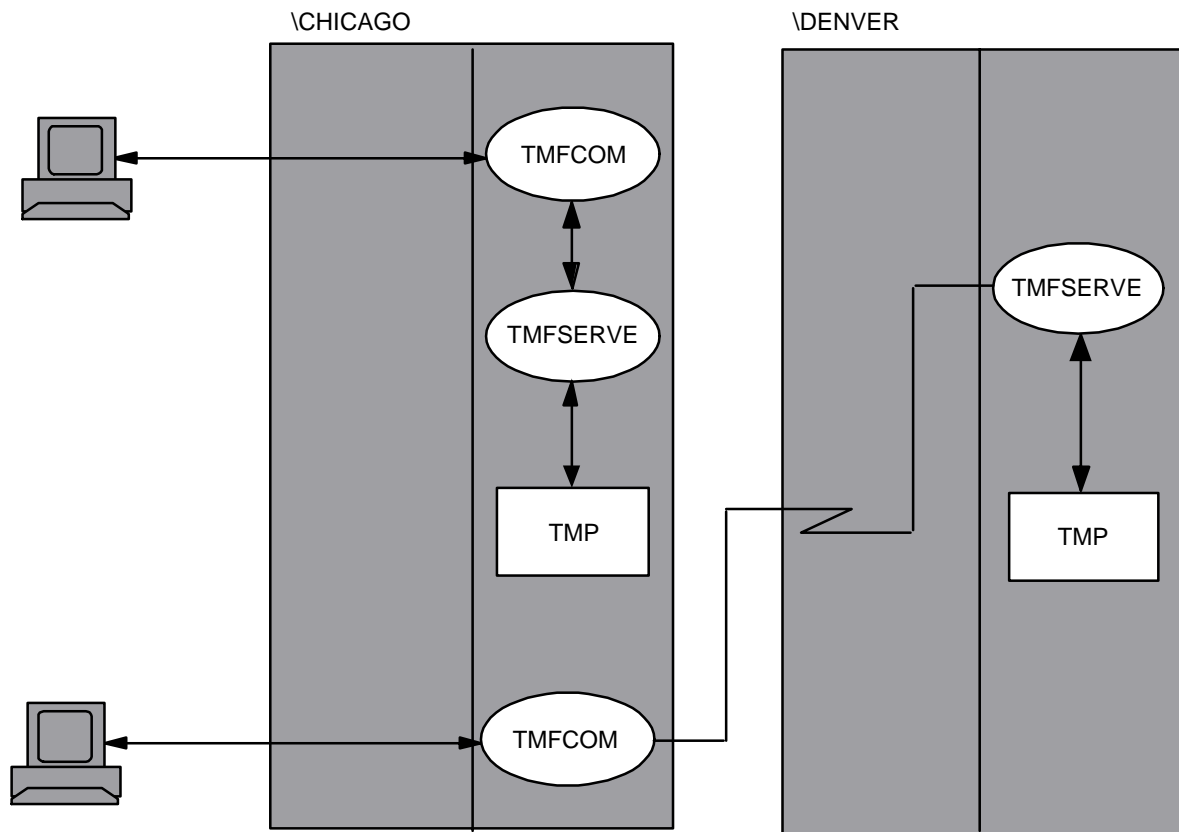
To obtain the name and other information about the currently open TMFSERVE process, enter the STATUS SERVER command.

Be certain to use an object file named "TMFCOM" for your TMFCOM process; otherwise, TMFCOM message text will not be displayed at your terminal.

## Accessing TMFSERVE on a Remote Node

To control and use TMF on another node, you must issue an OPEN SERVER command that establishes communication with a TMFSERVE process on that node.

The relationship between a TMFCOM process running on the \CHICAGO node with a TMFSERVE process running on the \DENVER node is illustrated in [Figure 3-2](#). The TMFCOM process on \CHICAGO sends commands to the TMFSERVE process on \DENVER, which in turn passes these commands to the TMP on \DENVER. The TMP on \DENVER distributes these commands to other TMF processes on \DENVER for service.

**Figure 3-2. TMFSERVE on a Remote Node**

VST004.vsd

## Understanding Interoperability Issues

When you use a TMFCOM OPEN SERVER command to open a TMFSERVE process on a remote node in an EXPAND network, you may find your TMFCOM process interacting with a newer or older version of TMFSERVE. If the remote TMFSERVE does not support the options introduced in the version of TMFCOM covered by this manual edition, the following results apply:

- A TMFCOM operational command (such as ALTER AUDITTRAIL or RECOVER FILES) that specifies one of these new options fails with an error message.
- A TMFCOM command that displays information (such as INFO AUDITTRAIL or INFO CATALOG) does not present information related to the new option.

[Table 3-15, TMFCOM vs TMFSERVE Version Interaction: Operational Commands](#), shows the behavior for operational commands when different versions of TMFCOM and TMFSERVE interact.



**Table 3-15. TMFCOM vs TMFSERVE Version Interaction: Operational Commands**

| TMFCOM Version | TMFSERVE Version | Result  |
|----------------|------------------|---|
| Current        | Current          | Works with no errors.   |
| Current        | Previous         | Produces a TMFCOM error message.  |
| Previous       | Current          | Works with no errors (because the new option cannot be specified through TMFCOM). |
| Previous       | Previous         | Works with no errors (because the new option cannot be specified through TMFCOM). |

[Table 3-16, TMFCOM vs TMFSERVE Version Interaction: Informational Commands](#), shows the behavior for informational commands when different versions of TMFCOM and TMFSERVE interact. Notice that in no cases are error messages returned.

**Table 3-16. TMFCOM vs TMFSERVE Version Interaction: Informational Commands**

| TMFCOM Version | TMFSERVE Version | Result   |
|----------------|------------------|--|
| Current        | Current          | Displays the value of the new option.  |
| Current        | Previous         | Displays no information for the new option.  |
| Previous       | Current          | Displays no information for the new option (TMFSERVE returns a new token, which is ignored by TMFCOM). |
| Previous       | Previous         | Displays no information for the new option (neither TMFCOM nor TMFSERVE recognize the new option).     |

The TMFCOM HELP command displays information about the new options only if it is connected to a current version of TMFSERVE.

## Examples

The following OPEN SERVER command establishes a connection between an existing TMFSERVE process and the current TMFCOM process. The TMFSERVE process is named \$SERVE and runs on the same node as your current TMFCOM process:

```
TMF 61> OPEN SERVER $SERVE
```

The STATUS SERVER command displays the activity of the currently open TMFSERVE process, \$SERVE:

```
TMF 62> STATUS SERVER
TMF Server Status:
System:          \FOXII (147)
Date-Time:       12-Dec-2003 01:09:30
Process          $SERVE (6,91)
Creator ID:      (8,21)
Process ID:      (8,21)
```

```
Priority:      125
Object Name:   \FOXII.$SYSTEM.SYS00.TMFserve
HomeTerm:     \TSII.$ZTN1A
Swap Volume:   $SYSTEM
Version:       TMFserve - T8694G07 - (24NOV2003 - TMF)
```

The next OPEN SERVER command creates a new TMFserve process and establishes a connection between this process and the current TMFCOM process. The new TMserve process, named \$SV02, runs on Processor 3 on the same node as your TMFCOM, with an execution priority of 150:

```
TMF 63> OPEN SERVER $SV02, CPU 3, PRI 150
```

The last OPEN SERVER command establishes communication between your TMCOM process and an existing TMFserve server on a remote node. The TMFserve process is named \$SV03 and runs on the node named \DENVER:

```
TMF 64> OPEN SERVER \DENVER.$SV03
```

# RECOVER FILES

The RECOVER FILES command initiates file recovery: restoring a database to a consistent state.

```
RECOVER FILES {file-set
               {(file-set [,file-set]...)}

[,NOT {file-set
       {(file-set [,file-set]...)}      ]

[,FROMARCHIVE  ]

[,NOWAIT       ]

[,TOFIRSTPURGE ]

[,SERIAL serial-number]

[,SYSTEM node]

[,TIME {month[-]day[-]year, hr:min[:sec]  [GMT]}
       {day[-]month[-]year, hr: min[:sec] [GMT]}}]

[,TOMATPOSITION (atseqno, rba)]

[,UNLOAD {ON | OFF}                                ]

[,TOPHYSVOL {volume}                               ]

[,WHEREPHYSVOLIS {volume
                  {(volume [,volume]...)}      ]

[,RECOVERVDPHASE1
                        ]
[,MAP NAME[S] {old-filesset-list TO new-filesset}
               {(old-filesset-list TO new-filesset
                [,old-filesset-list TO new-filesset]...)}]
```

*file-set*

identifies the files to be recovered. Enter *file-set* in the following format:

```
[ [volume.]subvolume. ]file-id
```

---

**Note.** If you plan to refer to SQL/MX objects in a *file-set* list, you must use the Guardian names of the underlying files in all TMFCOM commands. You can run the MXGNAMES utility to convert one or more objects' ANSI names to their underlying Guardian file names. You can then use the Guardian file names in the *file-set* list. For convenience, MXGNAMES provides an option that generates names formatted as TMF file-sets for direct use in TMFCOM commands. For information about using MXGNAMES, see the *SQL/MX Reference Manual*.

---

In the *file-set* list, file names are expanded with default volume and subvolume names, and the space required for them is impacted by other overhead as well. All *file-set* information cannot exceed the size of the SPI message buffer that supports communication between TMFCOM and TMFSERVE, which is 28 KB. If this limit is exceeded, TMFCOM displays Error Message 1050.

You can specify a *file-set* as a file-name pattern. This pattern resembles a file name but designates a set of entities (that is, a set of disk volumes, subvolumes, or files) through the use of pattern-matching (wild-card) characters. The wild-card characters are:

- \* An asterisk matches zero or more letters, digits, or a combination of these.
- ? A question mark matches exactly one letter or digit.

The file-name pattern syntax is:

```
[ [$pattern.]pattern. ]pattern
```

In this syntax, *pattern* consists of one or more characters. Allowable characters are letters, digits, asterisks (\*), and question marks (?). The maximum length of a pattern is 8 characters, including wild-card characters. Wild-card characters can appear in any portion of a name, for as many times as there can be characters in that portion.

Examples:

- \* matches all files in the current subvolume
- \*z\* matches all files in the current subvolume that have IDs containing the letter "z."
- \$DISK?? matches all disk volumes at the current node that have seven-character names starting with "\$DISK," such as \$DISK01" and "\$DISK26."
- P\*. \* matches all disk files on the current volume that are in subvolumes whose names begin with the letter "P."

### NOT *file-set*

identifies one or more files that are not to be recovered, whether or not they are specified in *file-set* and regardless of their redo-needed or undo-needed status. The NOT *file-set* list allows the same pattern-matching (wild-card) characters described earlier in the discussion of the *file-set* option for this command and the *file-set* option for the [DUMP FILES](#) command on page [3-125](#). The NOT *file-set* information cannot exceed the size of the SPI message buffer that supports communication between TMFCOM and TMFSERVE, which is 28 KB. If this limit is exceeded, TMFCOM displays Error Message 1050.

---

**Note.** If you plan to refer to SQL/MX objects in a *file-set* list or a NOT *file-set* list, you must use the Guardian names of the underlying files in all TMFCOM commands. You can run the MXGNAMES utility to convert one or more objects' ANSI names to their underlying Guardian file names. You can then use the Guardian file names in the *file-set* list or NOT *file-set* list. For convenience, MXGNAMES provides an option that generates names formatted as TMF file-sets for direct use in TMFCOM commands. For information about using MXGNAMES, see the *SQL/MX Reference Manual*.

---

### FROMARCHIVE

recovers all files requested, regardless of the flags set in the file labels, using the relevant online dumps and audit trail dumps.

However, if you omit the FROMARCHIVE, TIME, TOMATPOSITION, SERIAL, and MAP NAMES parameters from the RECOVER FILES command, the only files recovered are those that have only the undo-needed flag set in the file label; the file recovery process ignores all other files. To recover those files with the undo-needed flag set, the file recovery process reads the audit trail in reverse and applies compensating records (before images) to the affected files.

If you omit the FROMARCHIVE parameter but include the TIME, TOMATPOSITION, SERIAL, or MAP NAMES parameter, the FROMARCHIVE operation becomes the default operation (so that all files requested are recovered). For example, if you enter the following command, FROMARCHIVE takes effect by default:

```
TMF 10> RECOVER *.*.*, SERIAL 92
```

### NOWAIT

requests that when TMFCOM accepts the RECOVER FILES command, it suppresses display of the EMS events for the file-recovery operation and immediately issues a prompt for another command.

You can check the status of the recovery operation later by issuing a [STATUS OPERATIONS](#) command.

If you do not specify NOWAIT, TMFCOM lists the EMS events on your terminal or writes them to the OUT file (if you specify that) as the recovery progresses.

**TOFIRSTPURGE**

terminates file recovery when the first audit record for purging any file in the specified file sets is encountered; the purge record is not reapplied.

**SERIAL** *serial-number*

recovers only files from the online dump identified by *serial-number*. If you enter a number less than 64, TMFCOM returns an error message. When you select the SERIAL option, TMF always also uses the FROMARCHIVE option, even if you do not explicitly select it.

**SYSTEM** *node*

identifies the node on which all tape media used for restoring online dumps are mounted. The syntax rules for specifying a node name appear under [Disk File Names](#), earlier in this section. The default is the local node. If you specify a remote node, Guardian labeled-tape processing must be enabled on that node and must be running on the NonStop operating system. TMF, however, need not be configured or running on the remote node.

Tape media used for restoring audit dumps are requested on the node specified in the audit dump configuration.

**TIME** {*month*[-]*day*[-]*year*, *hr*:*min*[:*sec*] [GMT] }  
 {*day*[-]*month*[-]*year*, *hr*: *min*[:*sec*] [GMT] }

applies the FROMARCHIVE option (whether it is specified or not), recovering all files requested from the relevant online and audit dumps, and directs that file recovery redo all transactions that were committed before the specified time. Transactions that began before the specified time but that were not yet committed are backed out. The database is recovered to the time specified, rather than to the most recent consistent state. Be certain to enter the date and time for this option completely and accurately.

---

△ **Caution.** When you issue a RECOVER FILES command to initiate file recovery to a timestamp, and the time listed in the TMF catalog is not the time that the online dump actually took place, you risk destruction of your database because you will probably not recover all the data you need.

When you use the TIME option in a RECOVER FILES command, the value you enter must be greater than the actual dump time by at least one second to ensure a proper recovery; otherwise, TMF recovers no files.

---

If you do not specify the TIME option, the time at which you issue the RECOVER FILES command is used for the file recovery operation.

**month**

is a three-character abbreviation for the name of the month:

{JAN|FEB|MAR|APR|MAY|JUN|JUL|AUG|SEP|OCT|NOV|DEC}

*day*

is a two-digit integer for the day of the month.

*year*

is a four-digit integer for the calendar year.

---

**Note.** You can use either dashes (-) or spaces, or both, as delimiters to separate the *month*, *day*, and *year* values, but you must use at least one of these delimiters.

---

*hr*

is an integer for the hour, from 0 through 23.

*min*

is an integer for the minute, from 00 through 59.

*sec*

is an integer for the second, from 00 through 59.

GMT

indicates that the TIME value is specified in Greenwich mean time. If GMT is omitted, the default is the local time at your site.

TOMATPOSITION (*atseqno*, *rba*)

applies the FROMARCHIVE option (whether it is specified or not), recovering all files requested from the relevant online and audit dumps, and directs that file recovery redo all transactions that were committed up to a specified location in the master audit trail (MAT). Transactions that began before the specified location but that were not yet committed are backed out. The database is recovered to the location specified, rather than to a time specified or to the most recent consistent state. The location is specified by the *atseqno* and *rba* parameters:

*atseqno*

specifies the sequence number of the MAT file where recovery is to stop. For this value, you enter only the numeric portion of the file name, with leading zeros optional. For example, to recover to a relative byte address in MAT file AA000239, you can enter either 000239 or 239 as the *atseqno* value.

*rba*

specifies the relative byte address (byte offset) in the MAT file where recovery is to stop.

UNLOAD {ON | OFF}

determines whether the tape volume is unloaded after it has been used. ON unloads the tape volume. OFF suppresses unloading. The default is ON.

TOPHYSVOL *volume*

identifies the physical target volume on which TMF should place the recovered files. If you specify both the TOPHYSVOL and the WHEREPHYSVOLIS options, the WHEREPHYSVOLIS option takes precedence.

- 
- △ **Caution.** The TOPHYSVOL option is intended for use with NonStop Storage Management Foundation (SMF) files only. Before you issue a RECOVER FILES command with this option, it is vital that you read [Options for the SMF Product](#) on page 3-212.
- 

WHEREPHYSVOLIS *volume*

identifies the original physical volumes where the files specified in the *file-set* parameter are stored. This option limits the files to be recovered to those that reside on the physical volumes. If you specify both the TOPHYSVOL and the WHEREPHYSVOLIS options, the WHEREPHYSVOLIS option takes precedence.

- 
- △ **Caution.** The WHEREPHYSVOLIS option is intended for use with SMF files only. Before you issue a RECOVER FILES command with this option, it is vital that you read [Options for the SMF Product](#) on page 3-212.
- 

RECOVERVDPPHASE1

recovers the underlying catalog for the virtual disk process (VDP). If you omit this option, the files specified in *file-set* are recovered, but the catalog is not.

- 
- △ **Caution.** The RECOVERVDPPHASE1 option is intended for use with SMF files only. Before you issue a RECOVER FILES command with this option, it is vital that you read [Options for the SMF Product](#) on page 3-212.
- 

```
MAP NAME[S] {old-fileset-list TO new-fileset}
             {(old-fileset-list TO new-fileset
               [,old-fileset-list TO new-fileset]...)}
```

allows files to be recovered to different volumes, subvolumes, and file IDs than the ones from which they were dumped. The new (target) files, however, must be located on the same node (system) and same instance of TMF as the old (source) files.

When you select the MAP NAMES option, TMF always also uses the FROMARCHIVE option, even if you do not explicitly select it.

---

**Note.** If you specify the MAP NAMES option, you cannot use the RECOVERVDPPHASE1 option.

---



*old-fileset-list* identifies the source files in the online dump disk or tape, and can specify one or more *file-sets*, in the following format:

```
file-set
(file-set [, file-set]...)
```

*new-fileset* identifies the target file names to which the files will be recovered, and can specify only one *file-set* in this format:

```
file-set
```

In both *old-fileset-list* and *new-fileset*, enter *file-set* in the standard format for this variable defined earlier under [file-set on page 3-202](#) and repeated here:

```
[volume.]subvolume.file-id
```

In *file-set*, file names are expanded with default volume names and subvolume names. Notice, however, that qualified file names cannot specify node (system) names or node numbers.

In *file-set*, you can specify both physical volumes and SMF disk volumes.

The MAP NAMES option does not permit a NOT *file-set* construct. That is, you cannot use the MAP NAMES option to identify files to be excluded from recovery to a target volume.

For the MAP NAMES option, you can specify *file-set* as a file-name pattern, allowing *file-set* to implicitly refer to multiple files. If you specify more than one renaming pattern in this option and there is a conflict, TMF uses the first applicable pattern. Within the context of MAP NAMES, the following syntax restrictions apply:

#### Wildcard Characters in *old-fileset-list*

In *old-fileset-list*, you can use wildcard characters in the volume, subvolume, and file-ID fields as follows:

- \* An asterisk matches from 0 through 8 letters, digits, or a combination of these, in the position where it appears.
- ? A question mark matches exactly one letter or digit, in the position where it appears.

#### Wildcard Characters in *new-fileset*

In *new-fileset*, you cannot use the question mark as a wildcard character. However, you can use the asterisk in this way:

- \* An asterisk can be used in place of a volume name, subvolume name, or file ID. The asterisk indicates “all.” For example, \$data1.\*.filea refers to all files named filea on the volume named \$data1. Using wildcard characters in partial names, such as \$data1.sub\*.filea, is not allowed.

Examples showing pattern-matching and wildcard-character use appear earlier in this discussion under [file-set on page 3-202](#) and later, under [Examples](#) on page 3-213.

## Security Restrictions

You can issue a RECOVER FILES command only if you are a member of the super user group.

## TMF State Requirement

Before you can enter the RECOVER FILES command, TMF must be started.

## Usage Guidelines

For the RECOVER FILES command to operate successfully, AUDITDUMP ON must have been specified in the [ADD AUDITTRAIL](#) or [ALTER AUDITTRAIL](#) command that established the configured attribute values of the audit dump process, and at least one online dump must have been taken of the affected audited files.

---

**Note.** For both online dumps and audit dumps, Expand links to remote disk volumes must support 56 KB input/output.

---

If you want to recover more than about 4,500 files, consider issuing multiple RECOVER FILES commands against small groupings of files. If you specify a greater number of files in a single command, recovery performance can be adversely affected.

The total number of characters (bytes) in the *file-set* list and NOT *file-set* list, together, cannot exceed 28672.

Multiple file recovery processes can be running at any point in time.

Each generation of an online dump provides a point from which the file recovery can start.

File recovery status is output to the EMS log file current at the time of the recovery.

---

**Caution.** Before recovering a file that exists in the system, consider backing up that file to preserve its current content and status.

---

HP strongly recommends that you direct output from the RECOVER FILES command to a disk file or a printer by using the standard OUT option; doing so enables you to save the entire output for future reference. If you omit the OUT option, the output might be lost because the default is the terminal from which you issue the command, and the terminal has limited memory. See [Section 2, Using TMFCOM](#), for information about directing output to an alternate destination.

---

Once file recovery is underway, it continues even if it encounters a damaged record block. In such a case, at the end of file recovery, TMF marks the file as “broken” and as “file recovery needed.” The file recovery summary and disk process operator messages give the location of the audit records that caused the “broken” indication. To

try to recover these files, you can enter the RECOVER FILES command again, using an earlier online dump on which the files might not be damaged.

If a file to be recovered does not exist in the system, you must specify the FROMARCHIVE option. In cases of purged-file recovery, however, use the TOFIRSTPURGE and FROMARCHIVE options (rather than the FROMARCHIVE and TIME options).

---

△ **Caution.** In some cases, you must examine and possibly alter the dump entries in the TMF catalog to ensure that a file recovery will be complete. For instance, when you want to recover an audited file that has been accidentally deleted, you should proceed as follows:

1. Use the INFO DUMPS, DETAIL command to see if the TMF catalog still has an entry for the most recent online dump for the file:
  - If there is no entry for the file, issue an [ADD DUMPS](#) command to replace the entry. Consult the output from a previous [INFO DUMPS](#) command or your hard-copy dump tracking forms to obtain the necessary information.
  - If there is an entry for the file but the dump status is released or invalid, issue an [ALTER DUMPS](#) command with the INVALID OFF and RELEASED OFF options to change the dump status.
2. Issue another INFO DUMPS, DETAIL command to make sure the TMF catalog has a correct entry for an online dump for the file.
3. Issue a RECOVER FILES command with the FROMARCHIVE option and either the TIME option or the TOFIRSTPURGE option and any other options needed to recover the purged file. If you want to recover all files up to the first file purged in a fileset, it is more straightforward and risk-free to use the TOFIRSTPURGE option rather than the TIME option.

Make a new online dump of the file once the file recovery process has completed.

---

Before recovering a file that has been purged or had its AUDIT attribute changed to off, see the discussion on recovering from accidental loss of an audited file in the *TMF Operations and Recovery Guide*.

NonStop SQL files can be recovered by using TMFCOM. Information on the recovery of NonStop SQL files appears in the *SQL/MX Installation and Management Guide*.

On systems with many data volumes (more than 250), or in cases where many files are accessed by a single transaction, the file recovery process (and the backout and volume recovery processes as well) can become suspended because of memory limitations within the process. If this suspension occurs, use the [ALTER PROCESS](#) command to increase the process' EXTENDEDSEGSIZE value and then use TACL to stop the process. The TMP will automatically restart the process with the larger extended segment size you specified.

You can specify only one of the following options in the same RECOVER FILES command: TIME, TOFIRSTPURGE, or TOMATPOSITION. If you enter more than one of these options, an error message is generated.

- 
- △ **Caution.** If you specify the TIME, TOFIRSTPURGE, or TOMATPOSITION option in a RECOVER FILES command, you must immediately make new online dumps of the data files recovered. Without these new dumps, you no longer have file-recovery protection for those files, and subsequent file-recovery operations can fail. In particular, if you later try to use old online dumps to recover files to a point beyond the time that the last RECOVER FILES command was issued, the file-recovery process fails during the redo phase and transmits EMS message 175:

```
Encountered a File Hiatus record for audit file filename at  
audit trail Index #index, SNO #sno, RBA #rba.
```

---

## SERIAL Option Considerations

TMF provides an easy way to recover all files associated with a particular online dump, or a subset of those files. For this operation, enter the RECOVER FILES command with the SERIAL option, specifying the serial number of the dump. As an example, the following command recovers all files for the dump identified by serial number 110:

```
TMF 7> RECOVER FILES *.*.*, SERIAL 110
```

## TIME and TOFIRSTPURGE Option Considerations

When you use the TIME option in a RECOVER FILES command, the value you enter must be greater than the actual dump time by at least one second to ensure a proper recovery; otherwise, TMF recovers no files.

Resetting the system date and time backward (to an earlier time) causes newly generated audit records to appear to have been created earlier than audit records that previously existed. A later file recovery using the TIME option might recover the database to an undesirable state, because file recovery stops when it encounters the first audit record generated after the time specified.

- 
- △ **Caution.** If there is no transaction activity on a system for up to 10 minutes after an online dump of a file was taken, and you perform an undesirable change that either unaudits or purges the file within that duration, a RECOVER FILES command using the TIME or TOFIRSTPURGE option to recover the file to a point within that 10-minute period might not be successful. For Enscribe files, this recovery might not be successful if the change involves any file-label modification records (and not only unaudit or purge operations). The recovery might fail with an error message or it might report that the file was recovered even though it was not. For a successful recovery in this kind of situation, use the TOMATPOSITION option in the RECOVER FILES command, specifying an RBA before the undesirable change was made.
- 

Under certain very unusual circumstances, file recovery with the TIME option might not succeed and an event message will be logged to report the error. Such a situation would occur, for example, if the AUDIT attribute was turned off immediately following an online dump, and the recovery was then attempted. Because no timestamp would exist in any record following the Online Dump End record, recovery to a specified time

after the dump would be impossible. In such a case, you can use the TOMATPOSITION option to recover the database to a specific location within the master audit trail (MAT) rather than to a specific time. However, the recovery steps involve the use of the SNOOP utility, a high level of TMF background, and perhaps the assistance of the Global Customer Support Center (GCSC) or your service provider. For more information about the procedure, see *TMF Operations and Recovery Guide*.

## TOMATPOSITION Option Considerations

The TOMATPOSITION option applies only to the MAT, not to the auxiliary audit trails. The TOMATPOSITION option is also used to support the takeover operation available with the NonStop RDF product, as described in *RDF/IMP, IMPX, and ZLT System Management Manual*.

Specifying the incorrect TOMATPOSTION location has the same effect as specifying the incorrect TIME value: the data files recovered will be consistent among themselves, but they might not be consistent with the other data files on the system.

When you specify the TOMATPOSITION option, the most recent valid online dump is used for recovery. If the TOMATPOSITION value specifies a relative byte address (rba) written before the rba where this online dump completed, file recovery fails after the dump is restored and EMS message 179 is generated:

```
The Audit Trail sequence number or the RBA is too low for the
online dump that is being recovered:
atseqno atseqno RBA rba.
```

To recover to an rba written before the most recent valid online dump completed, you must first invalidate the online dumps that occurred after that location, by using the [ALTER DUMPS](#) command with the INVALID ON option. If all the files you wish to recover are from a single online dump, you can use the SERIAL option of the RECOVER FILES command instead of invalidating all the later online dumps

Under very unusual circumstances, file recovery with the TOMATPOSITION option may report that files were recovered when in fact they were not. If the TOMATPOSITION value specified an rba that was written immediately before the most recent online dump completed, the file recovery process may not detect the invalid location. To avoid this problem, specify only TOMATPOSITION rbas that occur after the most recent valid online dump was completed.

## MAP NAMES Option Considerations

The MAP NAMES option allows you to recover files to different volumes, subvolumes, and file IDs than the ones from which they were dumped.

In addition to performing production file recovery with MAP NAMES, you can use this option to practice or rehearse file recovery without actually overwriting existing data. Furthermore, MAP NAMES gives you an alternative way to replicate data by using a second copy of your file sets.

Because the MAP NAMES option restores files only to the same instance of TMF, you cannot use this option to restore these files to another system.

When you use the MAP NAMES option, the files in the new locations have the same user IDs and attributes as those in the old locations.

When you use the MAP NAMES option, the FROMARCHIVE option is also used in all cases, even if you do not explicitly select it. For recovery to a new location, Enscribe files are purged and restored before recovery is started. However, you should create SQL objects before recovery is started to keep the catalog consistent.

---

**Note.** If you specify the MAP NAMES option, you cannot use the RECOVERVDPHASE1 option.

---

The MAP NAMES option does not correct interdependencies between files. It does not add SQL objects into a SQL catalog; instead, you must manually add the new tables to the SQL catalog you wish. File partitions are treated as distinct files; thus, recovered file partitions must be manually updated to reflect any changes.

---

△ **Caution.** If you use the MAP NAMES option of the RECOVER FILES command to recover files to a new location, you must immediately make new online dumps of the target data files recovered. Without these new dumps, you will not have file-recovery protection for those files, and subsequent file-recovery operations can fail. In particular, if you later try to use old online dumps of the target files to recover the target files to a point beyond the time that the last RECOVER FILES command was issued, the file-recovery process fails during the redo phase and transmits EMS message 175:

```
Encountered a File Hiatus record for audit file filename at
audit trail Index #index, SNO #sno, RBA #rba.
```

---

## Options for the SMF Product

The TOPHYSVOL and WHEREPHYSVOLIS parameters allow you to refer to physical disk volumes managed by the NonStop Storage Management Foundation (SMF) product. The SMF software enhances the automatic management of system storage. A file stored on an SMF virtual disk is identified by a logical file name. A file that is not associated with an SMF virtual disk process, however, is identified by a direct file name. A file has either a logical or a direct file name, but it never has both. Logical and direct file names are simply the external file names that applications use to identify files either managed by the SMF product or not managed by it, respectively. The DP2 disk process, however, does not use external file names; instead, that process identifies a file by a physical file name, which is simply an internal file name that maps to the file's logical or direct file name. As an example, for a file stored on an SMF virtual disk, the logical file name might be \SYS.\$L1.SOME.DATA and the physical file name might be \$PHYS27.ZYS00000.A0000K29. For a file that is not managed by the SMF software, the direct file name and physical file name are both the same (for instance, \SYS.\$P1.MORE.DATA).

For systems using the SMF product, the TMF catalog contains the logical file names of files managed by SMF and the direct file names of files not managed by SMF. Thus,

when taking dumps of these files, or referring to them in the RECOVER FILES command, you must refer to these files by their logical or direct file names, not by their physical file names. Nevertheless, with the TOPHYSVOL parameter, you can identify the physical target disk volume where you wish TMF to place the logical recovered files. Furthermore, with the WHEREPHYSVOLIS parameter, you can recover all the logical files on a specified physical volume. The TOPHYSVOL and WHEREPHYSVOLIS options are mutually exclusive: if you specify both of these options, WHEREPHYSVOLIS takes precedence and TOPHYSVOL is ignored.

---

△ **Caution.** The TOPHYSVOL option does not work when direct file names appear in *file-set* (in other words, with files not managed by the SMF product).

---

It is important to remain aware of the free space available on the destination physical volume; if the physical volume becomes full, the file recovery process aborts.

If a physical disk that contained a virtual disk process (VDP) catalog fails, you must recover the files in two steps:

1. Issue a RECOVER FILES command with the RECOVERVDPHPHASE1 option to recover the VDP catalog.
2. Issue a RECOVER FILES command, without the RECOVERVDPHPHASE1 option, to recover the files.

For more information about the SMF product, see *Storage Management Foundation User's Guide*.

## Output Displayed

As file recovery progresses, TMFCOM displays EMS messages about various events that take place. The format and content of these messages varies. Examples appear along with the command-entry examples.

## Examples

The following RECOVER FILES command recovers the audited file T1, residing on the disk volume \$DATA02 and the subvolume TMF. The command is entered interactively and, because the NOWAIT option is not specified, the resulting EMS messages are displayed on the terminal as the recovery progresses:

```
TMF 71> RECOVER FILES $DATA02.TMF.T1, FROMARCHIVE
Recover Files operation [5] begun.
Operation TimeStamp: 30-Dec-2003 10:57:06
Event Messages:
10:57 30DEC03 195,00,082 NonStop TMF on \:PLUTO *0163* RECOVER FILES [5]
          FileRecovList Process #1: Files to Recover:
          \PLUTO.$DATA02.TMF.T1
10:57 30DEC03 195,00,084 NonStop TMF on \:PLUTO *0401* RECOVER FILES [5]
          FileRecovery Process #1: Phase 1 of recovery completed
          at atseqno 2 RBA 37645324.
10:57 30DEC03 195,00,084 NonStop TMF on \:PLUTO *0403* RECOVER FILES [5]
          FileRecovery Process #1: Phase 3 of recovery completed
          at atseqno 999999 RBA 2147483647.
10:57 30DEC03 195,00,084 NonStop TMF on \:PLUTO *0164* RECOVER FILES [5]
          FileRecovery Process #1: Recovered Files:
          \PLUTO.$DATA02.TMF.T1
```

```

10:57 30DEC03 195,00,012 NonStop TMF on \:PLUTO *0165* RECOVER FILES [5] TMP
        Process FileRecovMgmt: Files Recovered: 1; Files Not
        Recovered: 0.
10:57 30DEC03 195,00,012 NonStop TMF on \:PLUTO *0156* RECOVER FILES [5] TMP
        Process: End operation.
TMF 72>

```

The next RECOVER FILES command initiates the recovery process interactively at a terminal. A moment later the BREAK key is pressed, terminating the display. The file recovery operation continues, but events are no longer reported at the terminal. (The operator can use the DISPLAY OPERATION command later to display the events for the operation.)

```

TMF 72> RECOVER FILES $DATA02.TMF.*
Recover Files operation [14] begun.
Operation TimeStamp: 5-Jan-2002 10:17:45
*** Display of RECOVER FILES operation [14] terminated by break ***
~

```

In the next command, the NOWAIT option is specified; events are not reported at the terminal, but are written instead to the file named EVENTS on the default volume and subvolume. TMFCOM prompts at once for another command:

```

TMF 73> RECOVER /OUT EVENTS/ FILES $DATA02.TMF.*, NOWAIT
TMF 74>

```

You can check the status of the recovery operation by issuing a STATUS OPERATIONS command.

The next RECOVER FILES command recovers all dump files belonging to the dump identified by Serial Number 98:

```

TMF 74> RECOVER *.*.*, SERIAL 98

```

This RECOVER FILES command illustrates how to recover SMF files to a volume other than the current physical location of the files:

```

TMF 75> RECOVER FILES $L1.*.*, TOPHYSVOL $D1, FROMARCHIVE
.
.
.

```

The following RECOVER FILES command illustrates the recovery of all audited logical (SMF) files located on the logical volumes \$L1, \$L2, \$L3, and \$L4, which reside on the physical volume \$P7:

```

TMF 76> RECOVER FILES ($L1.*.*, $L2.*.*, $L3.*.*, $L4.*.*), WHEREPHYSVOLIS $P7, &
>>>FROMARCHIVE
.
.
.

```

The next RECOVER FILES command uses the TOMATPOSITION option to recover the files on the \$DATA04 volume to the rba 2001 in the Audit-Trail File AA000150:

```

TMF 77> RECOVER FILES $DATA04.*.*, TOMATPOSITION (150, 2001), &
>>>FROMARCHIVE
.
.

```



Here is a RECOVER FILES command that uses the MAP NAMES option to map all files being recovered to a different location. This example also presents the EMS messages generated by this operation:

```
TMF 78> RECOVER FILES ($DATA09.AMFRNL2.*, $DATA10.AMFRNL2.*, &
>>>$DATA11.AMFRNL2.*, $DATA12.AMFRNL2.*, $DATA06.AMFRNL2.*), MAP NAMES &
>>>($DATA09.AMFRNL2.* TO $DATA13.AMFRNL2.*, &
>>>$DATA10.AMFRNL2.* TO $DATA14.AMFRNL2.*, &
>>>$DATA11.AMFRNL2.* TO $DATA15.AMFRNL2.*, &
>>>$DATA12.AMFRNL2.* TO $DATA16.AMFRNL2.*, &
>>>$DATA06.AMFRNL2.* TO $DATA17.AMFRNL2.* )
Recover Files operation [ 17 ] begun.
Operation TimeStamp: 21-Aug-2003 17:51:50
Event Messages:
17:51 21AUG03 195,00,203 NonStop TMF on \PLUTO *0163* RECOVER FILES [17]
FileRecovList Process #1: Files to Recover:
\PLUTO.$DATA06.AMFRNL2.ODETAIL TO
\PLUTO.$DATA17.AMFRNL2.ODETAIL,
\PLUTO.$DATA06.AMFRNL2.PARTLOC TO
\PLUTO.$DATA17.AMFRNL2.PARTLOC,
\PLUTO.$DATA09.AMFRNL2.ORDERS TO
\PLUTO.$DATA13.AMFRNL2.ORDERS,
\PLUTO.$DATA09.AMFRNL2.PARTLOC TO
\PLUTO.$DATA13.AMFRNL2.PARTLOC,
\PLUTO.$DATA09.AMFRNL2.REPORDS TO
\PLUTO.$DATA13.AMFRNL2.REPORDS,
\PLUTO.$DATA10.AMFRNL2.MYORDER TO
\PLUTO.$DATA14.AMFRNL2.MYORDER,
\PLUTO.$DATA11.AMFRNL2.XORDCUS TO
\PLUTO.$DATA15.AMFRNL2.XORDCUS,
\PLUTO.$DATA12.AMFRNL2.PARTLOC TO
\PLUTO.$DATA16.AMFRNL2.PARTLOC
17:51 21AUG03 195,00,200 NonStop TMF on \PLUTO *0401* RECOVER FILES [17]
FileRecovery Process #1: Phase 1 of recovery
completed at atseqno 2 RBA 4436896.
17:51 21AUG03 195,00,200 NonStop TMF on \PLUTO *0402* RECOVER FILES [17]
FileRecovery Process #1: Phase 2 of file recovery
completed at atseqno 2 RBA 4436788.
17:51 21AUG03 195,00,200 NonStop TMF on \PLUTO *0403* RECOVER FILES [17]
FileRecovery Process #1: Phase 3 of recovery
completed at atseqno 2 RBA 4437908.
17:52 21AUG03 195,00,200 NonStop TMF on \PLUTO *0164* RECOVER FILES [17]
FileRecovery Process #1: Recovered Files:
\PLUTO.$DATA17.AMFRNL2.PARTLOC,
\PLUTO.$DATA17.AMFRNL2.ODETAIL,
\PLUTO.$DATA16.AMFRNL2.PARTLOC,
\PLUTO.$DATA15.AMFRNL2.XORDCUS,
\PLUTO.$DATA14.AMFRNL2.MYORDER,
\PLUTO.$DATA13.AMFRNL2.REPORDS,
\PLUTO.$DATA13.AMFRNL2.PARTLOC,
\PLUTO.$DATA13.AMFRNL2.ORDERS
17:52 21AUG03 195,00,012 NonStop TMF on \PLUTO *0165* RECOVER FILES [17] TMP
Process FileRecovMgmt: Files Recovered: 8; Files Not
Recovered: 0.
17:52 21AUG03 195,00,012 NonStop TMF on \PLUTO *0156* RECOVER FILES [17] TMP
Process: End operation.
```

This next RECOVER FILES operation recovers ten files, restoring two of them (\PLUTO.\$DATA06.AMFRNL3.A1 and \PLUTO.\$DATA06.AMFRNL3.A2) to the same location and the remaining eight to a different location:

```
TMF 79> RECOVER FILES ($DATA09.AMFRNL2.*, $DATA06.AMFRNL3.*, &
>>>$DATA10.AMFRNL2.*, $DATA11.AMFRNL2.*, &
>>>$DATA12.AMFRNL2.*, $DATA06.AMFRNL2.*), MAP NAMES &
>>>($DATA09.AMFRNL2.* TO $DATA13.AMFRNL2.*, &
>>>$DATA10.AMFRNL2.* TO $DATA14.AMFRNL2.*, &
>>>$DATA11.AMFRNL2.* TO $DATA15.AMFRNL2.*, &
>>>$DATA12.AMFRNL2.* TO $DATA16.AMFRNL2.*, &
>>>$DATA06.AMFRNL2.* TO $DATA17.AMFRNL2.* )
Recover Files operation [ 20 ] begun.
Operation TimeStamp: 21-Aug-2003 18:00:55
Event Messages:
18:00 21AUG03 195,00,209 NonStop TMF on \PLUTO *0163* RECOVER FILES [20]
FileRecovList Process #1: Files to Recover:
\PLUTO.$DATA06.AMFRNL2.ODETAIL TO
\PLUTO.$DATA17.AMFRNL2.ODETAIL,
\PLUTO.$DATA06.AMFRNL2.PARTLOC TO
\PLUTO.$DATA17.AMFRNL2.PARTLOC,
\PLUTO.$DATA06.AMFRNL3.A1,
\PLUTO.$DATA06.AMFRNL3.A2,
\PLUTO.$DATA09.AMFRNL2.ORDERS TO
\PLUTO.$DATA13.AMFRNL2.ORDERS,
\PLUTO.$DATA09.AMFRNL2.PARTLOC TO
\PLUTO.$DATA13.AMFRNL2.PARTLOC,
\PLUTO.$DATA09.AMFRNL2.REPORDS TO
\PLUTO.$DATA13.AMFRNL2.REPORDS,
\PLUTO.$DATA10.AMFRNL2.MYORDER TO
\PLUTO.$DATA14.AMFRNL2.MYORDER,
\PLUTO.$DATA11.AMFRNL2.XORDCUS TO
\PLUTO.$DATA15.AMFRNL2.XORDCUS,
\PLUTO.$DATA12.AMFRNL2.PARTLOC TO
\PLUTO.$DATA16.AMFRNL2.PARTLOC
18:00 21AUG03 195,00,112 NonStop TMF on \PLUTO *0401* RECOVER FILES [20]
FileRecovery Process #1: Phase 1 of recovery
completed at atseqno 2 RBA 4478944.
18:00 21AUG03 195,00,112 NonStop TMF on \PLUTO *0402* RECOVER FILES [20]
FileRecovery Process #1: Phase 2 of file recovery
completed at atseqno 2 RBA 4478836.
18:00 21AUG03 195,00,112 NonStop TMF on \PLUTO *0403* RECOVER FILES [20]
FileRecovery Process #1: Phase 3 of recovery
completed at atseqno 2 RBA 4479956.
18:01 21AUG03 195,00,112 NonStop TMF on \PLUTO *0164* RECOVER FILES [20]
FileRecovery Process #1: Recovered Files:
\PLUTO.$DATA17.AMFRNL2.PARTLOC,
\PLUTO.$DATA17.AMFRNL2.ODETAIL,
\PLUTO.$DATA16.AMFRNL2.PARTLOC,
\PLUTO.$DATA15.AMFRNL2.XORDCUS,
\PLUTO.$DATA14.AMFRNL2.MYORDER,
\PLUTO.$DATA13.AMFRNL2.REPORDS,
\PLUTO.$DATA13.AMFRNL2.PARTLOC,
\PLUTO.$DATA13.AMFRNL2.ORDERS,
\PLUTO.$DATA06.AMFRNL3.A2, \PLUTO.$DATA06.AMFRNL3.A1
18:01 21AUG03 195,00,012 NonStop TMF on \PLUTO *0165* RECOVER FILES [20] TMP
Process FileRecovMgmt: Files Recovered:10; Files Not
Recovered: 0.
18:01 21AUG03 195,00,012 NonStop TMF on \PLUTO *0156* RECOVER FILES [20] TMP
Process: End operation.
```

# RELOCATE DISKDUMPS

The RELOCATE DISKDUMPS command changes the disk dump media locations specified in the TMF catalog.

```
RELOCATE DISKDUMPS      {serial-number
                        {(serial-number [, serial-number]...)}
                        {*}}
      ,OLDDMLLOC diskmedia-loc, NEWDMLOC diskmedia-loc
```

*serial-number*

directs this RELOCATE DISKDUMPS command to change the catalog entries for only the dumps with the specified serial number. You can obtain information about dump serial numbers by using the [INFO DUMPS](#) command.

\*

directs this RELOCATE DISKDUMPS command to change the catalog entries for all dumps in the location specified by OLDDMLLOC.

OLDDMLLOC *diskmedia-loc*

specifies the old disk media location for the dumps, as listed in the TMF catalog. *diskmedia-loc* must be a valid disk media location. Specify one disk media location in the following format:

```
{[node.]{volume | *}}
```

An asterisk (\*) indicates “all”; thus, *node.\** indicates all volumes. If you do not specify the node name, the node where your TMFCOM process is running is used.

NEWDMLOC *diskmedia-loc*

specifies the new disk media location for the dumps. *diskmedia-loc* must be a valid disk media location. Specify one disk media location in the following format:

```
{[node.]{volume | *}}
```

An asterisk (\*) indicates “all”; thus, *node.\** indicates all volumes. If you do not specify the node name, the node where your TMFCOM process is running is used.

## Security Restrictions

You can issue the RELOCATE DISKDUMPS command only if you are a member of the super user group.

## TMF State Requirement

Before you can use the RELOCATE DISKDUMPS command, TMF must be started.

## Usage Guidelines

After moving or restoring dumps to a different location, use the RELOCATE DISKDUMPS command to change the entries for the dumps in the TMF catalog. This command selects disk media names in the catalog that match the serial numbers and OLDDMLOC value specified, and changes their location to the NEWDMLOC value.

When the catalog entries are updated with the new locations, the dumps will be available for recovery purposes.

If you use the wild-card character (\*) in the OLDDMLOC parameter, you must also use \* for the NEWDMLOC parameter.

You can use this command for any reason you might have to move or restore dumps to a different location. As an example, you might want to move dumps from one disk to another just to free some space on the current disk. In doing this, you first copy the dumps to the required disk volume; then, you issue the RELOCATE DISKDUMPS command to modify the TMF catalog to reflect the relocation of the dumps; finally, you purge the dumps from the old volume.

## Example

The following RELOCATE DISKDUMPS command changes all catalog entries for dumps made on the primary system (\PRI) to indicate that they now reside on the remote system (\REM).

```
TMF 80> RELOCATE DISKDUMPS *, OLDDMLOC \PRI.*, NEWDMLOC \REM.*
```

# RESET AUDITTRAIL

The RESET AUDITTRAIL command resets audit trail attributes to their default values, canceling the effects of any previous [SET AUDITTRAIL](#) commands. Values changed by RESET AUDITTRAIL are local to your current session and only apply to [ADD AUDITTRAIL](#) commands subsequently issued within this session.

```
RESET AUDITTRAIL {attribute [,attribute...]...}
                  {
                    *
                  }
```

where *attribute* is any of the following:

```
[AUDITDUMP           ]
[FILESIZE            ]
[ACTIVEVOL[S]        ]
[FILESPERVOLUME      ]
[OVERFLOWVOL[S]      ]
[OVERFLOWTHRESHOLD   ]
[RESTOREVOL[S]        ]
[BEGINTRANSDISABLE   ]
```

## AUDITDUMP

resets the AUDITDUMP attribute to ON, so that dumping takes place.

## FILESIZE

resets the FILESIZE attribute to 100, reflecting an audit-trail file size of 100 megabytes (100 x 1,048,576 bytes).

## ACTIVEVOL[S]

removes any ACTIVEVOL[S] specification; thus, no active-audit volumes are identified to hold the audit trail. Because active-audit volumes are required for the audit trail configuration, you must specify the active-audit volumes in the ACTIVEVOL[S] parameter of a new [ADD AUDITTRAIL](#) or [SET AUDITTRAIL](#) command.

## FILESPERVOLUME

resets the FILESPERVOLUME attribute to 5, denoting that each active-audit volume configured on disk can hold five audit-trail files.

## OVERFLOWVOL[S]

removes any OVERFLOWVOL[S] specification; thus, no overflow-audit volumes are identified to hold the overflow. This attribute is not required and has no default. However, if you do not specify an overflow-audit volume in the OVERFLOWVOL[S] parameter of a new [ADD AUDITTRAIL](#) or [SET AUDITTRAIL](#) command, TMF generates an EMS warning event message when you add the audit trail.

**OVERFLOWTHRESHOLD**

resets the OVERFLOWTHRESHOLD attribute to 80 (if an overflow volume is configured), indicating that overflow copying begins when the audit trail capacity in use exceeds 80%.

**RESTOREVOL [ S ]**

removes any RESTOREVOL[S] specification; thus, no restore-audit volumes onto which audit dumps can be restored are identified. This attribute has no default but is required if the AUDITDUMP attribute is ON. Thus, if AUDITDUMP is ON, you must specify a restore-audit volume in the RESTOREVOL[S] parameter of a new [ADD AUDITTRAIL](#) or [SET AUDITTRAIL](#) command.

**BEGINTRANSDISABLE**

resets the BEGINTRANSDISABLE attribute to 90, indicating that new transactions are disabled when the audit trail capacity in use reaches 90%.

\*

resets all audit trail attributes to their default values.

## Security Restrictions

Anyone can issue the RESET AUDITTRAIL command.

## TMF State Requirement

You can enter the RESET AUDITTRAIL command at any time, whether or not TMF has been started.

## Usage Guidelines

The values specified in RESET AUDITTRAIL are local to your current session only and are not propagated to TMFSERVE. These values take effect when you issue subsequent [ADD AUDITTRAIL](#) commands within your session and cease to apply when your session ends.

You might want to use RESET AUDITTRAIL when you expect to issue several subsequent [ADD AUDITTRAIL](#) commands, using the default values for certain attributes in most instances and overriding the effects of previous [SET AUDITTRAIL](#) commands (as discussed later in this section). In this way, the RESET AUDITTRAIL values are carried forward to all the [ADD AUDITTRAIL](#) commands in your session, so that you need to enter only specific nondefault values as you wish.

## Examples

The following RESET AUDITTRAIL command resets the BEGINTRANSDISABLE attribute to 90%, overriding an earlier SET command that established a different value for this attribute:

```
TMF 81> RESET AUDITTRAIL BEGINTRANSDISABLE
```

Now, all subsequent ADD AUDITTRAIL commands that do not include the BEGINTRANSDISABLE parameter will use this 90% value for that attribute.

```
TMF 82> ADD AUDITTRAIL MAT, FILESPERVOLUME 6, &  
>>>ACTIVEVOLS ($MAT1, $MAT2), &  
>>>OVERFLOWVOLS ($DATA1, $DATA2), &  
>>>RESTOREVOLS $DATA1  
TMF 83> ADD AUDITTRAIL AUX01, FILESPERVOLUME 4, &  
>>>ACTIVEVOLS $AUX011, &  
>>>OVERFLOWVOLS $DATA3, &  
>>>RESTOREVOLS $DATA3  
TMF 84> .  
      .  
      .
```

The next RESET AUDITTRAIL command resets all audit trail attributes to their default values:

```
TMF 84> RESET AUDITTRAIL *
```

# RESET DATAVOLS

The RESET DATAVOLS command resets data volume attributes to their default values, canceling the effects of any previous [SET DATAVOLS](#) commands. Values changed by RESET DATAVOLS are local to your current session and apply only to [ADD DATAVOLS](#) commands subsequently issued within this session.

```
RESET DATAVOL[S] { attribute [, attribute] ... }
                  { * }
```

where *attribute* is either of the following:

```
[AUDITTRAIL ]
[RECOVERYMODE]
```

## AUDITTRAIL

resets the AUDITTRAIL attribute to MASTER, indicating that the data volume sends its audit records to the master audit trail.

## RECOVERYMODE

resets the RECOVERYMODE attribute to ONLINE, indicating that TMF retains sufficient audit trails on disk so that volume recovery does not need to restore audit information from an archived audit dump.

\*

resets all data volume attributes to their default values.

## Security Restrictions

Anyone can issue the RESET DATAVOLS command.

## TMF State Requirement

You can enter the RESET DATAVOLS command at any time, whether or not TMF has been started.

## Usage Guidelines

The values specified in RESET DATAVOLS are local to your current session only, and are not propagated to TMFSERVE. These values take effect when you issue subsequent [ADD DATAVOLS](#) commands within your session and cease to apply when your session ends.

You might want to use RESET DATAVOLS when you expect to issue several subsequent [ADD DATAVOLS](#) commands, using the default values for certain attributes in most instances and overriding the effects of previous [SET DATAVOLS](#) commands (as discussed later in this section). In this way, the RESET DATAVOLS values are



carried forward to all the [ADD DATAVOLS](#) commands in your session, so that you need to enter only specific nondefault values as you wish.

## Examples

The following RESET DATAVOLS command resets the RECOVERYMODE attribute to ONLINE, overriding an earlier SET command that established the ARCHIVE value for this attribute:

```
TMF 86> RESET DATAVOLS RECOVERYMODE
```

As a result, all subsequent [ADD DATAVOLS](#) commands that do not include the RECOVERYMODE parameter will use the ONLINE value for that attribute:

```
TMF 87> ADD DATAVOLS $DATA1, AUDITTRAIL AUX01
TMF 88> ADD DATAVOLS $DATA2, AUDITTRAIL AUX02
TMF 89> .
      .
      .
```

The next RESET DATAVOLS command resets all data volume attributes to their default values:

```
TMF 89> RESET DATAVOLS *
```

# RESET DUMPS

The RESET DUMPS command removes (clears) online and audit dump specifications, canceling the effects of previous [SET DUMPS](#) commands. This clearing by RESET DUMPS is local to your current session and affects only [ADD DUMPS](#) commands subsequently issued within this session.

```
RESET DUMPS {attribute [, attribute...]...}  
            { * }
```

where *attribute* is:

```
[DISKMEDIA]  
[SERIAL   ]  
[TAPEMEDIA]  
[TYPE     ]
```

## DISKMEDIA

removes any DISKMEDIA specification; thus, no disk volume is identified on which the online dump is stored.

## SERIAL

removes any SERIAL specification; thus, no serial number is identified for the dump added to the catalog.

## TAPEMEDIA

removes any TAPEMEDIA specification; thus, no tape volume is identified on which the online dump is stored.

## TYPE

removes the TYPE specification; thus, the type of dump (online or audit) is not identified.

\*

removes all dump specifications controlled by RESET DUMP (that is, the DISKMEDIA, SERIAL, TAPEMEDIA, and TYPE specifications).

## Security Restrictions

Anyone can issue the RESET DUMPS command.

## TMF State Requirement

You can enter the RESET DUMPS command at any time, whether or not TMF has been started.

## Usage Guidelines

The clearing of specifications by RESET DUMPS is local to your current session only, and is not propagated to TMFSERVE. This clearing affects only [ADD DUMPS](#) commands issued subsequently within your session and ceases to apply when your session ends.

---

**Note.** You must establish values for the SERIAL and TYPE entries and for either the DISKMEDIA or the TAPEMEDIA entry. Default values are not allowed for these entries. Therefore, if you enter a RESET DUMPS command for any of these entries, you must subsequently respecify them in a new [ADD DUMPS](#) or [SET DUMPS](#) command.

---

If TAPEMEDIA is accidentally set for a dump to disk operation, enter RESET DUMPS TAPEMEDIA to clear it.

If DISKMEDIA is accidentally set for a dump to tape operation, enter RESET DUMPS DISKMEDIA to clear it.

## Example

The following RESET DUMPS command clears the SERIAL specification:

```
TMF 91> RESET DUMPS SERIAL
```

The next RESET DUMPS command clears all dump specifications:

```
TMF 92> RESET DUMPS *
```

# RESET TAPEMEDIA

The RESET TAPEMEDIA command removes certain tape volume attributes, or resets others to their defaults, canceling the effects of previous [SET TAPEMEDIA](#) commands. These changes by RESET TAPEMEDIA are local to your current session, and affect only [ADD TAPEMEDIA](#) commands subsequently issued within the session.

```
RESET TAPEMEDIA {attribute [, attribute]...}
                { * }
```

where *attribute* is:

```
[ DENSITY ]
[ DRIVE   ]
[ LABEL   ]
[ STATUS  ]
[ UNLOAD  ]
```

## DENSITY

removes any DENSITY specification; thus, no density is specified at which the tapes are written. See [Section 4, TMFCOM Tape Media Considerations](#), for the appropriate tape density setting for the tape drive you are using.

## DRIVE

removes the DRIVE specification; thus, no tape drive is specified for labeling tape volumes. This specification has no default but is required if the LABEL specification is ON. Thus, if LABEL is ON, you must specify the tape drive in the DRIVE parameter of a new [ADD TAPEMEDIA](#) or [SET TAPEMEDIA](#) command.

## LABEL

resets the LABEL specification to OFF, indicating that tape volume labeling does not occur.

## STATUS

removes any STATUS specification; thus, tape volume status (ASSIGNED, BAD, RELEASED, or SCRATCH) is not specified. This specification has no default but is required if the LABEL specification is ON. Thus, if LABEL is ON, you must specify the tape volume status in the STATUS parameter of a new [ADD TAPEMEDIA](#) or [SET TAPEMEDIA](#) command.

## UNLOAD

resets the UNLOAD specification to ON, indicating that the tape volume is unloaded after it is labeled.

\*

resets all tape volume specifications to their default values and clears all tape volume specifications that have no defaults.

## Security Restrictions

Anyone can issue the RESET TAPEMEDIA command.

## TMF State Requirement

You can enter the RESET TAPEMEDIA command at any time, whether or not TMF has been started.

## Usage Guidelines

The resetting or clearing of specifications by RESET TAPEMEDIA is local to your current session only, and is not propagated to TMFSERVE. These changes take effect when you issue subsequent [ADD TAPEMEDIA](#) commands within your session and cease to apply when your session ends.

If TAPEMEDIA is accidentally set for a dump to disk operation, enter RESET DUMPS TAPEMEDIA to clear it.

If DISKMEDIA is accidentally set for a dump to tape operation, enter RESET DUMPS DISKMEDIA to clear it.

## Example

The following RESET TAPEMEDIA command resets the LABEL specification to OFF, overriding an earlier SET command that established the ON value for this specification:

```
TMF 95> RESET TAPEMEDIA LABEL
```

All subsequent ADD TAPEMEDIA commands that do not include the LABEL parameter will use the OFF value for that specification:

```
TMF 96> ADD TAPEMEDIA TT05, DRIVE $TAPE1
TMF 97> ADD TAPEMEDIA TT06, DRIVE $TAPE2
TMF 98> .
      .
      .
```

The next RESET TAPEMEDIA command resets or clears all tape volume specifications, as applicable:

```
TMF 98> RESET TAPEMEDIA *
```

# RESOLVE TRANSACTION

The RESOLVE TRANSACTION command causes an unresolved distributed transaction to either commit or back out its changes. To fully resolve the transaction, this command must be issued on all nodes on which the transaction resides, except for the transaction's home node.

```
RESOLVE TRANS[ACTION[S]]

    {transaction-id
    {(transaction-id [,transaction-id]...)}

    ,STATE {ABORTED | COMMITTED}

where transaction-id is:

    [node .                               ]
    [node-number .                       ]
    [node(tm-flags) .                     ] cpu.seq-num
    [node-number(tm-flags) .             ]
    [(tm-flags) .                         ]
```

*transaction-id*

identifies the distributed transaction to be resolved:

*node*

is the name of the home node for the transaction, which is the node (system) on which the transaction originated. The syntax rules for specifying a node name appear under [Disk File Names](#), earlier in this section. The default for *node* is the node on which the TMFSERVE process communicating with your TMFCOM process is running.

*node-number*

is the number of the home node for the transaction, which is the node (system) on which the transaction originated. A node number specification consists of a backslash followed by the number assigned to the node. The default for *node-number* is the node on which the TMFSERVE process communicating with your TMFCOM process is running.

*tm-flags*

is a number representing flags used internally by TMF. If this number is zero, it does not appear in displays of the *transaction-id* presented by the [STATUS TRANSACTIONS](#) command.

---

**Note.** If the *tm-flags* value is a number other than zero, you must include this number in parentheses in the RESOLVE TRANSACTION command. For example:

---

TMF 3> **RESOLVE TRANSACTION \SYSB(1).3.45678**

---

*cpu*

is the number of the processor on which the transaction originated.

*seq-num*

is a sequence number assigned by TMF to the transaction, which distinguishes it from other transactions originating on the same processor on the same home node.

STATE {ABORTED | COMMITTED}

identifies the target state for the unresolved transaction. ABORTED aborts the transaction. COMMITTED commits the transaction. For information about when to use each of these options, see [Usage Guidelines](#), later in this discussion.

## Security Restrictions

You can issue the RESOLVE TRANSACTION command only if you are a member of the super user group.

## TMF State Requirement

Before you can enter the RESOLVE TRANSACTION command, TMF must be started or in the process of starting.

## Usage Guidelines

The RESOLVE TRANSACTION command is used to resolve distributed transactions only.

When the communication lines between nodes fail, TMF is sometimes left with transactions that it cannot determine whether to commit or abort. The only transactions that might cause such confusion are those in the prepared state at remote nodes; they remain in this state until either the connection between the home and remote nodes is reestablished or the transactions are resolved by the operator. (A distributed transaction enters the prepared state on a remote node after it completes phase 1 of its commitment and its transaction state (prepared) record is written to the

audit trail. For a complete discussion of all transaction states, see the [STATUS TRANSACTIONS](#) command description later in this section.)

If the connection between the nodes is reestablished, TMF automatically resolves the transactions that are in doubt. If communications cannot be reestablished, however, you may need to resolve the transactions manually. For transactions in the prepared state at a remote node, you can do this by issuing the RESOLVE TRANSACTION command at that node.

Before you issue the RESOLVE TRANSACTION command, however, determine the transaction's status by entering a [STATUS TRANSACTIONS](#) command at both the home and remote nodes of the transaction. Be certain that you fully understand the transaction's status at both nodes before proceeding; no action at the home node, however, is required for resolving unresolved distributed transactions. [Table 3-17](#) shows what action you should take at the remote node, based on the status of the transaction at the home node.

---

**Table 3-17. Resolving Distributed Transactions**

| Transaction State at Home Node | Action to Take at Remote Node  |
|--------------------------------|--|
| Aborted                        | Issue the RESOLVE TRANSACTION command with the STATE ABORTED option.   |
| Aborting                       | Issue the RESOLVE TRANSACTION command with the STATE ABORTED option.   |
| Committed                      | Issue the RESOLVE TRANSACTION command with the STATE COMMITTED option. |
| nonexistent                    | Issue the RESOLVE TRANSACTION command with the STATE ABORTED option.   |

---

While the status of a distributed transaction is active or prepared at the home node, its final outcome cannot be determined, and the RESOLVE TRANSACTION command should not be used to resolve it.

---

△ **Caution.** If you are uncertain about the results or ramifications of the RESOLVE TRANSACTION command, contact the Global Customer Support Center (GCSC) or your HP service provider before issuing this command.

Using the RESOLVE TRANSACTION command to resolve a transaction that is in any state other than those listed in [Table 3-17](#) will result in an inconsistent database.

---

See the [STATUS TRANSACTIONS](#) command discussion in this manual and the information on distributed transactions in the *TMF Operations and Recovery Guide* for more information about resolving distributed transactions, and for a discussion of transaction states.



## Examples

The following RESOLVE TRANSACTION command aborts the specified transaction, also displaying information about the transaction's status:

```
TMF 4> RESOLVE TRANSACTION \NEWYORK.6.5, STATE ABORTED
      \NEWYORK.6.5, State: aborted
      Distributed, Parent: \NEWYORK
```

A STATUS TRANSACTIONS command has revealed that for the transaction \SANFRAN.7.8932, the *tm-flags* value was set to 1. Because the *tm-flags* value was a non-zero number, this value must appear in any RESOLVE TRANSACTION command to resolve this transaction:

```
TMF 5> RESOLVE TRANSACTION \SANFRAN(1).7.8932, STATE COMMITTED
      \SANFRAN(1).7.8932, State: committed
      Distributed, Parent: \SANFRAN
```

# SET AUDITTRAIL

The SET AUDITTRAIL command establishes attribute values for audit trails created by subsequent [ADD AUDITTRAIL](#) commands issued during the same TMFCOM session.

```
SET AUDITTRAIL attribute [,attribute]...
```

where *attribute* is any of the following:

```
[AUDITDUMP {ON | OFF}]
[FILESIZE integer ]
[ACTIVEVOL[S] {volume
               {(volume [,volume] ...) } } ]
[FILESPERVOLUME integer]
[OVERFLOWVOL[S] {volume
                 {(volume [,volume]...) } } ]
[OVERFLOWTHRESHOLD integer]
[RESTOREVOL[S] {volume
                {(volume [,volume]...) } } ]
[BEGINTRANSDISABLE integer]
```

AUDITDUMP {ON | OFF}

determines whether audit dumping is performed for the audit trail. ON specifies that audit dumping takes place, and OFF prevents audit dumping. The default is ON.

If AUDITDUMP is ON, after you enter the [ADD AUDITTRAIL](#) command, use the [ALTER AUDITDUMP](#) command to modify the parameters relating to audit dumping. Unless you issue an [ALTER AUDITDUMP](#) command, the attributes of the audit dump process default to the following values:

BLOCKSIZE, the tape data block length: 28 (x 1024 bytes)

COPIES, the number of copies to be made in series or in parallel of each dump: 1 parallel

SYSTEM, the node (system) on which tape copies are made: the local node

VERIFYTAPE, the tape-verification selection: OFF

UNLOAD, the tape-unloading option: ON

If you omit the AUDITDUMP parameter, the default ON is used, and the audit trail is dumped to tape using the AUDITDUMP default attributes listed above.

*FILESIZE integer*

specifies the size of the audit-trail files, in megabytes (MB). The minimum value is 1 and the maximum value is either 2047 or the maximum disk size (if less than 2047 megabytes.) The default value is 100, resulting in files of 100 MB (100 x 1,048,576 bytes) each.

*ACTIVEVOL[S]*

specifies the disk volumes that hold the audit-trail files. At least one volume must be specified for this purpose, either in a SET AUDITTRAIL command or a subsequent [ADD AUDITTRAIL](#) command. Up to 16 volumes can be specified as active-audit volumes.

A single volume cannot be configured as an active volume for more than one audit trail. A volume configured as an active-audit volume cannot be configured as a data volume. To add more volumes to the configuration later, you can use the [ALTER AUDITTRAIL](#) command.

Volumes of the same size should be used for a specific audit trail. TMF allocates the same number of files on each volume, and all volumes must be able to hold all of those files. If a mixture of large and small disks is used for an audit trail, some portion of the large disks will be underused because the active-audit trail will not use them and they cannot be used for audited files; such large volumes, however, could be configured as overflow space or audit restore space.

*FILESPERVOLUME integer*

specifies the number of files in the audit trail that reside on each active-audit volume configured. The audit trail's capacity is determined by the number of audit-trail files per volume, the number of active-audit volumes configured for the trail, and the size of those files.

TMF preallocates the configured number of files when a subsequent [ADD AUDITTRAIL](#) command is executed. If TMF cannot allocate all the files, the [ADD AUDITTRAIL](#) command fails.

The minimum value for FILESPERVOLUME is 1 and the maximum value is 100. The default is 5.

The FILESPERVOLUME value can be altered later, even while TMF is running, by using the [ALTER AUDITTRAIL](#) command.

*OVERFLOWVOL[S]*

specifies the volumes to be used as overflow space for the audit trail, if required (this attribute is not required by TMF, but if you omit it, TMF issues a warning event message at TMF startup, indicating that an overflow-audit volume should be configured). You can specify from 1 to 16 overflow volumes. These can be the same volumes as the active-audit volumes; they can also be data volumes or any volumes not configured for any other TMF purposes. To change the overflow-audit volumes later, you can use the [ALTER AUDITTRAIL](#) command.

For the START TMF command to execute successfully, all overflow-audit volumes must be up.

OVERFLOWTHRESHOLD *integer*

specifies the percentage of audit trail capacity in use above which TMF begins overflow copying. This is an optional parameter. The minimum value is 50 and the maximum is 100. The default is 80(%).

RESTOREVOL[S]

specifies the volumes to be used when restoring audit dumps during execution of a [RECOVER FILES](#) command. If audit dumping is configured for the audit trail, you must specify at least one volume for this purpose; you can specify up to 16 volumes. The volumes specified can be the same as the active-audit volumes; they can also be data volumes or volumes not configured for any other TMF purposes. You can change the restore locations later by entering the [ALTER AUDITTRAIL](#) command.

BEGINTRANSDISABLE *integer*

specifies the percentage of audit trail capacity that can be consumed before TMF disables new transactions. When the consumed capacity meets or exceeds this percentage, new transactions are not allowed; when the consumed capacity falls below this threshold, new transactions are allowed. The minimum value is 50, and the maximum is 100. The default is 90(%).

## Security Restrictions

Anyone can issue the SET AUDITTRAIL command.

## TMF State Requirement

You can enter the SET AUDITTRAIL command at any time, whether or not TMF has been started.

## Usage Guidelines

The values specified in SET AUDITTRAIL are local to your current session only and are not propagated to TMFSERVE. These values take effect when you issue subsequent [ADD AUDITTRAIL](#) commands within your session and cease to apply when your session ends.

You might want to use SET AUDITTRAIL when you expect to issue several subsequent [ADD AUDITTRAIL](#) commands, using the [SET AUDITTRAIL](#) values for certain attributes in most instances and overriding the default values. In this way, the [SET AUDITTRAIL](#) values are carried forward to all the [ADD AUDITTRAIL](#) commands in your session, so that you need to enter specific alternative values only as you want.

You can display the attribute values established by the most recent SET AUDITTRAIL command in your session by entering the [SHOW AUDITTRAIL](#) command.

For additional guidelines, see the [ADD AUDITTRAIL](#) command description earlier in this section.

## Examples

The following SET AUDITTRAIL command sets the BEGINTRANSDISABLE attribute to 85%:

```
TMF 11> SET AUDITTRAIL BEGINTRANSDISABLE 85
```

All subsequent ADD AUDITTRAIL commands that do not include the BEGINTRANSDISABLE parameter use this value for that attribute (rather than the default, 90).

```
TMF 12> ADD AUDITTRAIL MAT, FILESPERVOLUME 6, &
>>>ACTIVEVOLS ($MAT1, $MAT2), &
>>>OVERFLOWVOLS ($DATA1, $DATA2), &
>>>RESTOREVOLS $DATA1
TMF 13>ADD AUDITTRAIL AUX01, FILESPERVOLUME 4, &
>>>ACTIVEVOLS $AUX011, &
>>>OVERFLOWVOLS $DATA3, &
>>>RESTOREVOLS $DATA3
TMF 14> .
.
.
```

The next SET AUDITTRAIL command sets the FILESPERVOLUME attribute to 10, and the AUDITDUMP attribute back to ON, for subsequent ADD AUDITTRAIL commands in this session:

```
TMF 14> SET AUDITTRAIL FILESPERVOLUME 10, AUDITDUMP ON
TMF 15> .
.
.
```

# SET DATAVOLS

The SET DATAVOLS command establishes attribute values for data volumes added by subsequent [ADD DATAVOLS](#) commands issued during the same TMFCOM session.

```
SET DATAVOL[S] attribute [,attribute]...
```

where *attribute* is any of the following:

```
AUDITTRAIL {MASTER | MAT }
            {AUX[ILINARY]nn}
```

```
RECOVERYMODE {ONLINE | ARCHIVE}
```

```
AUDITTRAIL {MASTER | MAT }
            {AUX[ILINARY]nn}
```

identifies the audit trail to which newly added data volumes would send their audit records. MASTER or MAT specifies the master audit trail. AUXILIARY<sub>nn</sub> or AUX<sub>nn</sub> specifies an auxiliary audit trail. *nn* is a two-digit number ranging from 01 to 15 that specifies one of the 15 potential auxiliary audit trails in the configuration. The default is MASTER.

The specified audit trail must exist in the configuration when the [ADD DATAVOLS](#) command is executed.

```
RECOVERYMODE {ONLINE | ARCHIVE}
```

determines whether or not audit-trail files associated with the data volume are held on disk until they are no longer required for volume recovery.

ONLINE retains sufficient audit-trail files on disk so that volume recovery does not need to restore audit records from an archived audit dump. Retaining audit-trail files on disk, however, can affect audit trail management and transaction processing. As an example, if this retention causes an audit trail to exceed a capacity threshold, transaction processing is disabled.

ARCHIVE releases audit-trail files so that they are not retained on disk solely for volume recovery; if volume recovery becomes necessary, one or more audit dumps might need to be restored. You can only select ARCHIVE mode if audit dumping is configured for the associated audit trail.

The default is ONLINE.

## Security Restrictions

Anyone can issue the SET DATAVOLS command.

## TMF State Requirement

You can enter the SET DATAVOLS command at any time, whether or not TMF has been started.

## Usage Guidelines

The values specified in SET DATAVOLS are local to your current session only and are not propagated to TMFSERVE. These values take effect when you issue subsequent [ADD DATAVOLS](#) commands within your session and cease to apply when your session ends.

You might want to use SET DATAVOLS when you expect to issue several subsequent [ADD DATAVOLS](#) commands, using the SET DATAVOLS values for certain attributes in most instances and overriding the default values. In this way, the SET DATAVOLS values are carried forward to all the [ADD DATAVOLS](#) commands in your session, so that you need to enter specific alternative values only as you want.

You can display the attribute values established by the most recent SET DATAVOLS command in your session by entering the [SHOW DATAVOLS](#) command.

For additional guidelines, see the [ADD DATAVOLS](#) command description earlier in this section.

## Examples

The following SET DATAVOLS command sets the RECOVERYMODE attribute to ARCHIVE:

```
TMF 16> SET DATAVOLS RECOVERYMODE ARCHIVE
```

All subsequent ADD DATAVOLS commands that do not include the RECOVERYMODE parameter will use this value for that attribute (rather than the default, ONLINE):

```
TMF 17> ADD DATAVOLS ($DATA1, $DATA2), AUDITTRAIL MAT
TMF 18> ADD DATAVOLS $DATA3, AUDITTRAIL AUX01
TMF 19> .
      .
      .
```

The next SET DATAVOLS command sets the AUDITTRAIL attribute to AUX02, and the RECOVERYMODE attribute back to ONLINE, for subsequent ADD DATAVOLS commands in this session:

```
TMF 19> SET DATAVOLS AUDITTRAIL AUX02, RECOVERYMODE ONLINE
TMF 20> .
      .
      .
```

# SET DUMPS

The SET DUMPS command establishes specifications for online and audit dump entries added to the catalog by subsequent [ADD DUMPS](#) commands issued during the same TMFCOM session.

```
SET DUMPS attribute [,attribute]...
```

where *attribute* is:

```
[DISKMEDIA disk ]
[DISKMEDIA (disk [,disk]...)]

[TAPEMEDIA tape-volume:part:copy]

[SERIAL serial-number]

[TYPE AUDITDUMP ]
[TYPE ONLINEDUMP (master, data)]
```

DISKMEDIA *disk*

for dumps to disk, identifies the disk volume on which the dumps are recorded.

- For an audit dump to disk, only one disk is allowed when specifying *disks*, and the format for *disk* is:

```
[node.]volume.{xx}serial-number.{zz}sequence-number
```

*node* is the node where the dump is located.

*volume* is the volume on which the dump is located.

{*xx*}*serial-number* identifies the dump. *xx* is a two-character prefix, and *serial-number* is the six-digit serial number of the dump. When *node* specifies the node at which your TMFSERVE process is running, *xx* can be any two alphabetic characters. When *node* is different than the one at which your TMFSERVE process is running, *xx* can be any two characters except ZT. If *node* is not specified, ZT is assigned as *xx* by default. Examples are:

AM001017 (for a node local to your TMFSERVE process)

ZT001018 (for a node local to your TMFSERVE process)

CA002001 (for a node remote to your TMFSERVE process)

{*zz*}*sequence-number* identifies the audit trail sequence. *zz* is the audit dump prefix (AA for the master audit trail, BB for the first auxiliary audit trail, and so forth). *sequence-number* is the six-digit current sequence number of



the audit trail when dumped. For example:

AA000036

- For an online dump to disk, any number of disks can be specified by *disk* without limit, and the format for *disk* is:

*[node.]volume.{xx}serial-number.Ffile-number*

*node* is the node where the dump is located.

*volume* is the volume on which the dump is located.

*{xx}serial-number* identifies the dump. *xx* is a two-character prefix, and *serial-number* is the six-digit serial number of the dump. When *node* specifies the node at which your TMFSERVE process is running, *xx* can be any two alphabetic characters. When *node* is different than the node at which your TMFSERVE process is running, *xx* can be any two alphabetic characters except ZT. If *node* is not specified, ZT is assigned as *xx* by default.

Examples are:

DM000010 (for a node local to your TMFSERVE process)

ZT001021 (for a node local to your TMFSERVE process)

NW001022 (for a node remote to your TMFSERVE process)

*file-number* is the number of the online dump file that indicates its sequence in the fileset. There are multiple files for each online dump. The *file-number* is always preceded by the prefix F. Thus, the first file recorded is identified as F0000001, the second file is identified as F0000002, and so forth.

For both online and audit dumps, when the specified disk resides on a local node, specifying the node name is optional. If the node is not specified, the default is the node on which your TMFSERVE process is running.

When the specified disk resides on a remote node, *disk* must be fully qualified, and *volume* must not exceed seven characters, counting the leading dollar sign (\$).

Use the [INFO DUMPS](#) command with the DETAIL option to get a report containing the dump name and the exact time of the dump.

If you specify DISKMEDIA, you must not specify TAPEMEDIA; if you specify TAPEMEDIA, you must not specify DISKMEDIA.

TAPEMEDIA *tape-volume:part:copy*

for dumps to tape, identifies the tape volume on which the dump is recorded. Include this option if the catalog does not already contain an entry for the dump with the specified serial number.

*tape-volume*

is the name of the tape volume.

*part*

specifies which part of the dump is stored on this tape volume. For example, if each copy of the dump requires three tape volumes, *part* is 1, 2, or 3. Because *part* must be specified, it does not have a default value.

*copy*

specifies the copy number of the dump stored on this tape volume. The maximum value for *copy* is 3. For serial copies of dumps being added to the catalog, *copy* should be 1. Because *copy* must be specified, it does not have a default value.

SERIAL *serial-number*

specifies the serial number of the dump to be added to the catalog. You must enter *serial-number* in this parameter. *serial-number* should be the serial number of the original dump; if you enter a number less than 64, TMF returns an error message. Use the INFO DUMPS, DETAIL command to obtain a report containing the dump name and the exact time of the dump.

TYPE [AUDITDUMP  
[ONLINEDUMP (*master*, *data*)]

specifies the type of dump (audit or online) being named to the catalog.

AUDITDUMP indicates that this is an audit dump. You must include this option if you are adding an audit dump entry to the catalog.

ONLINEDUMP indicates that this is an online dump. You must include this option if you are adding an online dump entry to the catalog, specifying the following items:

*master*

is an integer specifying the master audit trail sequence number.

*data*

is an integer specifying the auxiliary audit trail sequence number.

You can obtain both master and auxiliary audit trail sequence numbers by examining output from a previous INFO DUMPS, DETAIL command or a previous DUMP FILES command.

## Security Restrictions

Anyone can issue the SET DUMPS command.

## TMF State Requirement

You can enter the SET DUMPS command at any time, whether or not TMF has been started.

## Usage Guidelines

The values specified in SET DUMPS are local to your current session only and are not propagated to TMFSERVE. These values take effect when you issue subsequent [ADD DUMPS](#) commands within your session and cease to apply when your session ends.

You might want to use SET DUMPS when you expect to issue several subsequent [ADD DUMPS](#) commands, using the SET DUMPS values for certain specifications in most instances and overriding the default values. In this way, the SET DUMPS values are carried forward to all the [ADD DUMPS](#) commands in your session, so that you enter specific alternative values only as you want.

Use the [INFO DUMPS](#) command with the DETAIL option to get the names of online dumps; the DETAIL option is not needed to obtain the names of audit dumps.

Specifying the same value for *attribute* more than once in a SET DUMPS command results in an error message.

You can display the specification values established by the most recent SET DUMPS command in your session by entering the [SHOW DUMPS](#) command.

For additional guidelines, see the [ADD DUMPS](#) command description earlier in this section.

## Examples

The following SET DUMPS command sets the TYPE attribute to AUDITDUMP:

```
TMF 25> SET DUMPS TYPE AUDITDUMP
```

All subsequent ADD DUMPS commands that do not include the TYPE parameter will use this value for that attribute.

```
TMF 26> ADD DUMPS $ZTMFAT.ZTMFAT.AA000024, SERIAL 1066, &
>>>DISKMEDIA $DUMPER.ZT001066.AA000024,          &
>>>TIME JAN 12 2002, 20:20:00
```

```
TMF 27> .
.
.
```

The next SET DUMPS command sets the DISKMEDIA option for an online dump named \$DUMP.ZT001000.F0000001:

```
TMF 27> SET DUMPS DISKMEDIA $DUMP.ZT001000.F0000001
```

```
TMF 28> .
.
.
```

The last SET DUMPS command sets the DISKMEDIA option for two files, \$TMF.ZT001000.F0000001 and \$TMF.ZT001000.F0000002:

```
TMF 28> SET DUMPS DISKMEDIA ($TMF.ZT001000.F0000001, &
>>>$TMF.ZT001000.F0000002)
```

```
~ .
.
.
```

# SET TAPEMEDIA

The SET TAPEMEDIA command establishes attribute values for tape volumes added by subsequent [ADD TAPEMEDIA](#) commands issued during the same TMFCOM session.

```
SET TAPEMEDIA attribute [,attribute] ...
```

where *attribute* is:

```
[DENSITY density]
```

```
[DRIVE [node.]tape-drive]
```

```
[LABEL {ON [!] | OFF}]
```

```
[STATUS {ASSIGNED}
          {BAD}
          {RELEASED}
          {SCRATCH} ]
```

```
[UNLOAD {ON | OFF}]
```

*DENSITY density*

specifies the density at which tapes are written. See [Section 4, TMFCOM Tape Media Considerations](#), for the density settings appropriate for the kind of tape drive you are using.

*DRIVE [*node.*]tape-drive*

identifies the tape drive on which you want to label tape volumes. Guardian labeled-tape processing must be enabled on the node (system) specified in *node*. This parameter is required for labeling tapes.

*node*

is the name of the local or remote node. The syntax rules for specifying a node name appear under [Disk File Names](#), earlier in this section. The default is your local node.

*tape-drive*

is the name of the tape drive.

LABEL ON [!] | OFF

for dumps to tape, determines whether tape volumes are labeled:

ON [!]

ON labels tape volumes on the tape drive specified. If you choose this option, you must also specify the name of the tape drive in the DRIVE parameter. TMF invokes labeled-tape processing for the specified tape volumes to determine whether each one is labeled successfully.

The exclamation point [!] is optional. It specifies that the tape volume is to be labeled whether or not it has already been labeled. If you omit this option, labeled tape processing does not label the volume if it has already been labeled, and does not add a new name to the catalog.

OFF

suppresses tape volume labeling. When recovering the catalog, use this option to add tape volumes back into it.

The default is ON.

---

**Note.** The following SET TAPEMEDIA parameters are only valid when LABEL ON is specified:

- DENSITY
- DRIVE
- UNLOAD

These parameters are not valid (and are ignored) if LABEL ON! (for overwriting existing labels) or LABEL OFF is specified.

---

STATUS

describes the status of the tape volumes named in this command:

ASSIGNED

indicates that the tape volume contains dumps. If the name of the tape volume is found in the catalog, the dump serial number is included in a new entry in the catalog. If the name of the tape volume is not found in the catalog, a new entry is made in the catalog, indicating that the contents of the tape are unknown.

BAD

indicates that the tape volume is defective and cannot be used later for dumping files or for file recovery.

**RELEASED**

indicates that the contents of the tape volume are no longer required for file recovery, but the tape cannot be used for dumping.

**SCRATCH**

indicates that the tape volume can be used for dumping files.

**UNLOAD {ON | OFF}**

determines whether the tape volume is unloaded after it is labeled. ON unloads the tape volume. OFF suppresses unloading. The default is ON.

## Security Restrictions

Anyone can issue the SET TAPEMEDIA command.

## TMF State Requirement

You can enter the SET TAPEMEDIA command at any time, whether or not TMF has been started.

## Usage Guidelines

The values specified in SET TAPEMEDIA are local to your current session only and are not propagated to TMFSERVE. These values take effect when you issue subsequent [ADD TAPEMEDIA](#) commands within your session and cease to apply when your session ends.

You might want to use SET TAPEMEDIA when you expect to issue several subsequent [ADD TAPEMEDIA](#) commands, using the SET TAPEMEDIA values for certain specifications in most instances and overriding the default values. In this way, the SET TAPEMEDIA values are carried forward to all the [ADD TAPEMEDIA](#) commands in your session, so that you need to enter specific alternative values only as you want.

The catalog is not updated if the tape volume is not labeled successfully.

Specifying the same value for *attribute* more than once in a SET TAPEMEDIA command returns an error.

You can display the specification values established by the most recent SET TAPEMEDIA command in your session by entering the [SHOW TAPEMEDIA](#) command.

For additional guidelines, see the [ADD TAPEMEDIA](#) command description earlier in this section.

## Examples

The following SET TAPEMEDIA command sets the UNLOAD specification OFF:

```
TMF 46> SET TAPEMEDIA UNLOAD OFF
```

All subsequent ADD TAPEMEDIA commands that do not include the UNLOAD parameter will use this value for that specification (rather than the default, ON):

```
TMF 47> ADD TAPEMEDIA TT02, DRIVE $TAPE1
TMF 48> ADD TAPEMEDIA TT03, DRIVE $TAPE2
TMF 49> .
      .
      .
```

The next SET TAPEMEDIA command sets the LABEL specification to ON!, and the UNLOAD specification back to ON for subsequent ADD TAPEMEDIA commands in this session:

```
TMF 49> SET TAPEMEDIA LABEL ON!, UNLOAD ON
TMF 50> .
      .
      .
```



# SHOW AUDITTRAIL

The SHOW AUDITTRAIL command displays the attribute values established by [SET AUDITTRAIL](#) commands in the current TMFCOM session. If no [SET AUDITTRAIL](#) command was issued during the session, the SHOW AUDITTRAIL command presents no information.

```
SHOW AUDITTRAIL [attribute [,attribute]...]
                [ *                               ]
```

where *attribute* is any of the following:

```
[AUDITDUMP           ]
[FILESIZE            ]
[ACTIVEVOL[S]        ]
[FILESPERVOLUME      ]
[OVERFLOWVOL[S]      ]
[OVERFLOWTHRESHOLD  ]
[RESTOREVOL[S]       ]
[BEGINTRANSDISABLE  ]
```

## AUDITDUMP

displays the AUDITDUMP setting, indicating whether audit dumping takes place.

## FILESIZE

displays the FILESIZE setting, indicating the size of the audit-trail files, in megabytes.

## ACTIVEVOL[S]

displays the ACTIVEVOL[S] setting, listing the disk volumes that hold the audit-trail files.

## FILESPERVOLUME

displays the FILESPERVOLUME setting, indicating the number of files in the audit trail that reside on each configured active-audit volume.

## OVERFLOWVOL[S]

displays the OVERFLOWVOL[S] setting, listing the disk volumes that will hold overflow information for the audit trail, if generated.

## OVERFLOWTHRESHOLD

displays the OVERFLOWTHRESHOLD setting, indicating the percentage of audit trail capacity that can be used before TMF begins overflow copying.

**RESTOREVOL[S]**

displays the RESTOREVOL[S] setting, listing the disk volumes to be used when restoring audit dumps during execution of a RECOVER FILES command.

**BEGINTRANSDISABLE**

displays the BEGINTRANSDISABLE setting, indicating the percentage of audit trail capacity that can be consumed before TMF disables new transactions.

\*

displays all attribute values established by the [SET AUDITTRAIL](#) command in this session. If you do not specify any option, this is the default.

## Security Restrictions

Anyone can enter the SHOW AUDITTRAIL command.

## TMF State Requirement

You can enter the SHOW AUDITTRAIL command at any time, whether or not TMF has been started.

## Usage Guidelines

If no attribute is specified, the SHOW AUDITTRAIL command displays the values of all attributes set by [SET AUDITTRAIL](#) commands during this session.

## Output Displayed

The SHOW AUDITTRAIL command displays the values of the attributes listed under the syntax description earlier in this discussion. For details about these attributes, see the [ADD AUDITTRAIL](#) or [SET AUDITTRAIL](#) command description.

## Examples

The following SHOW AUDITTRAIL command displays the value of the ACTIVEVOL[S] audit trail attribute, as established by a SET AUDITTRAIL command earlier in this session:

```
TMF 5> SHOW AUDITTRAIL ACTIVEVOLs
      activevol:  $MAT, $MAT2
```

The next SHOW AUDITTRAIL command displays all audit trail attribute values established by SET AUDITTRAIL commands during this session:

```
TMF 6> SHOW AUDITTRAIL  
    activevol:  $MAT, $MAT2  
    filespervolume:  4  
    overflowvols:  $DATA1  
    overflowthreshold:  80
```

# SHOW DATAVOLS

The SHOW DATAVOLS command displays the attribute values established by [SET DATAVOLS](#) commands in the current TMFCOM session. If no [SET DATAVOLS](#) command was issued during the session, the SHOW DATAVOLS command presents no information.

```
SHOW DATAVOL[S] [attribute [,attribute]...]
                  [*
                  ]
```

where *attribute* is:

```
[AUDITTRAIL]
[RECOVERYMODE]
```

## AUDITTRAIL

displays the AUDITTRAIL setting, indicating the audit trail to which the newly added data volumes send their audit records.

## RECOVERYMODE

displays the RECOVERYMODE setting, indicating whether the audit-trail files associated with the volume are held on disk until they are no longer needed for volume recovery.

\*

displays all attribute values established by the [SET DATAVOLS](#) command in this session. If you do not specify any option, this is the default.

## Security Restrictions

Anyone can enter the SHOW DATAVOLS command.

## TMF State Requirement

You can enter the SHOW DATAVOLS command at any time, whether or not TMF has been started.

## Usage Guideline

If no attribute is specified, the SHOW DATAVOLS command displays the values of all attributes set by [SET DATAVOLS](#) commands during this session.

## Output Displayed

The SHOW DATAVOLS command displays the values of two attributes: AUDITTRAIL and RECOVERYMODE. For details about these attributes, see the [ADD DATAVOLS](#) or [SET DATAVOLS](#) command description.

## Examples

The following SHOW DATAVOLS command displays the value of the AUDITTRAIL data volume attribute, as established by a [SET DATAVOLS](#) command earlier in this session:

```
TMF 10> SHOW DATAVOLS AUDITTRAIL  
        audittrail auxiliary01
```

The next SHOW DATAVOLS command displays all data volume attribute values established by [SET DATAVOLS](#) commands during this session:

```
TMF 11> SHOW DATAVOLS  
        audittrail auxiliary01  
        recoverymode online
```

If the [SET DATAVOLS](#) command has not been used in this session, the SHOW DATAVOLS command presents no display:

```
TMF 12> SHOW DATAVOLS  
TMF 13>
```

# SHOW DUMPS

The SHOW DUMPS command displays the attribute values established by [SET DUMPS](#) commands in the current TMFCOM session.

```
SHOW DUMPS [attribute [, attribute]...]
           [ * ]
```

where *attribute* is:

```
[DISKMEDIA]
[TAPEMEDIA]
[SERIAL    ]
[TYPE      ]
```

## DISKMEDIA

for a dump to disk, displays the DISKMEDIA setting, indicating the node, volume, subvolume, and disk on which the dump is located.

## TAPEMEDIA

for a dump to tape, displays the TAPEMEDIA setting, indicating the tape volume on which the dump is stored.

## SERIAL

displays the SERIAL setting, indicating the serial number of the dump added to the catalog.

## TYPE

displays the TYPE setting, specifying the type of dump (audit or online) added to the catalog.

\*

displays all dump specification values established by the [SET DUMPS](#) command in this session. If you do not specify any option, this is the default.

## Security Restrictions

Anyone can enter the SHOW DUMPS command.

## TMF State Requirement

You can enter the SHOW DUMPS command at any time, whether or not TMF has been started.

## Usage Guideline

The SHOW DUMPS command does not display default values for specifications.

If no parameter is specified in SHOW DUMPS, this command displays the values of all specifications set by [SET DUMPS](#) commands during this session.

If a [SET DUMPS](#) command was not issued during the session, the SHOW DUMPS command presents no information.

## Output Displayed

The SHOW DUMPS command displays the values of the specifications listed under the syntax description earlier in this discussion. For details about these specifications, see the [ADD DUMPS](#) or [SET DUMPS](#) command description.

## Examples

The following SHOW DUMPS command displays the specification value established earlier in this session by a [SET DUMPS](#) command:

```
TMF 21> SET DUMPS TYPE AUDITDUMP
```

```
·
·
·
```

```
TMF 22> SHOW DUMPS
      type auditdump
```

The next SHOW DUMPS command displays the DISKMEDIA setting most recently established during this session:

```
TMF 23> SET DUMPS DISKMEDIA $BACKUP.FT001000.F0000001
```

```
·
·
·
```

```
TMF 24> SHOW DUMPS DISKMEDIA
      DISKMEDIA \NORTH1.$BACKUP.FT001000.F0000001
```

If the most recent SET DUMPS DISKMEDIA command included a list of disk volumes, these are displayed when a SHOW DUMPS DISKMEDIA command is entered:

```
TMF 30> SHOW DUMPS DISKMEDIA
      DISKMEDIA (\NORTH2.$TMF.FT001000.F0000001, &
                \NORTH2.$TMF.FT001000.F0000002)
```

If the [SET DUMPS](#) command has not been used during this session, the SHOW DUMPS command presents no information:

```
TMF 10> SHOW DUMPS
TMF 11>
```

# SHOW TAPEMEDIA

The SHOW TAPEMEDIA command displays the attribute values established by [SET TAPEMEDIA](#) commands in the current TMFCOM session.

```
SHOW TAPEMEDIA [attribute [, attribute]...]
                [ * ]
```

where *attribute* is:

```
[ DENSITY ]
[ DRIVE   ]
[ LABEL   ]
[ STATUS  ]
[ UNLOAD  ]
```

## DENSITY

displays the DENSITY setting, indicating the density at which tape volumes are written.

## DRIVE

displays the DRIVE setting, indicating the tape drive on which tape volumes are written.

## LABEL

displays the LABEL setting, indicating whether tape volumes are labeled.

---

**Note.** The following SHOW TAPEMEDIA parameters are only valid when LABEL is specified:

- DENSITY
- DRIVE
- UNLOAD

These parameters are not valid (and are ignored) if LABEL is omitted.

---

## STATUS

displays the STATUS setting, indicating the status of the tape volumes as ASSIGNED, BAD, RELEASED, or SCRATCH.

## UNLOAD

displays the UNLOAD setting, indicating whether the tape volume is unloaded after it is labeled, or remains loaded.



\*

displays all tape media specification values established by the [SET TAPEMEDIA](#) command in this session. If you do not specify any option, this is the default.

## Security Restrictions

Anyone can enter the SHOW TAPEMEDIA command.

## TMF State Requirement

You can enter the SHOW TAPEMEDIA command at any time, whether or not TMF has been started.

## Usage Guidelines

The SHOW TAPEMEDIA command does not display default values for attributes.

If no parameter is specified, SHOW TAPEMEDIA displays the values of all specifications set by [SET TAPEMEDIA](#) commands during this session.

If a [SET TAPEMEDIA](#) command was not issued during the session, the SHOW TAPEMEDIA command presents no information.

## Output Displayed

The SHOW TAPEMEDIA command displays the values of the specifications listed under the syntax description earlier in this discussion. For details about these specifications, see the [ADD TAPEMEDIA](#) or [SET TAPEMEDIA](#) command description.

## Examples

The following SHOW TAPEMEDIA command displays the specification values for tape volumes set by a previous [SET TAPEMEDIA](#) command:

```
TMF 36> SET TAPEMEDIA LABEL ON!, UNLOAD ON
      .
      .
      .
TMF 50> SHOW TAPEMEDIA
      label on !
      unload on
```

If the SET TAPEMEDIA command has not been used in this session, the SHOW TAPEMEDIA command presents no display:

```
TMF 10> SHOW TAPEMEDIA
TMF 11>
```

# START TMF

The START TMF command starts TMF transaction processing, optionally displaying the EMS events for the start operation as this operation progresses. Specifically, START TMF starts audit trail service and data volumes, resolves network transactions, cleans up transactions that were active when TMF last went down, and initiates new transactions.

```
START TMF [ ,NOWAIT ]  
          [ , {ENABLE | DISABLE} BEGINTRANS ]
```

## NOWAIT

requests that when TMFCOM accepts the START TMF command, it suppresses display of the EMS events for the start operation and immediately issues a prompt for another command.

You can determine the status of the start operation, however, by issuing the [STATUS TMF](#) or [DISPLAY OPERATIONS](#) command.

If you do not specify NOWAIT, TMFCOM displays the EMS events on your terminal or writes them to the OUT file (if you specify that) as the start operation progresses.

```
{ENABLE | DISABLE} BEGINTRANS
```

determines whether new transactions are allowed immediately after TMF is started. ENABLE permits new transactions, and DISABLE disallows them. If you omit this parameter, the default is the current value of the BEGINTRANS object as established by the most recent [ENABLE BEGINTRANS](#) or [DISABLE BEGINTRANS](#) command. If neither of these commands has been entered, the default is ENABLE.

## Security Restrictions

You can issue a START TMF command only if you are a member of the super user group.

## TMF State Requirement

You can enter the START TMF command at any time that TMF is not running, provided that audit trails have been configured.

## Usage Guidelines

Audit trails must be configured before TMF will start. After TMF has been started once, auxiliary audit trails cannot be added.

During START TMF operation, volume recovery might not be able to recover some files. The unrecovered files will not be available when the operation completes but they will be listed in EMS messages. To make them available, you might need to use the [RECOVER FILES](#) command to perform a file recovery. (To support file recovery, of course, you must regularly produce online and audit dumps.)

If you issue a [STOP TMF](#) command during a start operation, the command might not take effect immediately; this delay, however, should not be long.

If you do not specify the NOWAIT option, TMFCOM prompts you for a new command when the start operation is complete.

---

△ **Caution.** Verify the status of the stop operation after using a [STOP TMF](#) command with the NOWAIT option, but before entering the START command. Enter the [DISPLAY OPERATIONS](#) command to verify that the shutdown was complete; otherwise, damage to your database can occur during the restart. In the [DISPLAY OPERATIONS](#) command, use the operation number returned by the [STOP TMF](#) command-acceptance message or the appropriate date-time specification.

If you use the START TMF command with the NOWAIT option, also use [DISPLAY OPERATIONS](#) to verify the status of the start operation.

In addition, if multiple users are accessing your TMFSERVE process through multiple TMFCOM processes, be sure to communicate your intentions to stop or restart TMF.

---

If you issue a START TMF command immediately after entering a STOP TMF, ABRUPT command, the start operation might create the new TMP before the old TMF backout process terminates. Very soon thereafter, the backout process abends but is not restarted by the new TMP. In this event, aborting transactions are hung. To recover from this situation, use TMFCOM to enter an ALTER TMF, SWITCHPROCESS TMP command. As a result, the TMP restarts the backout process and aborting transactions are completed.

If TMF crashes while unresolved distributed transactions exist in the prepared state and you subsequently enter a START TMF command, TMF detects those transactions and the start operation cannot proceed until they are resolved. If the parent node is now available, TMF automatically resolves those transactions and the start operation completes. But if the parent node is not available, you must use the [RESOLVE TRANSACTION](#) command to resolve the transactions manually before the start operation can conclude. For more information about the [RESOLVE TRANSACTION](#) command, see [RESOLVE TRANSACTION](#) on page 3-228.

## Output Displayed

As startup progresses, TMFCOM displays EMS messages about various events. The format and content of these messages varies. Many of these messages show the status (starting or started) of the primary processes for individual TMP components; these messages can help in analyzing problems that might occur during TMF startup.

## Examples

The following START TMF command, entered without the NOWAIT parameter, starts TMF and displays the EMS messages for the start operation as it progresses:

TMF 19> **START TMF**

```
START TMF operation [39] begun.
Operation TimeStamp: 24-Jun-2003 20:09:18
20:09 24JUN03 195,02,024 NonStop TMF on \PLUTO *0155* START TMF[1] TMP Process: Begin
Operation.
20:09 24JUN03 195,02,024 NonStop TMF on \PLUTO *0148* START TMF[1] TMP Process: START
TMF parameters: None.
20:09 24JUN03 195,02,024 NonStop TMF on \PLUTO *0120* START TMF[1] TMP Process:
Starting TMF using $SYSTEM as the configuration volume.
20:09 24JUN03 195,02,024 NonStop TMF on \PLUTO *0433* START TMF[1] TMP Process: TMP
Primary Component ProcessMgmt Starting.
.
.
.
20:09 24JUN03 195,02,024 NonStop TMF on \PLUTO *0433* START TMF[1] TMP Process: TMP
Primary Component AuditTrailMgmt Started.
20:09 24JUN03 195,02,024 NonStop TMF on \PLUTO *0433* START TMF[1] TMP Process: TMP
Primary Component MatWriteMgmt Starting.
.
.
.
20:09 24JUN03, 195,02,024 NonStop TMF on \PLUTO *0121* START TMF[1] TMP Process: TMP
Process: TMF Started.
20:09 24JUN03, 195,02,024 NonStop TMF on \PLUTO *0156* START TMF[1] TMP Process: TMP
Process: End operation.
```

TMF 20>

The next START TMF command, which includes the NOWAIT parameter, produces a message that confirms that startup has begun, followed by a prompt for a new command:

TMF 36> **START TMF, NOWAIT**

```
START TMF operation [55] begun.
Operation TimeStamp: 28-Jun-2003 18:22:04
TMF 37>
```

# STATUS AUDITDUMP

The STATUS AUDITDUMP command displays the current status of audit dump processes.

```
STATUS AUDITDUMP [MASTER | MAT
                  [AUX[ILIARY]nn
                  [(MASTER | MAT [,AUX[ILIARY]nn))] ]
                  [(AUX[ILIARY]nn [,MASTER | MAT))] ]
                  [(AUX[ILIARY]nn [,AUX[ILIARY]nn))] ]
                  [ *
                  ]
```

MASTER | MAT  
AUX[ILIARY]nn

identifies the audit trail associated with the audit dump process whose status is to be displayed. MASTER or MAT specifies the master audit trail. AUXILIARY<sub>nn</sub> or AUX<sub>nn</sub> specifies an auxiliary audit trail. *nn* is a two-digit number ranging from 01 through 15 that specifies one of the 15 potential auxiliary audit trails in the configuration. If you do not specify this option, TMFCOM displays the status of all audit dump processes. (When the value of *nn* is 09 or less, you must specify the leading zero in this value.)

\*

displays the status of all audit dump processes.

## Security Restrictions

Anyone can enter the STATUS AUDITDUMP command.

## TMF State Requirement

Before you can enter the STATUS AUDITDUMP command, TMF must be started.

## Usage Guideline

The STATUS AUDITDUMP command displays a subset of the information presented by the [STATUS TMF](#) command.

## Output Displayed

The STATUS AUDITDUMP command displays the primary items listed in [Table 3-18](#).

**Table 3-18. STATUS AUDITDUMP Output**

| Heading or Item | Description   |
|-----------------|---|
| Audit Dump      | Name of the audit trail associated with this dump: MASTER is the master audit trail, and AUXILIARY <sub>nn</sub> is an auxiliary audit trail. |
| State           | Audit dump process state: enabled or disabled.  |
| Status          | Audit dump process status: active or inactive.  |
| Process         | Audit dump process name.  |
| File            | Name of audit-trail file currently being dumped.  |

## Examples

The following STATUS AUDITDUMP command displays the status of the audit dump process associated with the master audit trail:

```
TMF 61> STATUS AUDITDUMP MAT
AuditDump Status:
  Master:  State: enabled, Status: active, Process: \LA.$X457
          File: $MAT1.ZTMFAT.AA000046
```

The next STATUS AUDITDUMP command displays the status of all audit dump processes:

```
TMF 62> STATUS AUDITDUMP *
AuditDump Status:
  Master:  State: enabled, Status: active, Process: \LA.$X457
          File: $MAT1.ZTMFAT.AA000046
AUXILIARY01: The AUXILIARY01 audit trail is not configured
              for dumping
AUXILIARY02: State: DISABLED, Status inactive
AUXILIARY03: State: ENABLED, Status inactive
```

# STATUS AUDITTRAIL

The STATUS AUDITTRAIL command displays the current audit trail activity.

```
STATUS AUDITTRAIL [MASTER | MAT
                  [AUX[ILIARY]nn
                  [(MASTER | MAT [ ,AUX[ILIARY]nn]) ]
                  [(AUX[ILIARY]nn [ ,MASTER | MAT]) ]
                  [(AUX[ILIARY]nn [ ,AUX[ILIARY]nn]) ]
                  [ *
                  ]

[ ,DETAIL]
```

MASTER | MAT  
AUX[ILIARY]nn

identifies the audit trail whose activity is to be displayed. MASTER or MAT specifies the master audit trail. AUXILIARYnn or AUXnn specifies an auxiliary audit trail. nn is a two-digit number ranging from 01 through 15 that specifies one of the 15 potential auxiliary audit trails in the configuration. If you do not specify this option, TMFCOM displays the status of all audit trails. (When the value of nn is 09 or less, you must specify the leading zero in this value.)

\*

displays the status of all audit trails.

DETAIL

displays a list of all audit-trail files currently being managed on disk for the specified audit trail. If DETAIL is not specified, TMFCOM displays only information about the current audit-trail file and the audit trail in general.

## Security Restrictions

Anyone can enter the STATUS AUDITTRAIL command.

## TMF State Requirement

Before you can enter the STATUS AUDITTRAIL command, TMF must be started.

## Usage Guideline

The STATUS AUDITTRAIL command displays the following items:

- The name of the audit trail
- The percentage of audit trail capacity in use
- The name of the oldest audit-trail file that is not yet ready to be reused or purged by TMF
- The reason a file is pinned (retained) on disk
- One or more audit-trail file names, and an indication of their status

These items are explained next under “Output Displayed.”

If the audit trail is out of space and TMF cannot locate a file for audit trail rollover, it will stop generating audit information and display the following message with a STATUS AUDITTRAIL command:

```
WARNING: Out of audit trail space--TMF CANNOT PROCEED. To
continue, issue ALTER AUDITTRAIL command to add an overflow
volume, add an active volume, or increase FILESPERVOLUME,
IMMEDIATELY.
```

To recover from this situation, perform one of the following actions:

- Increase the value of the FILESPERVOLUME attribute:

```
TMFCOM ALTER AUDITTRAIL {MAT | AUXnn},
FILESPERVOLUME integer
```

- Add an active volume to the audit trail configuration:

```
TMFCOM ALTER AUDITTRAIL {MAT | AUXnn},
ADDACTIVEVOL $volume
```

- Add an overflow volume to the audit trail configuration:

```
TMFCOM ALTER AUDITTRAIL {MAT | AUXnn},
ADDOVERFLOW $volume
```

Increasing FILESPERVOLUME allocates new audit-trail files on each of the currently configured active-audit volumes. This is the recommended approach because it distributes the load of the new files across the active volumes. Use this option only if the active volumes have sufficient room to accommodate the new files.



Adding an active-audit volume to the audit trail increases audit trail capacity by the number of files specified in the FILESPERVOLUME attribute. You can use this option only if you have a volume that is not already configured as an active-audit volume or a data volume.

Adding an overflow volume to the audit trail will not immediately resolve the rollover or audit generation problem; the condition will clear, however, once the oldest pinned audit-trail file is copied to the overflow volume.

Regardless of the option you choose to circumvent this problem, it is important that you determine the cause of the problem to make sure it does not recur.

## Output Displayed

The STATUS AUDITTRAIL command displays the items listed in [Table 3-19](#).

---

**Table 3-19. STATUS AUDITTRAIL Output** (page 1 of 4)

| Heading or Item                  | Description   |
|----------------------------------|---|
| Audit Trail Name                 | Name of the audit trail: master is the master audit trail, and auxiliary <sub>nn</sub> is an auxiliary audit trail. This name is followed by the notation "OVERFLOW SPACE IN USE!" if the audit trail has passed its overflow threshold and is using overflow space.                                |
| Active-Audit Trail Capacity Used | Percentage of audit trail capacity in use.  |
| First Pinned File                | Name of the oldest audit-trail file that cannot yet be reused or purged from the active or overflow-audit volume by TMF. This file is said to be pinned (retained) on disk. There is always at least one such audit-trail file on disk, so the display always shows a file name after this heading. |

---

**Table 3-19. STATUS AUDITTRAIL Output** (page 2 of 4)

| Heading or Item | Description  |
|-----------------|--|
| Reason          | <p>The reason the file named under “First Pinned File” is pinned, which is one of the following entries. (In the following list, each entry is followed by an explanation of its meaning.)</p> <p>Active transaction(s)</p> <p>An active transaction might have sent an audit record to this audit-trail file. The audit-trail file remains on the active or overflow-audit volume in case the transaction aborts, in which case the file is needed by the backout process.</p> <p>Current file</p> <p>The audit-trail file is currently receiving audit records.</p> <p>Might be needed to recover datavol <i>volume-name</i> volume</p> <p>This audit-trail file contains audit records needed by the volume recovery process to reapply transactions to the specified data volume. The file is kept on the active or overflow-audit volume so that if volume recovery is needed, the audit file does not have to be restored from archive media.</p> <p>Might be needed to restart TMF</p> <p>This audit-trail file contains control information needed by TMF to restart after a system failure. The control information is normally written to the file within ten minutes.</p> <p>Not dumped</p> <p>An audit dump process has been configured for this audit trail, and this file has not yet been copied to dump media.</p> <p>Not yet determined</p> <p>The TMP has not had enough time to determine why the oldest file is held on disk. (This message is displayed for a brief period after TMF is started and reports a state that lasts only briefly.)</p> <p>Required by RDF</p> <p>RDF requested the pinning of the file on disk, and eventually will release the file.</p> <p>Required by the MAT</p> <p>A master audit-trail file that must remain on the active or overflow volume contains references to audit records in this audit-trail file. This reason applies only to files in an auxiliary audit trail.</p> <p>Volume is down</p> <p>The volume on which the audit-trail file resides is down, so TMF cannot perform the operation needed to release the file.</p> |

**Table 3-19. STATUS AUDITTRAIL Output** (page 3 of 4)

| Heading or Item | Description   |
|-----------------|---|
| Files           | <p>Names of one or more audit-trail files, followed in parentheses by an indication of their status. If the DETAIL option is not specified, only the current audit file is displayed. If the DETAIL option is specified, the display also shows the names of every other file in the audit trail that is being managed on disk by TMF; these include preallocated, overflow, and restored files.</p> <p>The status reported is one of the following entries. (In the following list, each status entry is followed by an explanation of its meaning.)</p> <p>active</p> <p>Active-audit trail files that precede the current file contain audit records needed for transactions in progress or, potentially, for use by volume recovery or a TMF restart. An active-audit trail file that is in sequence after the current file becomes the next current file. This file contains no audit records.</p> <p>available</p> <p>The file is no longer needed for transactions in progress, volume recovery, or TMF restart. If audit dumping is configured, the file has been dumped; it can be reused when the current file gets full.</p> <p>current</p> <p>The file is currently receiving audit records, so it is not ready to be dumped.</p> <p>dumped</p> <p>The file has been copied to archive media by the audit dump process. An available file is not marked as dumped because it must have been dumped to become available.</p> <p>dumping</p> <p>The file is being copied to archive media by the audit dump process, or is the next file to be dumped.</p> <p>not dumped</p> <p>The file has not yet been copied to archive media.</p> <p>not dumping</p> <p>The file will not be dumped because the audit trail was not configured for dumping when the file became full. This file can only be dumped manually, with a backup or duplication program.</p> <p>overflow</p> <p>Overflow-audit files are like active files but have been copied to an overflow volume. This happens when the space consumed by active-audit trail files exceeds the overflow threshold configured for the audit trail.</p> |

**Table 3-19. STATUS AUDITTRAIL Output** (page 4 of 4)

| Heading or Item            | Description  |
|----------------------------|--|
| Files ( <i>continued</i> ) | <p>overflowing</p> <p>The file is the target of an overflow operation, currently in progress. When this operation is done, this file's status changes to OVERFLOW.</p> <p>preallocated</p> <p>Preallocated files are “placeholders” that are used to guarantee that disk space configured for the audit trail is not consumed by other applications. A preallocated file becomes an active file when TMF selects it to be a new audit-trail file.</p> <p>prepared</p> <p>The active-audit trail file has been renamed, and will be the next file to receive audit records.</p> <p>restored</p> <p>The file has been restored to a restore-audit volume for use by volume or file recovery or by an application that reads audit records.</p> <p>restoring</p> <p>The file is being restored to a restore-audit volume.</p> |

## Examples

The following STATUS AUDITTRAIL command, which includes the DETAIL option, presents the detailed status of all configured audit-trail files:

```
TMF 72> STATUS AUDITTRAIL, DETAIL
Audit Trail Status:
Master:
  OVERFLOW SPACE IN USE!
  Active audit trail capacity used: 81%
  First pinned file: $DATA1.ZTMFAT.AA000044
  Reason: Active transaction(s).
Files:
  $SPARE.ZTMFAT.AA000032 (restored)
  $SPARE.ZTMFAT.AA000033 (restoring)
  $DATA1.ZTMFAT.AA000044 (overflow, dumped)
  $MAT2.ZTMFAT.AA000044 (available)
  $DATA1.ZTMFAT.AA000045 (overflowing, dumped)
  $MAT1.ZTMFAT.AA000045 (active, dumped)
  $MAT2.ZTMFAT.AA000046 (active, dumped)
  $MAT1.ZTMFAT.AA000047 (active, dumped)
  $MAT2.ZTMFAT.AA000048 (active, dumping)
  $MAT1.ZTMFAT.AA000049 (active, current)
Auxiliary01:
  Active audit trail capacity used: 54%
  First pinned file: $AUX1.ZTMFAT.BB000003
  Reason: Required by the MAT.
```

## Files:

```
$AUX1.ZTMFAT.BB000001 (available)
$AUX1.ZTMFAT.BB000002 (available)
$AUX1.ZTMFAT.BB000003 (active, dumped)
$AUX1.ZTMFAT.BB000004 (active, dumped)
$AUX1.ZTMFAT.BB000005 (active, dumped)
$AUX1.ZTMFAT.BB000006 (active, current)
$AUX1.ZTMFAT.ZTMF0001 (preallocated)
```

The next STATUS AUDITTRAIL command, issued without the DETAIL option, presents the summarized status that reflects only the current audit-trail files:

TMF 83> **STATUS AUDITTRAIL**

AuditTrail Status:

Master:

```
OVERFLOW SPACE IN USE!
Active audit trail capacity used: 81%
First pinned file: $DATA1.ZTMFAT.AA000044
Reason: Active transaction(s).
Current file: $MAT2.ZTMFAT.AA000049
```

Auxiliary01:

```
Active audit trail capacity used: 54%
First pinned file: $AUX1.ZTMFAT.BB000003
Reason: Required by the MAT.
Current file: $AUX1.ZTMFAT.BB000006
```

# STATUS BEGINTRANS

The STATUS BEGINTRANS command displays the current status of TMF transaction processing, indicating whether this feature is enabled or disabled.

STATUS BEGINTRANS

## Security Restrictions

Anyone can enter the STATUS BEGINTRANS command.

## TMF State Requirement

Before you can enter the STATUS BEGINTRANS command, TMF must be started.

## Usage Guideline

If transaction processing is disabled, STATUS BEGINTRANS reports all reasons why this is so.

## Output Displayed

The STATUS BEGINTRANS command displays transaction processing status, as controlled by the BEGINTRANSACTION system library procedure, which is either enabled or disabled. In addition, if the status is disabled, one or more messages reporting the cause of this status appears in the display.

## Examples

The following STATUS BEGINTRANS command displays the transaction processing status when transaction processing is enabled:

```
TMF 91> STATUS BEGINTRANS
BeginTrans Status: ENABLED
```

The next STATUS BEGINTRANS command displays the transaction processing status when transaction processing is disabled:

```
TMF 98> STATUS BEGINTRANS
BeginTrans Status: DISABLED
  Master Audit trail is above its BEGINTRANSDISABLE
    threshold.
  Auxiliary03 Audit Trail is above its BEGINTRANSDISABLE
    threshold.
The TRANSCOUNTTHRESH has been exceeded.
```

# STATUS CATALOG

The STATUS CATALOG command displays the current activity of the catalog process, indicating the availability of the TMF catalog.

STATUS CATALOG

## Security Restrictions

Anyone can enter the STATUS CATALOG command.

## TMF State Requirement

You can enter the STATUS CATALOG command at any time, whether or not TMF has been started.

## Output Displayed

The STATUS CATALOG command presents one of the following status indications:

- up: the catalog is up and available.
- down: the catalog is down and unavailable.
- active: TMF is working on the catalog, perhaps adding, altering, or deleting entries; the catalog is temporarily busy.

---

△ **Caution.** If the status indication is active, enter a STATUS \$XCAT, DETAIL command through TACL to verify that the catalog process (\$XCAT) is using process time and is in a valid wait state. If the catalog process is not running, a problem exists; you must determine why the process is down and correct the problem.

---

- waiting: TMF is being brought up; the catalog is not yet available.
- stopping: TMF is being shut down; the catalog is not now available.

## Examples

The following STATUS CATALOG command shows that the catalog is up:

```
TMF 15> STATUS CATALOG
Catalog Status:
  Status: up
```

The next STATUS CATALOG command indicates that the catalog is down and unavailable:

```
TMF 32> STATUS CATALOG
Catalog Status:
  Status: down
```

# STATUS DATAVOLS

The STATUS DATAVOLS command displays the current status of data volumes.

```
STATUS DATAVOL[S] [volume]
                   [(volume [,volume]...)]
                   [ * ]

[,STATE {ADDING
          STARTING
          STARTUPBEGUN
          AWAITTRANSRESOLVE
          QUEUEDFORRECOVERY
          RECOVERING
          STARTUPCOMPLETING
          STARTED
          NOTSTARTED
          DOWN
          DOWNCLEAN
          DOWNDIRTY
          SHUTTINGDOWN
          DISABLING
          DISABLED
          DISABLEDCLEAN
          DISABLEDDIRTY
          UNCONFIGURED
          WRONGDISKMOUNTED } ]

[,AUDITTRAIL {MASTER | MAT }
              {AUX[ ILIARY]nn } ]

[,RECOVERYMODE {ONLINE | ARCHIVE} ]

[,SORT {BYATID | BYNAME} ]
```

*volume*

identifies the data volume whose status is to be displayed.

\*

displays the status of all data volumes in the system. If you do not specify a data volume, this is the default.

STATE

displays information about only those data volumes that are in the state indicated by this parameter (STATE serves as a filter). For example, STATE ADDING retrieves information about only those volumes being added to the configuration as data volumes. If you omit the STATE parameter, information is displayed about all configured data volumes, regardless of their states.



**ADDING**

indicates volumes being added to the configuration as data volumes.

**STARTING**

indicates configured volumes that are in one of the startup states (startupbegun, awaittransresolve, queuedforrecovery, recovering, or startupcompleting).

**STARTUPBEGUN**

indicates configured volumes that have just begun automatic startup processing. The associated disk process is processing the initial startup requests for these volumes. Typically, a data volume remains in this state only very briefly.

**AWAITTRANSRESOLVE**

indicates configured data volumes that are being restarted, where the restart is waiting for the resolution of all transactions that may have modified data in the system. TMF must be notified of the outcome (commit or abort) of these transactions before these volumes can take part in recovery. A data volume can linger in this state if lengthy transactions exist in the system or if the outcome of a distributed transaction cannot be determined.

**QUEUEDFORRECOVERY**

indicates configured volumes that are ready for volume recovery but are waiting for process availability. A data volume remains in this state until it is assigned to a volume-recovery process. The data volume might have to wait for an existing volume recovery to complete.

**RECOVERING**

indicates configured volumes that are being operated on by a volume-recovery process.

**STARTUPCOMPLETING**

indicates configured volumes for which startup is nearly complete. The associated disk process is processing the final startup request. Typically, a data volume remains in this state only very briefly.

**STARTED**

indicates configured volumes that are ready to process TMF transactions. Audited requests are allowed only for data volumes in this state as long as transaction processing is enabled within TMF.

**NOTSTARTED**

indicates configured volumes that are not in the started state.

**DOWN**

indicates configured volumes that are in either the downclean or downdirty state.

**DOWNCLEAN**

indicates configured volumes that are inaccessible, perhaps because of device or process problems, and on which no updates have been made to audited files since the volume last underwent a clean shutdown. Audited files on volumes in this state do not contain any inconsistencies.

**DOWNDIRTY**

indicates configured volumes that are inaccessible, perhaps because of device or process problems, but for which normal shutdown processing did not complete successfully. A volume enters the downdirty state if it is down (or goes down) during shutdown, or if data volume startup fails for any reason.

When TMF cannot determine whether a volume is clean, TMF assumes that it is dirty. Audited files on a dirty volume can contain both logical and physical inconsistencies, but these are typically repaired by volume recovery when the volume is restarted.

**SHUTTINGDOWN**

indicates configured volumes that are being shut down by the STOP TMF command.

**DISABLING**

indicates configured volumes being disabled for auditing by the DISABLE DATAVOLS command.

**DISABLED**

indicates configured volumes that are in either the disabledclean or disableddirty state. Disabled data volumes are not automatically restarted each time they come up. Data volumes must be disabled before they can be deleted from the configuration.

**DISABLEDCLEAN**

indicates configured volumes that are disabled, and on which no updates have been made to audited files since the volume last underwent a clean shutdown. Audited files on volumes in this state do not contain any inconsistencies.

**DISABLEDDIRTY**

indicates configured volumes that are disabled, but for which normal shutdown processing did not complete successfully. A volume enters the disabledirty state if it is disabled by a DISABLE DATAVOLS command that includes the ABRUPT option.

When TMF cannot determine whether a volume is clean, TMF assumes that it is dirty. Audited files on a dirty volume can contain both logical and physical inconsistencies, but these are typically repaired by volume recovery when the volume is restarted.

**UNCONFIGURED**

indicates disk volumes that are not configured as TMF data volumes or TMF audit trail volumes.

**WRONGDISKMOUNTED**

indicates volumes for which the physical disk does not match the one configured for TMF (even though the volume name is the same).

AUDITTRAIL {MASTER | MAT }  
              {AUX[ ILIARY] *nn*}

displays information about only those data volumes that send their audit information to the specified audit trail. MASTER or MAT specifies the master audit trail. AUXILIARY*nn* or AUX*nn* specifies an auxiliary audit trail. *nn* is a two-digit number ranging from 01 through 15 that specifies one of the 15 potential auxiliary audit trails in the configuration. (When the value of *nn* is 09 or less, you must specify the leading zero in this value.)

If this attribute is not specified, information is displayed about data volumes that send audit records to any audit trail.

RECOVERYMODE {ONLINE | ARCHIVE}

displays information for only those data volumes that have been assigned the recovery mode (either online or archive) that you specify in this parameter.

SORT {BYATID | BYNAME}

displays the data volume information in the order indicated:

**BYATID**

lists data volumes first by audit-trail ID and then alphabetically by data volume name.

**BYNAME**

lists the data volumes by name, in alphabetical order. If the SORT parameter is omitted, this option is the default.

## Security Restrictions

Anyone can enter the STATUS DATAVOLS command.

## TMF State Requirement

Before you can enter the STATUS DATAVOLS command, TMF must either be in the started state or in the process of shutting down.

## Usage Guidelines

You can control the volumes displayed, using the STATE, AUDITTRAIL, and RECOVERYMODE parameters. These parameters can be used with one another to selectively filter the volumes displayed; for example, you can specify a display of all volumes sending audit to the MAT whose state is notstarted and whose recovery mode is ONLINE:

```
TMF 3> STATUS DATAVOLS, STATE NOTSTARTED, AUDITTRAIL MAT, &  
>>>RECOVERYMODE ONLINE
```

You can control the order in which the information is displayed: either by audit-trail ID or by data volume name.

If TMF is in the process of stopping, STATUS DATAVOLS might return a status of UNKNOWN for some configured data volumes. This status is reported because TMF cannot discern the difference between a volume that has been shut down during execution of a STOP TMF command and one that is not a configured data volume; UNKNOWN indicates that either condition is possible.

If you specify an asterisk (\*) for the volume name list and TMF is stopping, STATUS DATAVOLS only returns the status of configured data volumes that have not already been shut down.

If AWAITTRANSRESOLVE status is displayed for a data volume, the data volume is awaiting restart—but the restart cannot take place until TMF completes all transactions that were active in the system when the volume went down. Thus, transactions that are awaiting resolution can prevent the recovery of a data volume even if those transactions did not access that volume. You can identify transactions awaiting resolution by entering a STATUS TRANSACTIONS, DETAIL command. If you are running applications with long-running transactions and do not wish a data volume to linger in the AWAITTRANSRESOLVE state, you might want to bring those applications down. This action might allow these transactions to finish and volume recovery to complete.

## Output Displayed

The STATUS DATAVOLS command displays the items explained in [Table 3-20](#).

**Table 3-20. STATUS DATAVOLS Output**

| Heading or Item | Description  |
|-----------------|--|
| Volume          | Name of the data volume.                                       |
| Audit Trail     | Audit trail to which this data volume generates audit records. |
| Recovery Mode   | Recovery mode setting (either online or archive).              |
| State           | The current status of the data volume, indicating its state.   |

## Examples

The following STATUS DATAVOLS command, issued without specifying volume names or other options, displays the status of all data volumes in the configuration:

TMF 42> **STATUS DATAVOLS**

| Volume  | Audit Trail | Recovery Mode | State      |
|---------|-------------|---------------|------------|
| -----   | -----       | -----         | -----      |
| \$DATA1 | MAT         | Online        | Started    |
| \$DATA2 | MAT         | Online        | Started    |
| \$DATA3 | MAT         | Online        | Started    |
| \$DATA4 | MAT         | Online        | Recovering |
| \$DATA5 | MAT         | Archive       | Recovering |
| \$DATA6 | AUX01       | Online        | Started    |
| \$DATA7 | AUX01       | Online        | Started    |
| \$DATA8 | AUX01       | Archive       | Recovering |
| \$DATA9 | AUX01       | Online        | Started    |

The next STATUS DATAVOLS command displays all information for all disabled data volumes whose recovery mode is ONLINE:

TMF 56> **STATUS DATAVOLS, STATE DISABLED, RECOVERYMODE ONLINE**

| Volume  | Audit Trail | Recovery Mode | State            |
|---------|-------------|---------------|------------------|
| -----   | -----       | -----         | -----            |
| \$DATA4 | MAT         | Online        | Disabled (Clean) |

The last STATUS DATAVOLS command determines which volumes in the system are not part of the TMF configuration:

TMF 60> **STATUS DATAVOLS, STATE UNCONFIGURED**

| Volume  | Audit Trail | Recovery Mode | State        |
|---------|-------------|---------------|--------------|
| -----   | -----       | -----         | -----        |
| \$DATA0 | -           | -             | Unconfigured |
| \$SPARE | -           | -             | Unconfigured |

# STATUS OPERATIONS

The STATUS OPERATIONS command displays the status of TMF operations that were recently completed or are now in progress.

```
STATUS OPERATION[S]

[operation-number]
[(operation-number [,operation-number]...)]
[*]

[,TYPE {AUDITRESTORE }
      {DELETETMF }
      {DUMPFILES }
      {RECOVERFILES }
      {STARTTMF }
      {STOPTMF }
      {VOLUMERECOVERY}
      {ANY } ]

[,STATE {INPROGRESS}
        {FINISHED }
        {ALL } ]

[,BEGINTIME date-time [GMT]]

[,SORT {BYNUMBER | BYTIME} ]
```

*operation-number*

is an integer that uniquely identifies the operation for which status information is to be displayed. The lowest operation number assigned is 1, and the highest is 100.

\*

displays status information for all operations. If you do not specify an operation, this is the default.

TYPE

displays information only about the type of operation indicated by this parameter. This option applies only when the wild-card character (\*) is specified for *operation-number*. You can specify the following operations:

AUDITRESTORE

indicates restoration of audit dumps from tape or disk.

DELETETMF

indicates purging of the TMF configuration.

## DUMPFILERS

indicates online dumping to tape or disk.

## RECOVERFILES

indicates file recovery.

## STARTTMF

indicates TMF startup.

## STOPTMF

indicates TMF shutdown.

## VOLUMERECOVERY

indicates volume recovery.

## ANY

indicates any operation.

The default is ANY.

## STATE

displays information only about operations that are in the state indicated by this parameter. This option applies only when the wild-card character (\*) is specified for *operation-number*. You can specify the following states:

## INPROGRESS

indicates operations in progress.

## FINISHED

indicates completed operations.

## ALL

indicates all operations, whether in progress or completed.

The default is INPROGRESS.

BEGINTIME *date-time*

specifies the earliest start time of the operations to display. This option applies only when the wild-card character (\*) is specified for *operation-number*. Specify *date-time* in the following format:

{*month*[-]*day*[-]*year*, *hr:min[:sec]*}

{*day*[-]*month*[-]*year*, *hr:min[:sec]*}

where:

*month*

is a three-character abbreviation for the name of the month:

{JAN|FEB|MAR|APR|MAY|JUN|JUL|AUG|SEP|OCT|NOV|DEC}

*day*

is a two-digit integer for the day of the month.

*year*

is a four-digit integer for the calendar year.

---

**Note.** You can use either dashes (-) or spaces, or both, as delimiters to separate the *month*, *day*, and *year* values, but you must use at least one of these delimiters.

---

*hr*

is an integer for the hour, from 0 through 23.

*min*

is an integer for the minute, from 00 through 59.

*sec*

is an integer for the second, from 00 through 59.

GMT

indicates that the BEGINTIME option is specified in Greenwich mean time. If GMT is omitted, the default is the local time at your site.

SORT {BYNUMBER | BYTIME}

displays the operations in the order indicated:

BYNUMBER

lists the operations by operation number. If the SORT parameter is omitted, this option is the default.

BYTIME

lists the operations by ascending BEGINTIME.



## Security Restrictions

Anyone can enter the STATUS OPERATIONS command.

## TMF State Requirement

You can enter the STATUS OPERATIONS command at any time, whether or not TMF has been started.

## Usage Guidelines

When you specify an operation number with the STATUS OPERATIONS command, TMFCOM displays the status of the latest operation assigned that operation number by the current incarnation of the TMP.

TMF recognizes each specific instance of a particular kind of operation (such as TMF start, TMF stop, or file recovery) as a unique operation, and assigns an operation serial number that distinguishes this operation from other recent operations of the same type. If you use the NOWAIT option in the command that generates an operation, TMFCOM displays this number at your terminal. To retrieve information about the operation, enter this number in the *operation-number* parameter of the STATUS OPERATIONS command.

The operation numbers generated are re-used, to keep the number of digits low as an aid in remembering these numbers; after the highest number allowed (100) is assigned, TMF returns to the lowest possible number (1) to identify the next operation.

The TYPE, STATE, and BEGINTIME parameters serve as filters that let you select categories of operations to display. For example, with the TYPE parameter, you can display information for all operations of a certain type. With the BEGINTIME parameter, you can display information for all operations that began at or after a certain time.

If you use the TYPE, STATE, or BEGINTIME parameter, however, you cannot specify an *operation-number*; you must, instead, use the asterisk (\*) wild-card character.

The SORT option allows you to control the order in which operations are displayed: either by operation number or by BEGINTIME.

## Output Displayed

The STATUS OPERATIONS command displays the items explained in [Table 3-21](#).

**Table 3-21. STATUS OPERATIONS Output**

| Heading or Item | Description   |
|-----------------|---|
| Type            | The type of operation: AuditRestore, DeleteTMF, DumpFiles, RecoverFiles, StartTMF, StopTMF, and VolumeRecovery. |
| State           | The state of the operation: InProgress or Finished.   |
| Begin Time      | The date and time the operation began.  |
| End Time        | The date and time the operation completed.  |

## Examples

The following STATUS OPERATIONS command displays information about the operation identified by Operation Number 4, a DumpFiles operation:

```
TMF 5> STATUS OPERATIONS 4
      Type           State           Begin Time           End Time
-----
[4]   DumpFiles      InProgress      21-Jan-2002 13:32:41
```

The next STATUS OPERATIONS command retrieves information about all operations in progress:

```
TMF 6> STATUS OPERATIONS *, STATE INPROGRESS
      Type           State           Begin Time           End Time
-----
[4]   DumpFiles      InProgress      21-Jan-2002 13:32:41
[5]   StartTmf       InProgress      21-Jan-2002 14:57:14
```

The subsequent STATUS OPERATIONS command presents the status of all online dumping operations now in progress:

```
TMF 7> STATUS OPERATIONS *, TYPE DUMPFILES
      Type           State           Begin Time           End Time
-----
[4]   DumpFiles      InProgress      21-Jan-2002 13:32:41
```

The next STATUS OPERATIONS command retrieves the status of all operations of any kind, including those that recently completed:

```
TMF 8> STATUS OPERATIONS *, STATE ALL
      Type           State           Begin Time           End Time
-----
[1]   StartTmf       Finished        21-Jan-2002 08:20:16  21-Jan-2002 08:20:40
[2]   StopTmf        Finished        21-Jan-2002 13:31:43  21-Jan-2002 13:31:47
[3]   StartTmf       Finished        21-Jan-2002 13:32:27  21-Jan-2002 13:32:36
[4]   DumpFiles      Finished        21-Jan-2002 13:32:41  21-Jan-2002 14:57:16
[5]   StartTmf       Finished        21-Jan-2002 14:57:14  21-Jan-2002 14:57:18
```

The last STATUS OPERATIONS command displays information about all operations that began at or after 1:00 p.m., January 21, 2002:

```
TMF 9> STATUS OPERATIONS *, STATE ALL, BEGINTIME 21 JAN 2002, 13:00
      Type              State      Begin Time      End Time
-----
[2]   StopTmf           Finished    21-Jan-2002 13:31:43    21-Jan-2002 13:31:47
[3]   StartTmf          Finished    21-Jan-2002 13:32:27    21-Jan-2002 13:32:36
[4]   DumpFiles         Finished    21-Jan-2002 13:32:41    21-Jan-2002 14:57:16
[5]   StartTmf          Finished    21-Jan-2002 14:57:14    21-Jan-2002 14:57:18
```

# STATUS RESOURCEMANAGER

The STATUS RESOURCEMANAGER command displays the current status of both recoverable and volatile resource managers.

```
STATUS {RESOURCEMANAGER | RM} [rm-name
                                [(rm-name [,rm-name]...)]
                                [*]

    [, STATE {CREATED
              {CLOSED
              {OPEN
              {RECOVERING
              {CRASHED
              {VOLATILE } ]

    [, PROCESS {process
                {(process [,process]...)}
                {(cpu,pin)
                {((cpu,pin) [, (cpu,pin)]...)} ]

    [, DETAIL ]
```

*rm-name*

is the name of the resource manager whose status is to be displayed. This name can contain a maximum of 128 characters; the first character must be a letter, and the others can be alphanumeric or any of the following special characters:

\$ \_ - ^ @ &

Unlike the [ADD RESOURCEMANAGER](#), [CLOSE RESOURCEMANAGER](#), [DELETE RESOURCEMANAGER](#), and [INFO RESOURCEMANAGER](#) commands, the STATUS RESOURCEMANAGER command permits *rm-name* values that begin with the ZZ-VOLATILE- character string. In fact, this string is used when requesting the status of volatile resource managers. This string reflects a name assigned by TMF rather than by a user or gateway process.

Names containing adjoining hyphens (--) are not allowed, although names using multiple hyphens separated by one or more intervening characters are legal. (TMFCOM interprets two consecutive hyphens as the start of a comment.) Thus, the name "ALPHA--1" is not permitted, but "ALPHA-N-1" is allowed.

\*

displays the entries for all recoverable and volatile resource managers currently in the system, selected according to the STATE and PROCESS specifications if they are present. If you do not specify a resource manager name, this action is the default.

## STATE

displays information only for resource managers that are in the state indicated by this parameter. If this parameter is omitted, TMFCOM displays information for all resource managers requested, regardless of state. In this parameter, you can specify one of the following states:

### CREATED

displays information only for resource managers in the created state. A resource manager in this state has been created in the directory, but has not yet been opened by a gateway process.

### CLOSED

displays information only for resource managers that have been closed and that thus have no unresolved transaction branches.

### OPEN

displays information only for resource managers that are currently open by gateway processes and that can handle new transaction branch work.

### RECOVERING

displays information only for resource managers that are currently open by gateway processes but are still retrieving required information about unresolved transactions from the last time these resource managers were closed.

### CRASHED

displays information only for resource managers that are closed but have unresolved transaction branches. These resource managers will need recovery the next time they are opened.

### VOLATILE

displays information only for open volatile resource managers. Transaction branch information cannot be reported for volatile resource managers. However, both volatile and recoverable resource managers count toward the number of open resource managers allowed per processor. Therefore, having many open volatile resource managers reduces the number of possible open recoverable resource managers.

Although volatile resource managers do not have names assigned by users or gateway processes, the STATUS RESOURCEMANAGER command shows a subsystem-generated name in its display. This name starts with the string ZZ-VOLATILE-, followed by the identifying number of the processor where the volatile resource manager is open, and then by an internal identifying number ranging from 0 through 1023 inclusive. The digits in this identifying number are separated by dashes. (Examples appear at the end of this discussion.)

## PROCESS

displays information only for resource managers that are currently open by the gateway processes indicated by *process* or (*cpu, pin*) specifiers:

*process*

is the name of the gateway process that has the resource manager open.

(*cpu, pin*)

is the CPU number and process identification number (PIN) of the gateway process that has the resource manager open.

## DETAIL

specifies that detailed information be displayed about the resource manager, including data about each of the outstanding transaction branches handled for the recoverable resource managers. If you omit this option, transaction-specific information is not displayed.

---

**Note.** Because transaction-branch information is not available for volatile resource managers, TMF ignores the DETAIL option when displaying information about volatile resource managers.

---

If STATUS RESOURCEMANAGER is entered without any parameter list, the display includes information about all resource managers currently defined in the system.

## Security Restrictions

Anyone can enter the STATUS RESOURCEMANAGER command.

## TMF State Requirement

You can enter the STATUS RESOURCEMANAGER command at any time, whether or not TMF has been started. However, if you issue STATUS RESOURCEMANAGER when TMF is stopped, only information about open volatile resource managers is displayed.

## Usage Guidelines

---

**Note.** Resource managers, to which the STATUS RESOURCEMANAGER command applies, support heterogeneous transaction processing. Resource managers are used in products such as NonStop TUXEDO. Although TMFCOM and the TMFSERVE programmatic interface provide TMF commands and options for operating on resource managers, system managers who use them do so in the context of issues involving multiple software subsystems and inter-platform considerations. Complete discussion of heterogeneous transaction processing and the context in which resource-manager commands and options are used lies beyond the scope of the TMF manual library.

For more information about heterogeneous transaction processing, see the Open Group TRANSACTION PROCESSING Publications, available from X/Open Publications at the following Web location:

<http://www.opengroup.org/pubs/catalog/tp.htm>

These publications can also be ordered through many bookstores.

---

The STATUS RESOURCEMANAGER command allows you to select output on the basis of the current state of resource managers, the processes that currently have resource managers open, and the level of detail you wish. A transaction is displayed only if it matches all selection criteria specified in the STATUS RESOURCEMANAGER command. Thus, if you specify both the STATE and PROCESS options, information appears only for resource managers that match both criteria.

If you select the DETAIL option, the display might be very long, depending on the number of transaction branches currently outstanding for the recoverable resource managers reported upon.

If you do not specify a resource manager name in the *rm-name* parameter, or if you select the asterisk (\*) option, information is displayed for all resource managers that match the criteria set by STATE and PROCESS options.

## Output Displayed

For each resource manager reported upon, the STATUS RESOURCEMANAGER command displays the items explained in [Table 3-22](#).

**Table 3-22. STATUS RESOURCEMANAGER Output**

| Heading or Item          | Description   |
|--------------------------|---|
| Name                     | Name of the resource manager.   |
| Status                   | State of the resource manager: created, closed, open, recovering, crashed, or volatile.   |
| Process                  | If the status is open, recovering, or volatile, the name or <i>cpu.pin</i> of the process that currently has the resource manager open.   |
| Transaction Branch State | If the DETAIL option is selected and outstanding transaction branches exist for a recoverable resource manager, the transaction identifier, branch number, and state are displayed for each transaction. Note that the branch number is only unique during the life of a TMF startup operation or an individual gateway process' open of the resource manager; thus, if TMF is shut down or crashes, or if the gateway process fails or closes the resource manager for any reason, the branch numbers for transaction branches might change. |

## Examples

The following STATUS RESOURCEMANAGER command lists the status of all resource managers, without the DETAIL option:

```
TMF 16> STATUS RESOURCEMANAGER *
Resource Manager Status
-----
Name:      TUX-A1_UNIX@SUN
Status:    CLOSED
Process:    None

Name:      TUX-A2_NT@MICROSOFT
Status:    CRASHED
Process:    None

Name:      TUX-HP@FOXII
Status:    OPEN
Process:    $TXGW1

Name:      TUX-HP@ESSG
Status:    RECOVERING
Process:    ( 5, 73 )

Name:      TUX-A3_EXPERIMENTAL-TRANSACTION-PROCESSING-PROTOTYPE-
           PRODUCT
Status:    CREATED
Process:    None

Name:      TUX-HP-TEMP@TSII
Status:    CREATED
Process:    None
```



```
Name:      ZZ-VOLATILE-3-5
Status:    VOLATILE
Process:   (3,75)
```

```
Name:      ZZ-VOLATILE-5-7
Status:    VOLATILE
Process:   $TXVRM
```

The next command displays the status of all resource managers in the crashed state, without the DETAIL option:

```
TMF 28> STATUS RM, STATE CRASHED
Resource Manager Status
-----
Name:      TUX-A2_NT@MICROSOFT
Status:    CRASHED
Process:   None
```

The following command displays the status of all resource managers that are open by the process \$TXGW1, without the DETAIL option:

```
TMF 35> STATUS RM, PROCESS $TXGW1
Resource Manager Status
-----
Name:      TUX-HP@FOXII
Status:    OPEN
Process:   $TXGW1
```

The next command displays the status of all resource managers in the recovering state, and presents the transaction-level detail for each:

```
TMF 40> STATUS RM, STATE RECOVERING, DETAIL
Resource Manager Status
-----
Name:      TUX-HP@ESSG
Status:    RECOVERING
Process:   (5,73)
```

| Transid         | Branch | Number | State     |
|-----------------|--------|--------|-----------|
| \SVLDEV.3.39485 | 1      |        | prepared  |
| \PRUNE.5.2343   | 1      |        | prepared  |
| \TSII.3.93283   | 1      |        | committed |
| \TSII.3.93283   | 2      |        | committed |
| \TSII.3.93283   | 3      |        | prepared  |

In the following example, the STATUS RESOURCEMANAGER command shows all the volatile resource managers that are open:

```
TMF 45> STATUS RM, STATE VOLATILE
Resource Manager Status
-----
Name:      ZZ-VOLATILE-3-5
Status:    VOLATILE
Process:   (3,75)

Name:      ZZ-VOLATILE-5-7
Status:    VOLATILE
Process:   $TXVRM
```

The next command displays the status of all resource managers in the crashed state, and shows the transaction-level detail for each:

TMF 50> **STATUS RM, STATE CRASHED, DETAIL**

```
Resource Manager Status
-----
Name:      TUX-A2_NT@MICROSOFT
Status:    CRASHED
Process:    None

      Transid          Branch  Number          State
-----
\SVLDEV.3.39480          1
\PRUNE.5.2340            1          prepared
                        1          prepared
```

In the next example, the STATUS RESOURCEMANAGER command retrieves all the volatile resource managers that are open, and presents the transaction-level detail. Notice that a special message reflects the fact that transaction-level information is not available for volatile resource managers.

TMF 54> **STATUS RM, STATE VOLATILE, DETAIL**

```
Resource Manager Status
-----
Name:      ZZ-VOLATILE-3-5
Status:    VOLATILE
Process:    (3,75)

Detailed transaction status not available for Volatile
resource managers.

Name:      ZZ-VOLATILE-5-7
Status:    VOLATILE
Process:    $TXVRM

Detailed transaction status not available for Volatile
resource managers.
```

# STATUS SERVER

The STATUS SERVER command displays status and other information about the currently open TMFSERVE process.

STATUS SERVER

## Security Restrictions

Anyone can enter the STATUS SERVER command.

## TMF State Requirement

You can enter the STATUS SERVER command at any time, whether or not TMF has been started.

## Usage Guidelines

To display information about a TMFSERVE process other than the currently open one, issue an [OPEN SERVER](#) command to identify and open that process; then issue a STATUS SERVER command to obtain the information.

## Output Displayed

The STATUS SERVER command displays the items listed in [Table 3-23](#).

---

**Table 3-23. STATUS SERVER Output**

| Heading or Item | Description   |
|-----------------|---|
| System          | Name and ID number of node (system) on which the TMFSERVE process is running.                           |
| Date-Time       | Date and local civil time on the node on which the TMFSERVE process is running.                         |
| Process         | Process name of the TMFSERVE process, along with the processor and PIN at which the process is running. |
| Creator ID      | User ID of the person who created the TMFSERVE process.   |
| Process ID      | User ID of the person now using the TMFSERVE process.   |
| Priority        | Execution priority of the TMFSERVE process.   |
| Object Name     | Name of object program file that contains the code executed by the TMFSERVE process.                    |
| Home Term       | Home terminal for the TMFSERVE process.   |
| Swap Volume     | Swap volume used by the TMFSERVE process.   |
| Version         | Product version identification for TMFSERVE.  |

---

## Examples

The following STATUS SERVER command retrieves information about the currently open TMFSERVE process:

```
TMF 31> STATUS SERVER
TMF Server Status:
System:      \FRESNO (175)
Date-Time:   22-Dec-2003  20:30:15
Process:     $Z193 (8,1120)
Creator ID:  (8,86)
Process ID:  (8,86)
Priority:     146
Object Name: \FRESNO.$SYSTEM.SYS02.TMFSERVE
HomeTerm:    \FRESNO.$TERM36
Swap Volume: $VOL2
Version:     TMFSERVE - T8694G07 - (24NOV2003 - TMF)
```

The display reveals, in the Process entry, that the TMFSERVE is running at PIN 1120 in CPU 8.

The OPEN SERVER command identifies another TMFSERVE process on the remote node \CHICO:

```
TMF 32> OPEN SERVER \CHICO.$SVR04
```

The following STATUS SERVER command retrieves information about this remote TMFSERVE:

```
TMF 33> STATUS SERVER
TMF Server Status:
System:      \CHICO (138)
Date-Time:   22-Dec-2003  20:42:12
Process:     $SVR04 (3,2001)
Creator ID:  (6,22)
Process ID:  (8,86)
Priority:     130
Object Name: \CHICO.$SYSTEM.SYS01.TMFSERVE
HomeTerm:    \CHICO.$TERM20
Swap Volume: $CVOL3
Version:     TMFSERVE - T8694G07 - (24NOV2003 - TMF)
```

# STATUS TMF

The STATUS TMF command displays the current activity of TMF.

```
STATUS TMF
```

## Security Restrictions

Anyone can enter the STATUS TMF command.

## TMF State Requirement

You can enter the STATUS TMF command at any time, whether or not TMF has been started.

## Usage Guidelines

In addition to TMF activity in general, including any process-creation problems encountered during TMF generation, the STATUS TMF command presents status information about the audit dump, audit trail, and catalog processes. Thus, in addition to the general TMF information, the STATUS TMF command combines information from the [STATUS AUDITDUMP](#), [STATUS AUDITTRAIL](#), [STATUS BEGINTRANS](#), and [STATUS CATALOG](#) commands. However, information from the other STATUS commands ([STATUS DATAVOLS](#), [STATUS OPERATIONS](#), [STATUS RESOURCEMANAGER](#), [STATUS SERVER](#), and [STATUS TRANSACTIONS](#)) does not appear in the STATUS TMF display.

If the audit trail is out of space and TMF cannot locate a file for audit trail rollover, it will stop generating audit information and display the following message with a STATUS TMF command:

```
WARNING: Out of audit trail space--TMF CANNOT PROCEED. To
continue, issue ALTER AUDITTRAIL command to add an overflow
volume, add an active volume, or increase FILESPERVOLUME,
IMMEDIATELY.
```

To recover from this situation, perform one of the following actions by entering the ALTER AUDITTRAIL command with the appropriate attribute value:

- Increase the value of the number of files per volume through the FILESPERVOLUME attribute:
- Add an active-audit volume to the audit trail configuration through the ADDACTIVEVOL attribute:
- Add an overflow volume to the audit trail configuration through the ADDOVERFLOW attribute:

Increasing FILESPERVOLUME allocates new audit-trail files on each of the currently configured active-audit volumes. This is the recommended approach because it distributes the load of the new files across the active volumes. Use this option only if the active volumes have sufficient room to accommodate the new files.

Adding an active-audit volume to the audit trail increases audit trail capacity by the number of files specified in the FILESPERVOLUME attribute. You can use this option only if you have a volume that is not already configured as an active-audit volume or a data volume.

Adding an overflow volume to the audit trail will not immediately resolve the rollover or audit generation problem; the condition will clear, however, once the oldest pinned audit-trail file is copied to the overflow volume.

Regardless of the option you choose to solve this problem, it is important that you determine the cause of the problem to make sure it does not recur.

## Output Displayed

The STATUS TMF command displays the TMF items listed in [Table 3-24](#).

**Table 3-24. STATUS TMF Output** (page 1 of 2)

| Heading or Item | Description  |
|-----------------|--|
| System          | Name of the node (system) on which this TMF configuration is running.  |
| State           | <p>Current operating status of TMF, reported as one of the following entries and subentries. (In the following list, each status entry and subentry is followed by an explanation of its meaning.)</p> <p>Empty Audit Trail Configuration</p> <p>TMF has been brought up for the first time on this node and thus no configuration exists for it, or a <a href="#">DELETE TMF</a> command was executed. To start TMF, you must use the <a href="#">ADD AUDITTRAIL</a> command to add at least one audit trail.</p> <p>Configuring New Audit Trails</p> <p>TMF has not yet been started with this configuration. Thus, you can still add new audit trails with the <a href="#">ADD AUDITTRAIL</a> command, alter previously added audit trails with the <a href="#">ALTER AUDITTRAIL</a> command, or start TMF with the <a href="#">START TMF</a> command.</p> <p>Stopped</p> <p>TMF is stopped.</p> <p>Starting</p> <p>TMF is starting and is in one of the following conditions (noted in parentheses in the display):</p> <p>Services</p> <p>TMF is starting audit trail service and other services.</p> <p>Waiting for Network Transactions to be Resolved</p> <p>TMF is waiting for all network transactions to be resolved.</p> <p>Data Volumes</p> <p>TMF is starting data volumes.</p> <p>Running Backout</p> <p>TMF is backing out transactions that must be aborted.</p> <p>Started</p> <p>TMF has started.</p> |

**Table 3-24. STATUS TMF Output** (page 2 of 2)

| Heading or Item            | Description   |
|----------------------------|---|
| State ( <i>continued</i> ) | <p>Stopping</p> <p>TMF is stopping and is in one of the following conditions (noted in parentheses in the display):</p> <p>Waiting for Transactions to Finish</p> <p>TMF is waiting for all transactions to be completed.</p> <p>Data Volumes</p> <p>TMF is stopping data volumes.</p> <p>Waiting for RDF</p> <p>TMF is waiting for the Remote Database Facility (RDF) to shut down.</p> <p>Services</p> <p>TMF is stopping audit trail service and other services.</p> <p>Deleting</p> <p>TMF is purging its current configuration, audit trails, and volume and file recovery information for the database in response to a <a href="#">DELETE TMF</a> command.</p> |
| Transaction Rate           | Current TMF transaction rate, in transactions-per-second.   |

In addition to the information in [Table 3-24](#), hung transactions in the system are also reported.

The other items that appear in the display are explained in the command descriptions for the [STATUS AUDITDUMP](#), [STATUS AUDITTRAIL](#), [STATUS BEGINTRANS](#), and [STATUS CATALOG](#) commands.

## Examples

The following STATUS TMF command displays typical output:

```
TMF 61> STATUS TMF
TMF Status:
System: \FOXII, Time: 10-Feb-2002 09:54:27
State: started
Transaction Rate: 0.10 TPS
AuditTrail Status:
Master
Active audit trail capacity used: 68%
First pinned file: $MAT1.ZTMFAT.AA000044
Reason: Active transactions(s).
Current file: $MAT1.ZTMFAT.AA000045
AuditDump Status:
Master: State: enabled, Status: active, Process \LA.$X545,
File: $MAT2.ZTMFAT.AA000042
BeginTrans Status: ENABLED
```



```
Catalog Status:
  Status: active
    .
    .
    .
```

The output from the next STATUS TMF command illustrates that some of the status might not be available when TMF is in certain states. This example shows the display when TMF is in the stopped state:

```
TMF 64> STATUS TMF
TMF Status:
System: \FOXII, Time: 10-Feb-2002 09:58:14
State: stopped
Transaction Rate: 0.00 TPS
```

# STATUS TRANSACTIONS

The STATUS TRANSACTIONS command displays transaction identifiers and status.

```
STATUS TRANS[ACTION[S]]

[transaction-id
[(transaction-id [,transaction-id]...) ]

[,PROCESS {process
            { (process [,process]... ) }
            { (cpu, pin) }
            { ((cpu, pin) [, (cpu, pin)]...) } ]

[,STATE {ACTIVE
          {ABORTING
          {ABORTED
          {COMMITTED
          {ENDING
          {HUNG
          {PREPARED } ]

[,DETAIL | BRIEF ]
```

*transaction-id*

identifies the transaction for which status information is to be displayed.

*transaction-id* has the following format:

```
[node .
[node-number .
[node(tm-flags) . ]cpu.seq-num
[node-number(tm-flags) . ]
[(tm-flags) . ]
```

where:

*node*

is the name of the home node for the transaction, which is the node (system) on which the transaction originated. The syntax rules for specifying a node name appear under [Disk File Names](#), earlier in this section. The default is the node on which the TMFSERVE process communicating with your TMFCOM process is running.

*node-number*

is the number of the home node for the transaction. A node number specification consists of a backslash followed by the number assigned to the node. The default is the node on which the TMFSERVE process communicating with your TMFCOM process is running.

*tm-flags*

is a number representing flags used internally by TMF. If this number is zero, it does not appear in the STATUS TRANSACTIONS display.

*cpu*

is the CPU number of the processor on which the transaction originated.

*seq-num*

is a sequence number assigned to the transaction, which distinguishes it from other transactions originating in the same processor on the same home node.

## PROCESS

displays only the identifiers for transactions that originate in the process indicated by *process* or (*cpu, pin*):

*process*

is the name of the process that originated (executed the BEGIN-TRANSACTION statement for) the transaction.

(*cpu, pin*)

is the CPU number and process identification number (PIN) of the process that originated the transaction.

## STATE

displays information only for transactions that are in the state indicated by this parameter. If this parameter is omitted, TMFCOM displays information for all transactions, regardless of state. In this parameter, you can specify one of the following states:

### ACTIVE

displays only transactions in the active state. A transaction enters this state when it comes into being. A local transaction comes into being when an application calls the BEGINTRANSACTION procedure. A distributed transaction comes into being when an application accesses a file or server on a remote node on behalf of a locally originated transaction.

### ABORTING

displays only transactions in the aborting state. A transaction enters this state when an application calls the ABORTTRANSACTION procedure or the system aborts the transaction. The transaction remains in this state until all of its effects have been completely undone on this node.

**ABORTED**

displays only transactions that have been aborted. A transaction enters this state when all of its effects are undone and its transaction state (aborted) record is written to the audit trail. A transaction normally stays in this state only briefly. But, in the case of a distributed transaction on a remote node, if the remote node (the child node, in this case) unilaterally aborts a transaction, that transaction remains in the aborted state until the parent node is informed of the outcome. Only transactions in that situation are displayed by the STATUS TRANSACTIONS, STATE ABORTED command.

**COMMITTED**

displays only transactions in the committed state. A transaction enters this state after its transaction state (committed) record is written to the audit trail. A transaction normally stays in this state only briefly. However, in a distributed transaction, a parent node cannot remove a committed transaction until all child nodes have been informed of the commitment. The STATUS TRANSACTIONS, STATE COMMITTED command identifies such transactions.

**ENDING**

displays only transactions in the ending state. In this state, which lasts only very briefly, a transaction is waiting for locks to be released. This is the final state through which a transaction progresses to conclude its life.

**HUNG**

displays only transactions hung during backout. A transaction enters this state if the backout process attempts, but fails, to abort it.

**PREPARED**

displays only distributed transactions in the prepared state. A distributed transaction enters this state after it completes phase 1 of its commitment and its transaction state (prepared) record is written to the audit trail. The transaction remains in this state until the remote node is informed by its parent of the transaction's outcome.

**DETAIL | BRIEF**

specifies the level of information to be displayed: either detailed information for all transactions selected or only their transaction IDs.

If DETAIL is specified, detailed information is given for all selected transactions, including:

- The MAT number, which is the first audit record associated with this transaction
- Whether this transaction is pinning audit-trail files

- Whether the autoabort function is disabled for this transaction
- If the transaction was unilaterally aborted on the remote node, the reason it was aborted
- The age of the transaction

If BRIEF is specified, only the transaction identifiers of the selected transactions are listed. BRIEF is useful when you only need to list the transactions that match your selection criteria.

If neither DETAIL nor BRIEF is chosen, the command output is displayed in a tabular format like that in the first example shown under “Examples.”

If STATUS TRANSACTIONS is entered without any parameter list, the display includes all transactions currently defined in the system (including committed distributed transactions that have not been fully resolved on all nodes concerned).

## Security Restrictions

Anyone can enter the STATUS TRANSACTIONS command.

## TMF State Requirement

Before you can enter the STATUS TRANSACTIONS command, TMF must be started.

## Usage Guidelines

The following usage guidelines apply to the STATUS TRANSACTION command.

### General Guidelines

Transactions can be selected on the basis of transaction identifier, process name, process ID, or transaction state. Both summarized and detailed information can also be displayed for each transaction.

A transaction is displayed only if it matches all selection criteria specified in the STATUS TRANSACTIONS command. Whenever you specify a list for an option, a transaction is displayed if it matches any of the members in that list.

If you do not select the DETAIL option and a transaction has a recoverable resource manager as either a parent or a child, the Parent and/or Child columns in the display will contain the notation “Foreign” for each transaction branch. If a transaction has been exported multiple times by different resource managers or by the same resource manager, the entry “Foreign” will appear for each exported branch in the Child column.

TMF uses the “Foreign” notation in place of the name of the recoverable resource manager because resource manager names can contain up to 128 characters—too many to fit in a non-detailed display. However, you can display the actual names of foreign resource managers in the Parent or Child column by using the DETAIL option in the STATUS TRANSACTIONS command. In this case, the resource manager name

displayed is followed by a period and then the transaction branch number: for example, TUX-HP@ESSG.1. The branch number helps distinguish between two separate exports of the same transaction from the same resource manager.

Notice that the branch number is only unique during the life of a [START TMF](#) command or an individual gateway process' open of the resource manager. Thus, if TMF is shut down or crashes, or if the gateway process fails or closes the resource manager for any reason, the branch numbers for the transaction branches might change.

---

**Note.** Resource managers support heterogeneous transaction processing. They are used in products such as NonStop TUXEDO. Although TMFCOM and the TMFSERVE programmatic interface provide TMF commands and options for operating on resource managers, system managers who use them do so in the context of issues involving multiple software subsystems and inter-platform considerations. Complete discussion of heterogeneous transaction processing and the context in which resource-manager commands and options are used lies beyond the scope of the TMF manual library.

For more information about heterogeneous transaction processing, see the Open Group TRANSACTION PROCESSING Publications, available from X/Open Publications at the following Web location:

<http://www.opengroup.org/pubs/catalog/tp.htm>

These publications can also be ordered through many bookstores.

---

## Transactions in Prolonged Ending State

The STATUS TRANSACTIONS display sometimes shows transactions that remain in the ending state indefinitely. This situation can occur when transactions distributed over the network are in progress and the primary processor for the TMP fails, resulting in orphaned data structures for the transactions. These transactions do not cause problems with TMF operation. But if they exist when a [STOP TMF](#) command is issued, the [STOP TMF](#) command waits indefinitely for TMF to resolve the transactions (which it can never do).

These transactions cannot be affected by [RESOLVE TRANSACTION](#), [DELETE TRANSACTION](#), or [ABORT TRANSACTION](#) commands. The only way to force the TMP to discard the orphaned data structures is to follow one of these procedures:

- If a [STOP TMF](#) operation has not been requested, induce a TMP takeover. To do this, use TMFCOM to issue the ALTER TMF, SWITCHPROCESS TMP command.

---

**Note.** Do not induce a TMP takeover when a STOP TMF operation has already been started. In this case, your ALTER TMF command hangs because the TMP tries to complete the stop operation before processing this command.

---

- If a [STOP TMF](#) operation has already been requested, issue the STOP TMF, ABRUPT command to shut down TMF abruptly. Enter the STOP TMF, ABRUPT command from a TMFCOM process running under a different TACL process.

---

**Note.** Although this approach causes TMF to crash, this should not create a problem because TMF shutdown was already in progress. The only noticeable side effect is that when you next enter a START TMF command, the start operation takes a little longer to execute (a maximum of 10 minutes, but typically far less time).

---

## Output Displayed

The STATUS TRANSACTIONS command displays the items explained in [Table 3-25](#).

---

**Table 3-25. STATUS TRANSACTIONS Output**

---

| Heading or Item        | Description  |
|------------------------|--|
| Transaction Identifier | Transaction identifier that uniquely identifies the transaction; consists of node name, CPU number, and sequence number. In addition, if the <i>tm-flags</i> value used internally by the system is not zero, this value appears in parentheses following the node name. TMF uses one of the flags in <i>tm-flags</i> to track the number of times TMF has been deleted and subsequently re-initiated. (For more information about <i>tm-flags</i> , see the <a href="#">DELETE TMF</a> command discussion.) |
| Process                | CPU number and PIN of process running the transaction.   |
| State                  | State of the transaction: active, aborting, aborted, committed, hung, or prepared.   |
| Parent                 | Parent node for a distributed transaction. If the DETAIL option is not used, the notation “Foreign” indicates a foreign resource manager.  |
| Child                  | Child node for a distributed transaction. If the DETAIL option is not used, the notation “Foreign” indicates a foreign resource manager.   |
| Starting MAT Seq. No   | Audit-trail file sequence number (for example, AA000001) of first audit-trail file associated with this transaction.   |
| Reason                 | If the transaction is aborting or was aborted, brief message text explaining the reason.   |
| # Attempts to Abort    | If the transaction was subject to abort attempts, the number of those attempts. (After 3 attempts, TMF places the transaction in the hung state.)  |
| Start Time             | Age of the transaction on this node.   |

---

## Examples

The following STATUS TRANSACTIONS command lists all transactions on the local node, whose name is \TACO:

TMF 71> **STATUS TRANSACTIONS**

| Transaction Identifier | Process | State    | Parent  | Children |
|------------------------|---------|----------|---------|----------|
| \TACO.3.34567          | (3,234) | active   |         |          |
| \TACO.0.89760          | (0,034) | active   |         | \BURGER  |
|                        |         |          |         | \PIZZA   |
| \PIZZA.1.876           |         | prepared | \PIZZA  | \BURGER  |
| \TACO.2.37876          | (2,345) | hung     |         |          |
| \FILLET.3.456          |         | aborted  | \FILLET |          |

The next STATUS TRANSACTIONS command lists the transaction IDs for all transactions in the hung state:

TMF 82> **STATUS TRANSACTIONS, STATE HUNG, BRIEF**

\TACO.2.37876

The following command, issued on another node, illustrates the display of foreign resource managers when the DETAIL option is not selected:

TMF 90> **STATUS TRANSACTIONS**

| Transaction Identifier | Process | State     | Parent  | Children |
|------------------------|---------|-----------|---------|----------|
| \TSII.4.90273          |         | active    | Foreign |          |
| \SVLDEV.3.39485        |         | prepared  | \SLVDEV |          |
|                        |         |           |         | Foreign  |
| \PRUNE.5.2343          |         | prepared  | \PRUNE  |          |
|                        |         |           |         | Foreign  |
| \TSII.3.93283          | \$APPL3 | committed |         |          |
|                        |         |           |         | Foreign  |
|                        |         |           |         | Foreign  |
|                        |         |           |         | Foreign  |



The next command illustrates the display of foreign resource managers when the **DETAIL** option is selected:

TMF 95> **STATUS TRANSACTIONS, DETAIL**

```
Transaction
Identifier      : \TSII.4.90273
Start Time     : 20-Jan-2002 19:23:57
Process        :
Parent         : TUX-A2_NT@MICROSOFT.1
Child          :
State          : active
# Attempts to  :
Abort          : 0
Starting MAT   :
Seq. No       : 800
```

```
Transaction
Identifier      : \SVLDEV.3.39485
Start Time     : 21-Jan-2002 13:22:15
Process        :
Parent         : \SVLDEV
Child          : \TUX-HP@ESSG.1
State          : prepared
# Attempts to  :
Abort          : 0
Starting MAT   :
Seq. No       : 809
```

```
Identifier      : \PRUNE.5.2343
Start Time     : 21-Jan-2002 13:22:17
Process        :
Parent         : \PRUNE
Child          : TUX-HP@ESSG.1
State          : prepared
# Attempts to  :
Abort          : 0
Starting MAT   :
Seq. No       : 809
```

```
Transaction
Identifier      : \TSII.3.93283
Start Time     : 21-Jan-2002 13:02:09
Process        : $APPL3
Parent         :
Child          : TUX-HP@ESSG.1
Child          : TUX-HP@ESSG.2
Child          : TUX-HP@ESSG.3
State          : committed
# Attempts to  :
Abort          : 0
Starting MAT   :
Seq. No       : 809
```

The following STATUS TRANSACTIONS command presents detailed information about all transactions in the prepared state:

```
TMF 20> STATUS TRANSACTIONS, STATE PREPARED, DETAIL
```

```
Transaction
Identifier      : \PIZZA.1.8768
Start Time     : 24-Jan-2002 14:33:33
Process        : (2,34)
Parent         :
Child          : \BURGER
State          : prepared
# Attempts to  :
Abort          :
Starting MAT   :
Seq. No       : 5480
```

The last STATUS TRANSACTIONS command displays detailed information about all transactions in the aborted state:

```
TMF 24> STATUS TRANSACTIONS, STATE ABORTED, DETAIL
```

```
Transaction
Identifier      : \FILLET.3.456
Start Time     : 24-Jan-2002 14:45:10
Process        : (3,10)
Parent         :
Child          : \FOXII
State          : aborted
Reason         : transaction aborted because of disk process error
# Attempts to  :
Abort          : 1
Starting MAT   :
Seq. No       : 5481
```

# STOP TMF

The STOP TMF command shuts down TMF, terminating transaction processing and displaying the EMS events for the stop operation as it progresses.

```
STOP TMF [ ,NOWAIT]  
[ ,ABRUPT]
```

## NOWAIT

requests that when TMFCOM accepts the STOP TMF command, it suppresses display of the EMS event messages for the stop operation and immediately issues a prompt for another command. You can check the status of the stop operation later by issuing a [STATUS TMF](#) or [DISPLAY OPERATIONS](#) command.

If you do not specify NOWAIT, TMFCOM displays the EMS events on your terminal or writes them to the OUT file (if you specify that) as the stop operation progresses.

## ABRUPT

immediately shuts down TMF by aborting all transactions in progress. When you specify ABRUPT, TMF shuts down without attempting to complete transactions that have not committed or aborted. If you omit ABRUPT, however, TMF shuts down gracefully, remaining in operation until all transactions are committed or backed out, and all audit processing is complete.

## Security Restrictions

You can issue a STOP TMF command only if you are a member of the super user group.

## TMF State Requirement

Before you can enter the STOP TMF command without the ABRUPT option, TMF must be started. However, you can enter the STOP TMF command with the ABRUPT option at any time, whether or not TMF is running.

## Usage Guidelines

When it is accepted by TMFCOM, the STOP TMF command without the ABRUPT option performs the following actions, in the order listed:

- Prevents new transactions from beginning
- Waits for all in-progress transactions to complete
- Stops data volumes
- Terminates audit trail service

To confirm that it accepts the STOP TMF command, TMFCOM displays the following message:

```
STOP TMF operation [operation-number] begun.
```

The ABRUPT option terminates all TMF activities immediately, basically causing TMF to crash. With this option, transactions in the brief, final commit stage are allowed to finish, but all other in-progress transactions are aborted and no new transactions can start. User applications receive an error code indicating whether their transactions were committed or aborted. Aborted transactions are undone by the backout process when TMF is started again.

- 
- △ **Caution.** The STOP, ABRUPT command can disrupt customer applications. If the audit trails are not disturbed while TMF is down, any transactions aborted when the STOP, ABRUPT operation occurs should be properly backed out when TMF is restarted, and the database should remain consistent. Nevertheless, it is always safest to use the ABRUPT option only at times of low transaction activity.
- 

If you issue a STOP TMF command without the ABRUPT option and then decide you do not want to wait for unfinished work to complete, you can press the BREAK key and issue another STOP TMF command with ABRUPT specified. TMFCOM accepts this new command immediately, and the ABRUPT option takes effect at once.

If you issue the STOP TMF command with the NOWAIT option, you can verify later that the shutdown was complete by entering the [DISPLAY OPERATIONS](#) command. In the [DISPLAY OPERATIONS](#) command, use the operation number returned by the STOP TMF command-acceptance message or the appropriate date-time specification.

- 
- △ **Caution.** If multiple users are managing your TMF system, be sure to communicate your intentions to stop or restart TMF.
- 

## Output Displayed

As TMF shutdown progresses, TMFCOM displays EMS messages about various events. The format and content of these messages varies. Many of these messages show the status (stopping or stopped) of the primary and backup processes for the individual TMP components; these messages can help in analyzing problems that might occur during TMF shutdown.

Message examples appear along with the command-entry examples.

## Examples

The following STOP TMF command shuts down TMF, illustrating the confirming message that TMFCOM displays when it accepts this command and the event messages that appear during the shutdown:

```
TMF 90> STOP TMF
STOP TMF operation [40] begun.
Operation TimeStamp: 24-Jun-2003 20:16:04
20:16 24JUN03 195,03,063 NonStop TMF on \PLUTO *0155* STOP TMF [2] TMP Process: Begin
operation.
20:16 24JUN03 195,03,063 NonStop TMF on \PLUTO *0149* STOP TMF [2] TMP Process: STOP
```

```

TMF parameters: None.
20:16 24JUN03 195,03,063 NonStop TMF on \PLUTO *0122* STOP TMF [2] TMP Process:
Stopping TMF.
.
.
20:16 24JUN03 195,03,063 NonStop TMF on \PLUTO *0433* STOP TMF [2] TMP Process: TMP
Primary Component TranscrtlMgmt Stopped.
20:16 24JUN03 195,03,063 NonStop TMF on \PLUTO *0433* STOP TMF [2] TMP Process: TMP
Primary Component ProcessMgmt Stopped.
20:16 24JUN03 195,03,063 NonStop TMF on \PLUTO *0433* STOP TMF [2] TMP Process: TMP
Backup Component ProcessMgmt Stopped.
.
.
20:16 24JUN03 195,03,063 NonStop TMF on \PLUTO *0123* STOP TMF [2] TMP Process: TMP
Process: TMF Stopped.
20:16 24JUN03 195,03,063 NonStop TMF on \PLUTO *0156* STOP TMF [2] TMP Process: TMP
Process: End operation.
TMF 91>

```

The next STOP TMF command illustrates the message that appears when the BREAK key is pressed and the event message display is terminated. The stop operation, however, continues. (To see the suppressed event messages later, you can specify the operation number in a DISPLAY OPERATIONS command.)

```

TMF 95> STOP TMF
STOP TMF operation [52] begun.
Operation TimeStamp: 24-Jun-2003 19:54:06
19:54 24JUN03 198,02,021 NonStop TMF on \PLUTO *0124* STOP TMF [2] TMP Process: Begin
operation.
19:54 24JUN03 198,02,021 NonStop TMF on \PLUTO *0136* STOP TMF [2] TMP Process: STOP
TMF parameters: None.
*** Display of STOP TMF operation [52] terminated by break ***
TMF 96>

```

The next command shows the confirming message displayed by TMFCOM when NOWAIT is specified; EMS events are not displayed:

```

TMF 100> STOP TMF, NOWAIT
STOP TMF operation [56] begun.
Operation TimeStamp: 24-Jun-2003 21:10:42
TMF 101>

```

In this final example, two STOP TMF commands are issued. First, the STOP TMF command is issued without ABRUPT specified. The command begins execution, and the event-message display commences:

```

TMF 121> STOP TMF
STOP TMF operation [76] begun.
Operation TimeStamp: 26-Jun-2003 01:27:09
01:27 26JUN03 197,06,019 NonStop TMF on \PLUTO *0111* STOP TMF [2] TMP Process: Begin
operation.
19:54 26JUN03 197,06,019 NonStop TMF on \PLUTO *0100* STOP TMF [2] TMP Process: STOP
TMF parameters: None
.
.
.

```

Suddenly, however, the user changes her mind and decides to stop TMF without waiting for in-progress transactions to complete. The user presses the BREAK key to interrupt the display:

```

.
.
.

```

```
*** Display of STOP TMF operation [76] terminated by break***  
TMF 122>
```

Now, the user enters a new STOP TMF command, this time specifying ABRUPT. TMF quickly aborts its in-progress transactions and completes the stop operation:

```
TMF 122> STOP TMF, ABRUPT  
TMF 123>
```

# VOLUME

The VOLUME command specifies the default volume and subvolume names for expanding all unqualified file names.

```
VOLUME [ volume ]  
        [ [ volume. ] subvolume ]
```

*volume*

is the name of the default disk volume.

*subvolume*

is the name of the default subvolume.

## Security Restrictions

Anyone can enter the VOLUME command.

## TMF State Requirement

You can enter the VOLUME command at any time, whether or not TMF has been started.

## Usage Guidelines

The VOLUME command allows you to specify default *volume* and *subvolume* names that are recorded for your session and that remain in effect until you issue another VOLUME command or until your session ends. The VOLUME command overrides the default volume in effect when your TMFCOM session begins.

If you do not issue a VOLUME command, no default volume and subvolume are recorded for your session. In this case, unqualified file names are expanded according to the default in effect when you started TMFCOM.

If you enter the VOLUME command without specifying either *volume* or *subvolume*, the default action depends on the parameters you omit:

- If you enter *volume* but omit *subvolume*, the last default subvolume assigned through the VOLUME command remains in effect. If you have not used the VOLUME command to assign a default subvolume in this session, the default subvolume in effect when you started TMFCOM becomes the recorded default subvolume for your session.
- If you omit *volume* but enter *subvolume*, the last default volume assigned with the VOLUME command remains in effect. If you have not used the VOLUME command to assign a default volume in this session, the default volume in effect when you started TMFCOM becomes the recorded default volume for your session.

- If you omit both *volume* and *subvolume*, the recorded default volume and subvolume are cleared. As a result, unqualified file names are expanded according to the defaults in effect when you started TMFCOM.

At any time, you can display the currently recorded default volume and subvolume for your session, as well as the default volume and subvolume in effect at the beginning of your session, by entering the [ENV](#) command.

If you issue a VOLUME command but not a CMDVOL or an OBEYVOL command in your TMFCOM session, the volume you specify in your VOLUME command is used to expand file names in OBEY commands and in the contents of OBEY command files.

If you specify an invalid volume name in *volume*, or an invalid subvolume name in *subvolume*, an error message appears and the VOLUME command is ignored.

The distinction between the VOLUME, CMDVOL, and OBEYVOL commands is as follows:

- The VOLUME command applies to all unqualified file names encountered in all TMFCOM commands during the session, except those used as [CMDVOL](#) and [OBEYVOL](#) command parameters.
- The CMDVOL command applies to all unqualified file names except those used as OBEY command parameters.
- The OBEYVOL command applies only to unqualified file names used as OBEY command parameters.

## Examples

The following VOLUME command specifies that any and all unqualified file names that appear in TMFCOM commands during this session are expanded using the disk volume name \$DATA12:

```
TMF 5> VOLUME $DATA12
```

This assignment is reflected in the output from the following ENV command:

```
TMF 6> ENV
Default Volume      \TSII.$BOOKS1.STAFF
Volume              $DATA12.STAFF
Command Volume      None Specified
Obey Volume         None Specified
In File             \TSII.$ZTN1A.#PTGUJ0U
Out File            \TSII.$ZTN1A.#PTGUJ0U
Server Process      \TSII.$Z35W
```

The next VOLUME command specifies that any and all unqualified file names that appear in TMFCOM commands during this session are expanded using the disk volume name \$DATA16 and the subvolume name TRANREC:

```
TMF 9> VOLUME $DATA16.TRANREC
```



## !

The ! (exclamation-point) command re-executes a command previously issued in the current TMFCOM session, as recorded in the TMFCOM command history buffer.

|  |
|--|
| ! [- <i>linenum</i>   <i>linenum</i>   <i>string</i>   " <i>quoted-string</i> "] |
|--|

*-linenum*

identifies the command to be re-executed relative to the current command line. For instance, !-3 re-executes the third command prior to the current one.

*linenum*

identifies the command to be re-executed by command line number. For instance, !2 re-executes the second command of the current TMFCOM session.

*string*

identifies the command to be re-executed by a text string that specifies the first character or grouping of characters in that command. For instance, !ADD D re-executes the most recent TMFCOM command that begins with “ADD D” and !INFO DUMPS \$ re-executes the most recent INFO DUMPS command that begins with a volume name.

*"quoted-string"*

identifies the command to be re-executed by a text string, enclosed in double (or single) quotes, that specifies a character or grouping of characters located anywhere in the command to be re-executed. TMFCOM searches every character in the history buffer until it finds this string, and then re-executes the command that contains this string. Notice that the string can appear anywhere in a command, not just as the first character or first character grouping. As an example, !“DATAVOL” re-executes the most recent command that refers to a data volume.

## Security Restrictions

Anyone can enter the ! command.

## TMF State Requirement

You can enter the ! command at any time, whether or not TMF has been started.

## Usage Guidelines

Use the ! (exclamation-point) command to display and immediately re-execute a specific command entered previously in the current TMFCOM session. Use the ? (question mark) command, explained following this ! command description, to simply display a command.

If you use the ! command without a line number or text string, TMFCOM re-executes the last command you entered.

You can use the TMFCOM HISTORY command to display the line numbers associated with the most recently entered TMFCOM commands. Only commands currently stored in the command history buffer are displayed, and only those can be re-executed by the ! command.

## Examples

The following example shows how to immediately view and re-execute the command that was issued on line 8 of the current TMFCOM session:

```
TMF 10> !8
TMF 10> STATUS TMF
TMF Status:
  System: \FOXII, Time: 09-Dec-2003 16:11:14
  State: started
  Transaction Rate: 0.00 TPS
AuditTrail Status:
  Master
    Active audit trail capacity used: 30%
    First pinned file: $AUDIT.ZTMFAT.AA000342
    Reason: Current File
    Current file: $AUDIT.ZTMFAT.AA000342
AuditDump Status:
  Master: State: enabled, Status: inactive
BeginTrans Status: ENABLED
Catalog Status:
  Status: up
TMF 11>
```

The next example demonstrates how to re-execute the command that was entered four lines ago:

```
TMF 11> !-4
TMF 11> STATUS CATALOG
Catalog Status:
  Status: up
TMF 12>
```

The following command re-executes the last command that includes the string "VIOLET":

```
TMF 12> !"VIOLET"
TMF 12> INFO DATAVOLS $VIOLET
  Volume      Audit Trail      Recovery Mode
  -----
  $VIOLET      Master              Online
TMF 13>
```

In the following example, the ? command displays the last STATUS command entered in the session, and the ! command re-executes that command:

```
TMF 13> ?STATUS
TMF 13> STATUS CATALOG
TMF 13> !STATUS
TMF 13> STATUS CATALOG
Catalog Status:
    Status: up
```

!

# ?

The ? (question-mark) command displays a command previously entered in the current TMFCOM session, as recorded in the TMFCOM command history buffer.

|  |
|--|
| <code>? [-linenum   linenum   string   "quoted-string"]</code> |
|--|

*-linenum*

identifies the command to be displayed relative to the current command line. For example, ?-3 displays the third command before the current one.

*linenum*

identifies the command to be displayed by command line number. For example, ?2 displays the second command of the current TMFCOM session.

*string*

identifies the command to be displayed by a text string that specifies the first character or grouping of characters in that command. For example, ?ADD D displays the most recent TMFCOM command that begins with ADD D, and ?INFO DUMPS \$ displays the most recent INFO DUMPS command that begins with a volume name.

*"quoted-string"*

identifies the command to be displayed by a text string, enclosed in double (or single) quotes, that specifies a character or grouping of characters located anywhere in that command. TMFCOM searches every character in the history buffer until it finds this string, and then displays the command that contains this string. Notice that the string can appear anywhere in a command, not just as the first character or first character grouping. As an example, ?"DATAVOL" displays the most recent command that refers to a data volume.

## Security Restrictions

Anyone can enter the ? command.

## TMF State Requirement

You can enter the ? command at any time, whether or not TMF has been started.

## Usage Guidelines

Use the ? (question-mark) command to display a specific command entered previously in the current TMFCOM session. Use the ! (exclamation point) command, explained prior to this ? command discussion, to display and re-execute a command.

If you use the ? command without a line number or text string, TMFCOM displays the last command you entered.

You can use the TMFCOM HISTORY command to display the line numbers associated with the most recently entered commands. Only commands currently stored in the command history buffer are displayed, and only those can be selected for action by the ? command.

## Examples

The following example shows how to use the ? command to determine what command was issued as the fifth command in the TMFCOM session:

```
TMF 10> ?5
TMF 10> ADD DATAVOLS $SDATA2
```

The next example demonstrates how to discover what command was entered three lines ago:

```
TMF 11> ?-3
TMF 11> INFO DATAVOLS
```

In the following example, the ? command displays the last STATUS command entered in the session, and the ! command re-executes that command:

```
TMF 12> ?STATUS
TMF 12> STATUS SERVER
TMF 12> !STATUS
TMF 12> STATUS SERVER
TMF Server Status:
  System:                \FOXII      (147)
  Date-Time:             16-Dec-2003  20:06:07
  Node Name              \NEPAL
  Process:               $Y4QL (7,120)
  Creator ID:            (8,21)
  Process ID:            (8,21)
  Priority:              125
  Object Name:           \FOXII.$SYSTEM.SYS00.TMFSERVE
  HomeTerm:              \TSII.$ZTN1A
  Swap Volume:           $SYSTEM
  Version                TMFSERVE - T8694G07 - (24NOV2003 - TMF)
TMF 13>
```



# TMFCOM Tape Media Considerations

TMFCOM commands that affect labeled magnetic tape media (those that add or alter tape volume or dump entries in the TMF catalog) require you to be aware of certain tape-labeling behavior. Information on using labeled tape appears in this section, which discusses these topics:

| Topic  | Page                |
|--|---------------------|
| <a href="#">Tape Density Settings</a>                      | <a href="#">4-1</a> |
| <a href="#">Labeled-Tape Processing Interfaces</a>         | <a href="#">4-2</a> |
| <a href="#">Additional Guidelines for Using Tape Media</a> | <a href="#">4-3</a> |

## Tape Density Settings

Through TMFCOM, you can read and write information to various magnetic tape drives available on NonStop servers.

For open-reel drive options, you can specify the tape density in the DENSITY option of the ADD TAPEMEDIA and SET TAPEMEDIA commands, using the values noted under “Open-Reel Tape Tapes.” The DENSITY option applies only to open-reel drives, and cannot be used with any other kinds of tape drives or devices.

You can reset the DENSITY attribute to its default value using the RESET TAPEMEDIA command and display its current value using the SHOW TAPEMEDIA command.

For detailed labeling and density information for your tape drive, refer to the manual for that particular drive.

## Open-Reel Tape Drives

For all open-reel tape drives, the following rules apply to the ADD TAPEMEDIA and SET TAPEMEDIA commands:

- If you specify the DENSITY option when labeling an open-reel tape, you must set this option to either 1600 or 6250 bits per inch (bpi); the tape volume is then labeled with this value.
- If you omit the DENSITY option, one of the following can occur:
  - If a density value for the tape drive was specified through the system generation program, the tape volume is labeled with that value.
  - If a density value for the tape drive was not specified through the system generation program, the tape volume is labeled at 1600 bpi by default.

## Other Types of Tape Drives and Devices

For labeling tapes using tape drives and devices other than open-reel tape drives, you must omit the DENSITY option from the ADD TAPEMEDIA or SET TAPEMEDIA command. For example, for labeling tapes on cartridge tape drives, digital audio tape (DAT) devices, or digital liner tape (DLT) devices, you would not specify the DENSITY option. If you do include this option for such drives or devices, the command fails and you receive an error message.

## Labeled-Tape Processing Interfaces

TMF tape processing, including the processing of online and audit dumps, requires labeled tapes. If Guardian labeled-tape processing is not enabled for your system, you cannot use labeled tapes and thus cannot perform TMF tape-oriented operations.

In general, as a TMF user, you request tape-labeling functions through TMFCOM commands. For example, the ADD TAPEMEDIA command records tape volume names in the TMF catalog and writes them to tape volume labels; the DUMP FILES command locates the required tape volumes in the catalog and transmits dumps to them. You can, however, communicate directly with a running TMF process to identify tape volumes by using the MEDIACOM operator interface for labeled-tape processing.

When a TMF operation requires a tape and no appropriate TMF scratch tape is presently mounted, TMF sends a message to the operator console requesting that a tape be mounted. The operator can respond by mounting the requested tape or any other tape defined in the TMF catalog as a scratch tape. If you want to specify an alternate to the requested tape, you can use the ALTER TAPEMOUNT command in MEDIACOM. This command issues a new tape-mount request that supersedes the original TMF request. You might want to employ ALTER TAPEMOUNT, for example, to use a copy of an input tape when the tape originally requested is unavailable. Or, when a requested output tape is unavailable, you might want to use the next TMF scratch tape that can satisfy this request. More specifically, you can use the ALTER TAPEMOUNT command to perform these tasks:

- If a requested scratch tape is damaged or unavailable, you can direct TMF to select instead the next scratch tape listed in the TMF catalog.
- If the tape-mount request is for an online or audit dump restore operation, you can direct TMF to select an alternate copy of that dump. You might want to do this, for example, if multiple copies of the dump were made but one copy has been moved off-site or damaged.
- If duplicate copies of tapes were made, you can use ALTER TAPEMOUNT to select the appropriate tape, with the following constraints for open-reel tapes:
  - If the copies were made in parallel in a multi-reel set of tapes, they are interchangeable on a reel-for-reel basis—for example, the first reel in Set A is interchangeable with the first reel in Set B, the second reel in Set A is interchangeable with the second reel in Set B, and so forth, throughout the set.



The ALTER TAPEMOUNT command may be used to choose tapes from either set for input.

- If the copies were made in series, however, they are interchangeable on the first reel only. Thus, if you use the first reel from Set A, the ALTER TAPEMOUNT command cannot be used to choose a reel from Set B. To switch sets, you must reject the current request and start over again, using the correct set throughout.

ALTER TAPEMOUNT can only be issued by members of the super user group.

If the status of a TMF scratch tape changes to BAD while a mount request is pending for that tape, a subsequent MEDIACOM tape-mount request results in an error. To work around this situation, enter a REJECT command to reject the mount request.

For more information about using the MEDIACOM labeled-tape-processing commands, including syntax rules and usage guidelines, see *DSM/Tape Catalog Operator Interface (MEDIACOM) Manual*.

For information about the causes and corrections of common problems that can occur with your tape subsystem, see the discussion of labeled-tape operations in *Guardian User's Guide*.

## Additional Guidelines for Using Tape Media

Additional guidelines for using tape volumes and drives follow.

### Monitoring the Tape Drives

When a defective tape volume is mounted, the tape drive transmits an error message and waits for the operator to mount a new tape or abort the tape request. Even with an autoloader, an operator might have to intervene—the autoloader might not automatically flush a defective tape and make a new tape available to the requesting application. Instead, the autoloader might hang while unattended, and the application would wait indefinitely until someone loaded a correct tape. In all time-critical operations, an operator should monitor the tape drives to ensure that this condition does not prevail.

Always verify that pre-mounted TMF scratch tapes are not write-protected. If the first tape mounted in response to a mount request for a scratch tape is write-protected, TMF still requires that particular tape volume even though it cannot be used. This is particularly troublesome when multiple TMF scratch tapes have been pre-mounted on a multi-tape device, especially a silo device; in such a case, all remaining scratch tapes will be rejected and unloaded. To correct the problem, you must either remove the write protection from the original tape, or reject the request and start over with a tape that is not write-protected.

## Using the VERIFYTAPE Option

In the ALTER AUDITDUMP and DUMP FILES commands, the VERIFYTAPE option determines whether a tape is verified after a dump is written to the tape. When the VERIFYTAPE option is in effect, TMF examines the tape volume for data integrity after the tape is written. This examination involves reading the volume labels, file labels, data records, and checksums.

The improved data accuracy for the newer tape drives and controllers decreases potential risk and the need for verification. In any event, some verification always takes place at the tape-drive process level, regardless of whether VERIFYTAPE is set to ON or OFF. If you feel that you need the extra verification provided by the VERIFYTAPE option, use VERIFYTAPE on a trial basis and assess the impact on system performance. If performance falls below an acceptable level for your purposes, protect your data by using the COPIES 2 option instead of VERIFYTAPE. The COPIES 2 option provides you with an extra copy of the dump tape.

## Using the COPIES Option

In the ALTER AUDITDUMP and DUMP FILES commands, the COPIES option determines the number of copies made of each dump directed to tape. These copies can be produced in series or in parallel. In many cases, you will find that COPIES 1, which produces one copy and which is the default, is sufficient. Although producing multiple copies is sometimes recommended as insurance against a media failure during the dump, this action does involve additional processing time and overhead. Furthermore, the improved technology of the newer tape drives and controllers decreases the potential risk of having just one copy. However, consider all relevant factors at your site. If you decide you need multiple copies, and you are using the newer controllers and drives, the high speed of these units can produce these copies more efficiently if you specify COPIES 2 PARALLEL.

## Choosing the Proper BLOCKSIZE

In the ALTER AUDITDUMP and DUMP FILES commands, the BLOCKSIZE option sets the block length at which data is written to tape. For greatest efficiency, in most cases, HP recommends you set the BLOCKSIZE to 52 (for 52K bytes). However, if your tape drive is on a remote node connected through an EXPAND communication network, you should set the BLOCKSIZE based on the bandwidth available.

## Selecting a Tape Drive Model Type

If you have a choice of tape-drive model types, select a new device (such as a cartridge tape drive, DAT device, or DLT device) rather than an open-reel tape drive. The newer tape drives and devices are more reliable, and their tapes hold more data and are easier to transport.

Along with your tape drive, use autoloaders or cartridge magazines for rapid tape loading and automated cartridge systems to handle very large volumes of tapes.

Autoloaders help reduce the frequency of tape-handling by the operator and, depending on the model, allow you to place six or seven tapes in the hopper at once. The autoloader cycles through these tapes in order, loading them automatically.

Cartridge magazines allow you to transport and mount collective sets of tapes.

A Storage Technology Corporation (STK) Automated Cartridge System (ACS) reduces the need for operator intervention more dramatically. For example, a single STK 4400 ACS holds up to 6000 tapes, and multiple ACSs can be configured on a single system to provide even more capacity. An ACS can automatically mount a tape in less than 20 seconds.

## Considering File Size Factors

Processing fewer large files typically produces a higher data transfer rate than processing many small files. Operations with many small files involve additional overhead per block; the system must open each file individually, ensure that it is indeed the file requested, verify that it is available and that its security constraints do not preclude access to it, and so forth. When a program processes small files, the data transfer rate is lowered. With larger files, however, much greater rates are possible.

The number of files, copied from disk, that a tape volume can hold is highly variable and depends on two factors:

- The capacity of the tape volume.
- The average size of the files.

Some tape drives provide compression, some support 36-track recording, and some offer both of these features. Be certain that the drives you use are compatible with one another; otherwise, tapes written on one drive will not be readable on another. For example, a compressed tape produced on a Model 5190 drive cannot be read on a Model 5180 drive, which does not support compression.

For information about the characteristics and storage capacity of your tape drive, refer to the manual for that particular drive.



# **A** TMFCOM Messages

The messages generated by TMFCOM are described in this appendix. These messages appear on your workstation or terminal screen during a TMFCOM session. Alternatively, they can be directed to another output device or file through the TMFCOM command OUT option described in [Directing Output to an Alternate Destination](#) on page 2-8.

This section discusses these topics:

| <b>Topic</b>   | <b>Page</b>         |
|--|---------------------|
| <a href="#">TMFCOM Message Overview</a>                    | <a href="#">A-1</a> |
| <a href="#">Message Types</a>                              | <a href="#">A-2</a> |
| <a href="#">Message Numbers</a>                            | <a href="#">A-2</a> |
| <a href="#">Error Codes in Messages</a>                    | <a href="#">A-2</a> |
| <a href="#">File Names in Messages</a>                     | <a href="#">A-3</a> |
| <a href="#">Error Identifiers</a>                          | <a href="#">A-3</a> |
| <a href="#">Other Messages</a>                             | <a href="#">A-4</a> |
| <a href="#">Error Information and Failure Data Capture</a> | <a href="#">A-4</a> |
| <a href="#">Message Descriptions</a>                       | <a href="#">A-6</a> |

## **TMFCOM Message Overview**

The individual TMFCOM messages are described under [Message Descriptions](#), beginning on [A-6](#). In each message description, the following information appears:

- Message number
- Message text
- Error codes and parameters—explanations of variables that indicate specific error details, files, devices, and other elements that appear within the message text
- Cause—the condition or error that caused TMFCOM to produce the message
- Effect—the effect of the condition or error on the system
- Recovery—the steps required to recover from a reported error

## Message Types

TMFCOM displays three basic types of messages:

- Informative messages that report status or other normal conditions and that require no action on your part
- Error messages (preceded by the word ERROR) that indicate a mistake in a command that prevented its execution; these messages typically require you to re-enter the command correctly
- Warning messages (preceded by the word WARNING) that indicate a potentially dangerous or invalid operation, but that allow command execution to proceed; these messages sometimes require you to re-enter the command, but often they do not

---

**Note.** Repetitions of the same message number and text do not always belong to the same message type. For example, the same message might be accompanied by the ERROR notation in some circumstances and by the WARNING notation, or no notation at all, in others. For this reason, the message types are not noted in the individual message descriptions.

---

## Message Numbers

TMFCOM messages begin with unique identifying numbers. Under [Message Descriptions](#), the messages are listed in order of message number. The numeric range in which a message number falls indicates the general category of the message:

| Range     | Category  |
|-----------|---|
| 1-200     | Command syntax error                                  |
| 201-500   | Command execution error                               |
| 501-600   | Subsystem Programmatic Interface (SPI) internal error |
| 601-700   | Warning   |
| 900-1000  | TMF error   |
| 1001-3000 | TMFSERVE internal error                               |
| 7000-7999 | Parsing Error   |

---

**Note.** Messages 1 through 600 may be returned through the SPI interface as well as through TMFCOM.

---

## Error Codes in Messages

Some TMFCOM messages include other messages from TMFCOM or information from the operating system's File System, the NEWPROCESS process-control procedure, or the Enform query language and report writer. When present, this information appears in the *errcode* parameter along with the identity of the transmitting subsystem.

## File Names in Messages

Many TMFCOM messages include as parameters file names that identify objects such as disk files, tape volumes, and processes. Often these messages report syntax errors that result from improperly specifying these file names in commands. When present, these names appear in the message descriptions as parameters such as *file-id*, *volume*, and *process-name*, among others. For more information about these file names and the rules that govern them, see the discussion [File Names and Process Identifiers](#) on page 3-6, and the *Guardian Procedure Calls Reference Manual*.

## Error Identifiers

When TMFCOM detects an error in a command, it redisplay the command, places a caret (^) under the first character of the word or other element that contains the error (if this element can be identified), and then presents the error message. For example, in the following command, the caret identifies the keyword "TNF" as erroneous (it should have been entered as "TMF"):

```
TMF 2> STATUS TNF
      STATUS TNF
           ^
**ERROR-7072** Expecting AUDITDUMP, AUDITTRAIL, BEGINTRANS,
                CATALOG, DATAVOL[S], OPERATION[S], SERVER, TMF, or
                TRANS[ACTION[S]].
TMF 3>
```

You can easily correct this error and re-execute the command by using the FC or ! command.

If a command exceeds 240 characters (3 display lines on a typical workstation or terminal screen) and a parsing error (Message 7000 through 7999) is detected, TMFCOM does not redisplay the entire command when reporting the error. Instead, TMFCOM displays only one line of the command to give context and points to the error within that line. Specifically, TMFCOM presents only the 37 characters immediately preceding the error, the error itself indicated by the caret, and up to 40 characters following the error. TMFCOM indicates the omitted command text by showing an ellipsis (three periods) in its place. The display appears as follows:

| Location of Error in Command                             | Command Text Displayed  |
|--|---|
| Within the first 76 characters                           | The first 76 characters of the command, including the error.                                  |
| Within the middle of a command longer than 76 characters | The 37 characters immediately preceding the error and the last 40 characters after the error. |
| Within the last 76 characters                            | The last 76 characters of the command, including the error.                                   |

As an example, for the following excessively long command, TMFCOM would report the error as follows:

```
TMF 3> ALTER BEGINTRANS, TRANSCOUNTTHRESH (1500, 1600), TMFLIBMEMTHRESH (85,90), TMPM
EMTHRESH (90,95), TTRANSPERCPU (1024,1024), AUTOABORT 7200 SECONDS, RECRMOUNT 256,
RMOPENPERCPU 128, BRANCHESPERRM 128 (more characters follow, exceeding 240 total...)
... (85,90), TMPMEMTHRESH (90, 95), TTRANSPERCPU (1024,1024), AUTOABORT 720 ...
^
**ERROR-7172** Expecting AUTOABORT, BRANCHESPERRM, RECRMOUNT, RESET,
RMOPENCPU, TMFLIBMEMTHRESH, TMPMEMTHRESH, TRANSCOUNTTHRESH, or
TRANSPERCPU.
TMF 4>
```

## Other Messages

Other messages are generated as part of the normal output from TMFCOM commands such as INFO and STATUS. These messages are generally self-explanatory and are not documented in this appendix; many examples, however, appear in the individual command descriptions in [Section 3, TMFCOM Commands](#).

Event messages, directed to the Event Management Service (EMS) log file through the \$0 collector process, also appear on your screen. They often appear in displays produced by commands such as DUMP FILES, DISPLAY OPERATIONS, and RECOVER FILES. These are not TMFCOM messages, but are instead produced by other TMF processes; they are documented in the *Operator Messages Manual*, along with the event messages generated by many other subsystems.

## Error Information and Failure Data Capture

If a TMF internal error or a software or hardware failure occurs, capture as much information about the problem as possible. Then, contact the Global Customer Support Center (GCSC) or your service provider and supply all the relevant information.

For all problems, please capture and provide the following standard items:

- Your system serial number
- Product numbers and versions of TMF software and all related products involved in the problem
- Description of the problem and the accompanying symptoms
- Event Management Service (EMS) log files, containing recent event messages written during the last 12 to 24 hours (usually located on \$SYSTEM.ZLOGnn, or if that subvolume does not exist, \$SYSTEM.SYSTEM.OPRLOG)
- SAVEABEND files generated at the time of failure; these will be located on the same volume and subvolume as the program files (typically \$SYSTEM.SYSnn)
- TMF configuration files (including the TMF catalog), whose location is listed in the INFO TMF command display



- If the policies at your site permit:
  - Copies of any audit-trail files you believe might be related to the problem (for example, those for which audit-reading errors are reported)
  - Relevant information about the application running when the problem occurred

For problems involving processor (CPU) halts or hangs, in addition to the standard items, please supply:

- Processor (CPU) dumps
- Copies of the following files from the \$SYSTEM.SYS $nn$  subvolume ( $nn$  is the system image number):
  - CONFLIST
  - OSCONFIG
  - TSC
  - TSL
  - TSYSCLR (for RISC-based systems only)
  - TSYSDP2

For TMFCOM internal errors, in addition to the standard items, please provide:

- The TMFCOM error-message number and text
- Any additional data requested by the error message

# Message Descriptions

The numbered messages are described in the following pages.

## 1

The command specified an invalid TAPEMEDIA name, *tape-volume*. If this name begins with a digit, all characters must be digits. If this name begins with a letter, subsequent characters can be alphanumeric. Also, the command might require the “*part:copy*” specification.

*tape-volume*

is the invalid tape-volume name.

*part*

for a command dealing with a dump, identifies the portion of the dump stored on this tape volume.

*copy*

for a command dealing with a dump, identifies the copy number of the dump stored on this tape volume.

**Cause.** The command specified an invalid tape-volume name or failed to supply the *part:copy* specifier when it was required. The tape-volume name refers to the tape identifier magnetically encoded on the tape. Perhaps the name used in the command was too long or contained improper characters.

**Effect.** The command fails.

**Recovery.** Verify that the tape-volume name complies with the following syntax rules:

- The name must begin with a digit or a letter.
- If the name begins with a digit, up to five more digits can follow.
- If the name begins with a letter, up to five digits or letters in any combination can follow. (Either uppercase or lowercase letters are permitted.)

Re-enter the command with a correct tape-volume name or with the *part:copy* specifier, if required.

## 2

This request contained too many structures; either one structure or none is permitted.

**Cause.** A Subsystem Programmatic Interface (SPI) violation occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 3

The command specified a filename that was not correctly formed. The invalid filename was: *filename*.

*filename*

is the invalid file name.

**Cause.** The command contained an invalid file name. Perhaps the *node-name*, *volume*, *subvolume*, or *file-id* field was too long, contained improper characters, or was delimited by a character other than a period.

**Effect.** The command fails.

**Recovery.** Verify that the file name complies with the syntax rules listed in [File Names and Process Identifiers](#) on page 3-6. Then, re-enter the command with a correct file name.

## 4

In this context, the command cannot RESET a filename.

**Cause.** The RESET parameter requested a file name change, but file names cannot be changed within the context of this command.

**Effect.** The command fails.

**Recovery.** Re-enter the command without requesting a file-name change.

## 5

In this context, the command cannot specify a remote file.  
The remote filename is: *filename*.

*filename*

is the remote file name.

**Cause.** The command contained a parameter that referred to a remote file, but only local file names are allowed for that parameter.

**Effect.** The command fails.

**Recovery.** Re-enter the command, using a local file name.

## 6

The command specified an invalid TYPE parameter. TYPE must be either ONLINEDUMP or AUDITDUMP.

**Cause.** The TYPE parameter specified an invalid dump type. The only entry allowed for this parameter is either ONLINEDUMP or AUDITDUMP.

**Effect.** The command fails.

**Recovery.** Re-enter the command, using ONLINEDUMP to indicate an online dump or AUDITDUMP to indicate an audit dump.

## 7

The command specified an invalid STATUS parameter. STATUS must be ASSIGNED, RELEASED, BAD, SCRATCH, or INVALID.

**Cause.** The STATUS parameter specified an invalid dump status.

**Effect.** The command fails.

**Recovery.** Re-enter the command, using one of the following entries in the STATUS parameter: ASSIGNED, RELEASED, BAD, SCRATCH, or INVALID. For the meaning of these entries, see the ALTER TAPEMEDIA command description in [Section 3, TMFCOM Commands](#).

## 8

The command specified an invalid SORT parameter. SORT must be either BYFILE or BYSERIAL.

**Cause.** In an INFO DUMPS command, the SORT parameter specified an invalid order for listing dumps.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying BYFILE to list the dumps alphabetically by file name or BYSERIAL to list the dumps in descending order by serial number.

## 9

In this context, the command must specify exactly one TransID.

**Cause.** A DELETE TRANSACTION command specified multiple transactions; however, TMF allows you to select only one transaction at a time.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying only one transaction.

## 10

The command specified an invalid TransID (*transaction-id*).

*transaction-id*

is the invalid transaction identifier.

**Cause.** The command specified an invalid or nonexistent transaction in its *transaction-id* parameter list.

**Effect.** The command fails.

**Recovery.** Enter a STATUS TRANSACTIONS \* command to list the valid transactions in the system. Then, re-enter the previous command, using a correct transaction identifier.

## 11

In this context, a TransID cannot contain wild-card characters. The invalid TransID is *transaction-id*.

*transaction-id*

is the invalid transaction identifier.

**Cause.** The command specified one or more wild-card characters (\*) in its *transaction-id* parameter list. However, only fully qualified transaction identifiers can be used in this command.

**Effect.** The command fails.

**Recovery.** Enter a STATUS TRANSACTIONS \* command to list the valid transactions in the system. Then, re-enter the previous command, using only fully qualified transaction identifiers.

## 12

The command specified an invalid STATE parameter. STATE must specify ACTIVE, ABORTED, ABORTING, COMMITTED, ENDING, HUNG, or PREPARED.

**Cause.** The command specified an invalid transaction state in the STATE parameter.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying one of the following entries in the STATE parameter: ACTIVE, ABORTED, ABORTING, COMMITTED, ENDING, HUNG, or PREPARED. For more information about these states, see the STATUS TRANSACTIONS command description in [Section 3, TMFCOM Commands](#).

## 13

An output-only field in the structure for this command was not NULL on input.

**Cause.** A Subsystem Programmatic Interface (SPI) violation occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 14

The command specified an invalid process name.  
The invalid process name was *process-name*.

*process-name*

is the invalid process name.

**Cause.** The command specified an invalid process name. Perhaps this name was too long, contained improper characters, or did not begin with a dollar sign (\$).

**Effect.** The command fails.

**Recovery.** Verify that the process name complies with the syntax rules listed in [File Names and Process Identifiers](#) on page 3-6. Then, re-enter the command with the correct process name.

## 15

In this context, the command cannot specify a remote process name. The invalid process name was *process-name*.

*process-name*

is the remote process name.

**Cause.** A command parameter referred to a remote process, but only local process names are allowed for this parameter.

**Effect.** The command fails.

**Recovery.** Re-enter the command, using a local process name.

## 16

Exactly one structure must be supplied as input for this command.

**Cause.** A Subsystem Programmatic Interface (SPI) violation occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 30

The command specified an invalid volume name.  
The invalid volume name was *volume*.

*volume*

is the invalid volume name.

**Cause.** The command included an invalid volume name. Perhaps the name was too long, contained improper characters, or began with a character other than a dollar sign (\$).

**Effect.** The command fails.

**Recovery.** Verify that the volume name complies with the syntax rules: this name must consist of a dollar sign (\$) followed by one to seven alphanumeric characters, and the first alphanumeric character must be a letter. Then, re-enter the command with the proper volume name.

## 31

In this context, the command cannot specify a remote volume name. The remote volume name was *volume*.

*volume*

is the remote volume name.

**Cause.** The command contained a parameter that referred to a remote volume but only local volume names are allowed for that parameter.

**Effect.** The command fails.

**Recovery.** Re-enter the command, using a local volume name.

## 33

The command specified an invalid COPIES option. The COPIES option must specify a value between 1 and 2 for PARALLEL copies, or between 1 and 3 for SERIES copies.

**Cause.** The command specified an invalid number of dump copies to be made in the COPIES parameter. This parameter can indicate only 1 or 2 parallel copies, or from 1 to 3 serial copies.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a valid number of copies.



## 34

The command specified an invalid PRI option. The PRI parameter must be 204 for the TMP process, 211 for the TMFMON process, or a value between 1 and 199, inclusive, for all other processes.

**Cause.** The command specified an invalid process priority in the PRI option. The TMP process' priority must be 204. The TMFMON process' priority must be 211. The priority of all other processes must fall in the range 1 through 199, inclusive.

**Effect.** The command fails.

**Recovery.** Re-enter the command, using a valid priority number for the process specified.

## 35

The command specified an OUT filename that was invalid or that contained a wild-card character. The invalid OUT file name was *filename*.

*filename*

is the invalid or improper file name.

**Cause.** The command included an OUT option that specified an invalid or partially qualified file name. Perhaps the file name was too long or contained improper characters, or maybe the volume or subvolume specifiers were omitted.

**Effect.** The command fails.

**Recovery.** Verify that the file name complies with the syntax rules listed in [File Names and Process Identifiers](#) on page 3-6. Then, re-enter the command, using a fully-qualified, valid file name.

## 36

The command specified an invalid CPU parameter. Use a value from 0 through 15 inclusive to specify the CPU in which the process runs.

**Cause.** The command specified an invalid CPU parameter, perhaps using a value greater than 15 or a negative value. This parameter can only be an integer from 0 through 15.

**Effect.** The command fails.

**Recovery.** Re-enter the command, using an integer from 0 through 15, inclusive, for the CPU.

## 37

The command specified an invalid DENSITY parameter.

**Cause.** The command specified an invalid DENSITY parameter, which indicates the density at which dumps are written to magnetic tape.

**Effect.** The command fails.

**Recovery.** Re-enter the command, using correct density value. For information about determining an appropriate density, see [Section 4, TMFCOM Tape Media Considerations](#).

## 38

The command specified an invalid DRIVE parameter; either the drive name was invalid or contained a wild-card character. The invalid tape drive name was *device*.

*device*

is the invalid or improper tape drive name.

**Cause.** The command specified, in the DRIVE parameter, an invalid name for a device used for processing labeled tapes. (The DRIVE parameter is required if the LABEL parameter is set to ON.) Perhaps the drive name was too long or contained improper or wild-card characters.

**Effect.** The command fails.

**Recovery.** Verify that the tape drive name complies with the syntax rules listed in [File Names and Process Identifiers](#) on page 3-6. Then, re-enter the command with a proper tape drive name.

## 39

The command specified an invalid START file name, *filename*.

*filename*

is the invalid file name.

**Cause.** The command specified an invalid file name in the START parameter used to indicate the file at which an online dump begins in the specified volume and subvolume. Perhaps the name was too long or contained improper characters.

**Effect.** The command fails.

**Recovery.** Verify that the file name complies with the syntax rules listed in [File Names and Process Identifiers](#) on page 3-6. Then, re-enter the command with a valid file name.

## 40

The command specified an invalid SERIES value in the COPIES parameter.

**Cause.** The command specified an invalid number in the SERIES option of the COPIES parameter used to indicate the number of dump copies to be produced in serial order. This option produces dump copies one at a time, using a single tape drive. You can specify only 1, 2, or 3 copies in series.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying 1, 2, or 3 copies only.

## 41

The command specified an invalid VERIFYTAPE parameter.

**Cause.** The command specified an invalid VERIFYTAPE parameter used to determine whether tape verification is performed after a dump to tape takes place. You must specify either ON (to verify tape-record checksums, record sequence numbers, and tape format), or OFF (to suppress verification).

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying either ON or OFF in the VERIFYTAPE parameter.

## 42

The command specified an invalid BLOCKSIZE parameter.

**Cause.** The command specified an invalid BLOCKSIZE parameter used to set the data-block density at which dumps are written to tape. The value specified in BLOCKSIZE, multiplied by 1024, gives the block size in bytes. For BLOCKSIZE, you can specify one of the following values: 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52. The maximum block size is also limited by the density at which the tape drive writes data.

**Effect.** The command fails.

**Recovery.** Re-enter the command, using a valid BLOCKSIZE: an integral multiple of 4, in the range of 4 through 52.

## 43

A structure must be supplied with this command.

**Cause.** A Subsystem Programmatic Interface (SPI) violation occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 44

In this context, the command cannot specify a VOLUME name that contains wild-card characters.

**Cause.** The command included a volume name that contained wild-card characters; however, a complete volume name must be used in this command.

**Effect.** The command fails.

**Recovery.** Re-enter the command, using a complete volume name.

## 46

Subsystem Error: The priority order of the DISKMEDIA tokens is not consistent. Contact the Global Customer Support Center (GCSC) or your service provider.

**Cause.** A Subsystem Programmatic Interface (SPI) violation occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 47

The command must supply at least one filename.

**Cause.** The command did not contain a required file name specifier.

**Effect.** The command fails.

**Recovery.** Re-enter the command, including a valid file name.

## 49

In this context, the command requires a generic process name. A generic process name has the form “\$LAA”, where “L” is a letter, and the “A”’s are letters and/or numbers. The invalid generic process name was *process-name*.

*process-name*

is the invalid process name.

**Cause.** The command did not specify a generic process name where one is required. Such a name must begin with a dollar sign (\$), followed by a letter, followed by one or two letters or digits. The name cannot be more than four characters long.

**Effect.** The command fails.

**Recovery.** Verify that the process name begins with a dollar sign (\$), followed by a letter, followed by one or two letters or digits, and is no more than four characters long. Then, re-enter the command, using a valid generic process name.

## 50

The command specified an invalid RETAINDEPTH parameter. RETAINDEPTH must specify a value in the range 1 to 99, inclusive.

**Cause.** The command specified an invalid RETAINDEPTH parameter used to determine the number of online dump generations that can remain in the catalog. The number specified can range from 1 to 99.

**Effect.** The command fails.

**Recovery.** Re-enter the command, using a RETAINDEPTH value of from 1 through 99.

## 51

The command specified an invalid RELEASED parameter.

**Cause.** The command specified an invalid RELEASED parameter used to indicate whether an online or audit dump file is available for file recovery. This parameter can be set to OFF (to indicate that the file is available) or ON (to indicate that it is not available).

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying only either ON or OFF for the RELEASED parameter.

## 52

The command specified an invalid TAPEMEDIA STATUS parameter.

**Cause.** The command specified an invalid TAPEMEDIA STATUS parameter used to set the status of a tape volume, thus determining its availability for use in dumping. The STATUS parameter can indicate one of the following keywords only: RELEASED, ASSIGNED, BAD, or SCRATCH.

**Effect.** The command fails.

**Recovery.** Re-enter the command, setting STATUS to RELEASED, ASSIGNED, BAD, or SCRATCH. For further information about these values, see the INFO TAPEMEDIA command description in [Section 3, TMFCOM Commands](#).

## 54

The command specified an invalid LABEL parameter, or omitted this parameter.

**Cause.** The command specified an invalid LABEL parameter used to determine whether an output tape volume is labeled, or the command omitted this parameter altogether. LABEL, when present, can specify only ON (to request labeling) or OFF (to suppress labeling).

**Effect.** The command fails.

**Recovery.** Re-enter the command, including the LABEL parameter, set to either ON or OFF.

## 55

The command specified an invalid UNLOAD parameter.

**Cause.** The command specified an invalid UNLOAD parameter used to determine whether an output tape volume is unloaded after it is labeled. UNLOAD can specify only ON (to request unloading) or OFF (to suppress unloading).

**Effect.** The command fails.

**Recovery.** Re-enter the command with the UNLOAD parameter set to either ON or OFF.

## 56

The command specified an invalid master and/or auxiliary audit trail sequence number.

**Cause.** The command specified an invalid audit-trail file sequence number in the *filename* parameter, used to identify an online or audit dump to be added to the catalog.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a valid sequence number.

## 57

The command omitted a master and/or auxiliary audit trail sequence number.

**Cause.** The command did not include an audit-trail file sequence number.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a *filename* parameter that includes the audit-trail file sequence number.

## 58

In this context, the command must specify a file name that does not include wild-card characters. The invalid file name was *filename*.

*filename*

is the invalid file name.

**Cause.** The command included a file name that contained one or more wild-card characters (\*). However, wild-card characters are not allowed in this context.

**Effect.** The command fails.

**Recovery.** Re-enter the command without wild-card characters in the file name.

## 59

The command must supply at least one TAPEMEDIA entry.

**Cause.** The command did not specify tape media, but at least one tape-medium name is required.

**Effect.** The command fails.

**Recovery.** Re-enter the command with at least one tape medium specified.



## 60

The command specified an improper INVALID parameter.

**Cause.** The command specified an improper INVALID parameter used to determine whether a file is available for use in file recovery. You must set INVALID to either ON (to indicate that the file is available) or OFF (to indicate that it is not available).

**Effect.** The command fails.

**Recovery.** Re-enter the command with the INVALID parameter set to either ON or OFF.

## 61

The command specified an invalid MaxResp value. The specified value must be greater than or equal to -1. Contact the Global Customer Support Center (GCSC) or your service provider.

**Cause.** A Subsystem Programmatic Interface (SPI) error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 64

The command specified an invalid SYSTEM (node) name.

**Cause.** The command specified an invalid SYSTEM parameter used to indicate the node to which dumps are written or from which files are recovered.

**Effect.** The command fails.

**Recovery.** Verify that the node name consists of a backslash (\) followed by one to seven alphanumeric characters, and that the first alphanumeric character is a letter. Then, re-enter the command, specifying the valid node name in the SYSTEM parameter.

## 66

Subsystem Error: The value specified for the OVERWRITE LABEL option is invalid. This value can be only TRUE, FALSE, or NULL. Contact the Global Customer Support Center (GCSC) or your service provider.

**Cause.** A Subsystem Programmatic Interface (SPI) error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 67

An ADD TAPEMEDIA command contained an inconsistent combination of options. The LABEL option must always be present. If LABEL OFF is specified, the STATUS option must be present and the DRIVE, DENSITY, and UNLOAD options must NOT be present. If LABEL ON is specified, the DRIVE option must be present, the DENSITY and UNLOAD options are optional, and the STATUS option must NOT be present.

**Cause.** An ADD TAPEMEDIA command was entered with an incompatible set of options.

**Effect.** The command fails.

**Recovery.** Re-enter the command with a valid combination of options.

## 69

The command specified an invalid STATE parameter value, or omitted this parameter completely. This parameter must be set to either ABORTED or COMMITTED.

**Cause.** A RESOLVE TRANSACTION command specified an invalid option in the STATE parameter. You must set STATE to either ABORTED (to abort the transaction) or COMMITTED (to commit the transaction).

**Effect.** The command fails.

**Recovery.** Re-enter the command with STATE set to either ABORTED or COMMITTED.

## 70

When adding an audit dump into the catalog, the command must specify exactly one filename.

**Cause.** An ADD DUMPS command specified multiple file names in the *audit-trail-file* parameter, but only one file name is allowed. This name must be entered in the format shown under “Recovery.”

**Effect.** The command fails.

**Recovery.** Re-enter the command, using just one file name entered in the following format:

\$ZTMFAT.ZTMFAT.[*file-id*]

## 72

The command can only specify a maximum of 16 tape volumes with the TAPEMEDIA parameter.

**Cause.** The command specified more than 16 tape volumes in the TAPEMEDIA parameter, but only a maximum of 16 volumes are allowed.

**Effect.** The command fails.

**Recovery.** Re-enter the command specifying 16 or fewer volumes in TAPEMEDIA.

## 73

The command specified an invalid SERIAL parameter. The value must be greater than 63. Your value was *serial-number*.

*serial-number*

is the invalid serial number.

**Cause.** The command specified an invalid serial number in the SERIAL parameter used to identify a dump to be added to the catalog. This serial number must be 64 or greater.

**Effect.** The command fails.

**Recovery.** Re-enter the command with SERIAL indicating a dump serial number of 64 or greater.

## 74

Either a DUMP FILES command specified more than 16 disk volumes in the DISKMEDIA parameter, or an ALTER AUDITDUMP command would have resulted in a total of more than 16 disk volumes configured for the dump process.

**Cause.** Either a DUMP FILES or an ALTER AUDITDUMP command specified too many disk volumes, as follows:

- The DUMP FILES command named too many disk volumes in the DISKMEDIA parameter, which identifies the disk volumes that receive the dumps. No more than 16 volumes can be specified.
- The ALTER AUDITDUMP command attempted to add too many disk volumes to the dump process configuration. The total number of volumes is the sum of the number of disk media already configured, plus the number added by the ADDDISKMEDIA parameter, minus the number deleted by the DELDISKMEDIA parameter; at any particular time, this total cannot exceed 16 volumes.

**Effect.** The command fails.

**Recovery.** Re-enter the DUMP FILES command with DISKMEDIA set to the proper limit, or re-enter the ALTER AUDITDUMP command with the combination of already-configured volumes, volumes added by ADDDISKMEDIA, and volumes removed by DELDISKMEDIA totalling no more than 16.

## 75

The command can only specify DISKMEDIA with MEDIUM DISK.

**Cause.** An ALTER AUDITDUMP command specified the ADDDISKMEDIA or DELDISKMEDIA parameter, while the MEDIUM parameter was set to TAPE. ADDDISKMEDIA and DELDISKMEDIA add or delete, respectively, individual disk volumes from the dump process configuration. MEDIUM specifies the kind of medium on which the dump is recorded and can be set to DISK (for disk volumes) or TAPE (for tape volumes). ADDDISKMEDIA or DELDISKMEDIA can only be used when MEDIUM is set to DISK.

**Effect.** The command fails.

**Recovery.** Re-enter the command with the ADDDISKMEDIA or DELDISKMEDIA specification removed or with MEDIUM set to DISK.

## 76

One of the options specified is not allowed with MEDIUM DISK.

**Cause.** The command specified a parameter that is not compatible with the MEDIUM DISK parameter. MEDIUM DISK specifies the kind of medium on which dumps are recorded, but the incompatible parameter might be one that applies only to tape-based dumps—for example, BLOCKSIZE, COPIES, SYSTEM, or VERIFYTAPE.

**Effect.** The command fails.

**Recovery.** Re-enter the command without the incompatible parameter.

## 77

The command specified a DISKMEDIA name that was not correctly formed. The invalid DISKMEDIA was: *filename*.

*filename*

is the invalid file name.

**Cause.** The command specified one (or more) DISKMEDIA names as the destination disk volumes for dumps, but this name was not entered in a valid format. The formats permitted are the following:

*remote-node.volume.prefix*  
*local-node.volume.prefix*  
*local-node.volume*  
*volume*  
*volume.prefix*

**Effect.** The command fails.

**Recovery.** Verify that the file name complies with the syntax rules listed in [File Names and Process Identifiers](#) on page 3-6 and with the additional restrictions noted in this message description. Then, re-enter the command, specifying the correctly formed name for DISKMEDIA.

## 78

In this context, the command cannot RESET the DISKMEDIA.

**Cause.** A DUMP FILES command attempted to reset the DISKMEDIA attribute, but this attribute cannot be reset in the current context.

**Effect.** The command fails.

**Recovery.** Re-enter the command without specifying disk media reset.

## 79

In this context, the command cannot include wild-cards in the DISKMEDIA parameter list. The invalid DISKMEDIA was:  
*filename*.

*filename*

is the invalid file name.

**Cause.** The command contained one or more wild-card characters (\*) in its DISKMEDIA parameter list. However, only fully qualified medium identifiers must be used in this command.

**Effect.** The command fails.

**Recovery.** Re-enter the command, using only fully qualified medium identifiers.

## 80

TAPEMEDIA is only allowed with MEDIUM TAPE.

**Cause.** An ALTER AUDITDUMP command specified a tape-oriented parameter, while the MEDIUM parameter was set to DISK. The tape-oriented parameter might have been BLOCKSIZE, COPIES, SYSTEM, or VERIFYTAPE. MEDIUM specifies the kind of medium on which the dump is recorded, and can be set to TAPE (for tape volumes) or DISK (for disk volumes). BLOCKSIZE, COPIES, SYSTEM, or VERIFYTAPE can only be used when MEDIUM is set to TAPE.

**Effect.** The command fails.

**Recovery.** Re-enter the command with the tape-oriented specification removed or with MEDIUM set to TAPE.

## 81

The command requires at least one DISKMEDIA volume with MEDIUM DISK.

**Cause.** A DUMP FILES or ALTER AUDITDUMP command included the MEDIUM DISK parameter but did not specify a particular disk volume as the dump destination. For example, at least one disk volume must be specified in the DISKMEDIA parameter of the DUMP FILES command or in the ADDDISKMEDIA parameter of the ALTER AUDITDUMP command.

**Effect.** The command fails.

**Recovery.** Re-enter the command with at least one volume specified in DISKMEDIA or ADDDISKMEDIA.

## 82

The command specified an invalid DISKMEDIA name for a remote disk volume. The invalid DISKMEDIA is: *filename*.

*filename*

is the invalid file name.

**Cause.** The command specified an invalid name for a remote disk volume in the DISKMEDIA parameter.

**Effect.** The command fails.

**Recovery.** Verify that the file name and particularly the *volume* component, complies with the syntax rules listed in [File Names and Process Identifiers](#) on page 3-6. Then, re-enter the command with a valid DISKMEDIA file name.

## 83

In this context, the command cannot RESET the medium.

**Cause.** A DUMP FILES command attempted to reset the attribute specifying the output medium (disk or tape) to which online dumps are directed, but this medium cannot be reset in the current context.

**Effect.** The command fails.

**Recovery.** Re-enter the command without specifying a media reset.

## 87

Subsystem Error: When a second system for parallel dumps is specified, the first system must also be specified. Contact the Global Customer Support Center (GCSC) or your service provider.

**Cause.** A Subsystem Programmatic Interface (SPI) violation occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 89

The command specified an invalid IGNOREDATAERRORS option.

**Cause.** An ABORT TRANSACTION command specified an invalid option in the IGNOREDATAERRORS parameter. This parameter determines whether TMF ignores data errors while undoing the effects of a transaction and can only be set to either ON or OFF.

**Effect.** The command fails.

**Recovery.** Re-enter the command with IGNOREDATAERRORS set to either ON or OFF.

## 90

The command specified both a wild-card volume name and a specific volume name.

**Cause.** The command specified both a wild-card name and a specific volume name. The syntax rules allow either kind of name, but not both kinds, in the same command.

**Effect.** The command fails.

**Recovery.** Re-enter the command with the correct volume names.

## 91

Subsystem Error: A data volume command required at least one volume name, but none was specified. Contact the Global Customer Support Center (GCSC) or your service provider.

**Cause.** A Subsystem Programmatic Interface (SPI) error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.



## 92

Subsystem Error: An ADD DATAVOLS or ALTER DATAVOLS command specified an erroneous RECOVERYMODE value. Contact the Global Customer Support Center (GCSC) or your service provider.

**Cause.** A Subsystem Programmatic Interface (SPI) error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 93

Subsystem Error: An ADD DATAVOLS command specified an erroneous IGNOREPREVIOUSCONFIG value. Contact the Global Customer Support Center (GCSC) or your service provider.

**Cause.** A Subsystem Programmatic Interface (SPI) error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 94

Subsystem Error: A STATUS DATAVOLS command specified an erroneous STATE value. Contact the Global Customer Support Center (GCSC) or your service provider.

**Cause.** A Subsystem Programmatic Interface (SPI) error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 95

Subsystem Error: A DELETE DATAVOLS command specified an erroneous ALLOWINCONSISTENCY value. Contact the Global Customer Support Center (GCSC) or your service provider.

**Cause.** A Subsystem Programmatic Interface (SPI) error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 96

Subsystem Error: A DISABLE DATAVOLS command specified an erroneous ABRUPT value. Contact the Global Customer Support Center (GCSC) or your support representative.

**Cause.** A Subsystem Programmatic Interface (SPI) error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 97

The version of TMFCOM in use is out of date.

**Cause.** The command was issued through a version of TMFCOM that is out of date.

**Effect.** The command fails.

**Recovery.** Re-enter the command, using an appropriate version of TMFCOM.

## 99

TMF must be started for this command.

**Cause.** The command attempted to modify the TMF configuration, but requires TMF to be running to take effect.

**Effect.** The command fails.

**Recovery.** Enter (or ask your system manager to enter) the START TMF command to activate TMF. Then, re-enter the previous command.

## 101

Subsystem Error: A RECOVER FILES command specified the FROMARCHIVE option with an illegal value. Contact the Global Customer Support Center (GCSC) or your service provider.

**Cause.** A Subsystem Programmatic Interface (SPI) error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 102

Subsystem Error: A RECOVER FILES command specified the PLANONLY option with an illegal value. Contact the Global Customer Support Center (GCSC) or your service provider.

**Cause.** A Subsystem Programmatic Interface (SPI) error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 103

Subsystem Error: A RECOVER FILES command specified the TOFIRSTPURGE option with an illegal value. Contact the Global Customer Support Center (GCSC) or your service provider.

**Cause.** A Subsystem Programmatic Interface (SPI) error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 104

Subsystem Error: An ABORT TRANSACTION command specified the AVOIDHANGING option with an illegal value. Contact the Global Customer Support Center (GCSC) or your service provider.

**Cause.** A Subsystem Programmatic Interface (SPI) error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 105

The TMP ran out of resources to process the command.

**Cause.** A STATUS TRANSACTIONS command was entered, but the TMP ran out of system resources needed to execute the command.

**Effect.** The command fails.

**Recovery.** This is only a transient problem; re-enter the command.

## 106

Subsystem Error: Empty children were encountered.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 108

Subsystem Error: A STATUS TRANSACTION command specified the STATUS DETAIL option with an illegal value. Contact the Global Customer Support Center (GCSC) or your service provider.

**Cause.** A Subsystem Programmatic Interface (SPI) error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 110

The command specified an erroneous STATE value.

**Cause.** A STATUS OPERATIONS command specified a value other than INPROGRESS, FINISHED, or ALL for the STATE parameter.

**Effect.** The command fails.

**Recovery.** Re-enter the command with STATE set to INPROGRESS, FINISHED, or ALL.

## 111

The command specified an erroneous TYPE value.

**Cause.** A STATUS OPERATIONS command specified an improper value for the TYPE parameter. (See the command syntax description for the list of acceptable values.)

**Effect.** The command fails.

**Recovery.** Re-enter the command with TYPE set to a proper value.

## 112

The command specified an erroneous BEGINTIME value.

**Cause.** A STATUS OPERATIONS command specified an improper value for the BEGINTIME parameter. (See the command syntax description for the acceptable values.)

**Effect.** The command fails.

**Recovery.** Re-enter the command with BEGINTIME set to a proper value.

## 113

The command specified an erroneous *operation-number* value.

**Cause.** A STATUS OPERATIONS command specified an improper value for the *operation-number* parameter. (See the command syntax description for the acceptable values.)

**Effect.** The command fails.

**Recovery.** Re-enter the command with *operation-number* set to a proper value.

## 114

The command specified an erroneous TRANSCOUNTTHRESH value.

**Cause.** An ALTER BEGINTRANS command specified one or more improper values for the TRANSCOUNTTHRESH parameter. (See the command syntax description for the acceptable values.)

**Effect.** The command fails.

**Recovery.** Re-enter the command with TRANSCOUNTTHRESH set to proper values.

## 115

The command specified an erroneous TMFLIBMEMTHRESH value.

**Cause.** An ALTER BEGINTRANS command specified one or more improper values for the TMFLIBMEMTHRESH parameter. (See the command syntax description for the acceptable values.)

**Effect.** The command fails.

**Recovery.** Re-enter the command with TMFLIBMEMTHRESH set to proper values.

## 116

The command specified an erroneous TMPMEMTHRESH value.

**Cause.** An ALTER BEGINTRANS command specified one or more improper values for the TMPMEMTHRESH parameter. (See the command syntax description for the acceptable values.)

**Effect.** The command fails.

**Recovery.** Re-enter the command with TMPMEMTHRESH set to proper values.

## 117

The command specified an erroneous TRANSPERCPU value.

**Cause.** An ALTER BEGINTRANS command specified one or more improper values for the TRANSPERCPU parameter. (See the command syntax description for the acceptable values.)

**Effect.** The command fails.

**Recovery.** Re-enter the command with TRANSPERCPU set to proper values.

## 118

The command specified an erroneous AUTOABORT value.

**Cause.** An ALTER BEGINTRANS command specified an improper value for the AUTOABORT parameter. (See the command syntax description for the acceptable values.)

**Effect.** The command fails.

**Recovery.** Re-enter the command with AUTOABORT set to a proper value.

## 119

The command specified a DISKMEDIA name that was not correctly formed. The invalid DISKMEDIA was: *filename*.

*filename*

is the invalid file name.

**Cause.** An ALTER AUDITDUMP command specified one or more DISKMEDIA names in the ADDDISKMEDIA or DELDISKMEDIA parameter as the destination disk volumes for dumps, but this name was not entered in a valid format. The formats permitted are the following:

*remote-node.volume.prefix*  
*local-node.volume.prefix*  
*volume.prefix*

**Effect.** The command fails.

**Recovery.** Verify that the file name complies with the syntax rules listed in [File Names and Process Identifiers](#) on page 3-6 and with the additional restrictions noted in this message description. Then, re-enter the command, specifying the correctly formed name for DISKMEDIA.

## 120

The command specified an invalid DISKMEDIA name for a remote disk volume.  
The invalid DISKMEDIA was: *filename*.

*filename*

is the invalid file name.

**Cause.** An ALTER AUDITDUMP command specified an invalid name for a remote disk volume in the ADDDISKMEDIA or DELDISKMEDIA parameter.

**Effect.** The command fails.

**Recovery.** Verify that the file name and particularly the *volume* component complies with the syntax rules listed in [File Names and Process Identifiers](#) on page 3-6. Then, re-enter the command with a valid ADDDISKMEDIA or DELDISKMEDIA file name.

## 121

The volume name *volume*, for a volume being added as an active audit volume, is invalid.

*volume*

is the name of the volume specified in the command.

**Cause.** An ALTER AUDITTRAIL command specified an active-audit volume name that was not valid. For example, perhaps the name was too long or contained wild-card characters.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the ADDACTIVEVOL[S] parameter of the ALTER AUDITTRAIL command. Then, re-enter the command, using an appropriate volume name.



## 122

The volume name *volume*, for a volume being added as an overflow audit volume, is invalid.

*volume*

is the name of the volume specified in the command.

**Cause.** An ALTER AUDITTRAIL command specified an overflow-audit volume name that was not valid. For example, perhaps the name was too long or contained wild-card characters.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the ADDOVERFLOWVOL[S] parameter of the ALTER AUDITTRAIL command. Then, re-enter the command, using an appropriate volume name.

## 123

The volume name *volume*, for a volume being added as a restore audit volume, is invalid.

*volume*

is the name of the volume specified in the command.

**Cause.** An ALTER AUDITTRAIL command specified a restore-audit volume name that was not valid. For example, perhaps the name was too long or contained wild-card characters.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the ADDRESTOREVOL[S] parameter of the ALTER AUDITTRAIL command. Then, re-enter the command, using an appropriate volume name.

## 124

The volume name *volume*, for volume being deleted as an active audit volume, is invalid.

*volume*

is the name of the volume specified in the command.

**Cause.** An ALTER AUDITTRAIL command specified an active-audit volume name that was not valid. For example, perhaps the name was too long or contained wild-card characters.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the DELETEACTIVEVOL[S] parameter of the ALTER AUDITTRAIL command. Then, re-enter the command, using an appropriate volume name.

## 125

The volume name *volume*, for a volume being deleted as an overflow audit volume, is invalid.

*volume*

is the name of the volume specified in the command.

**Cause.** An ALTER AUDITTRAIL command specified an overflow-audit volume name that was not valid. For example, perhaps the name was too long or contained wild-card characters.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the DELETEOVERFLOWVOL[S] parameter of the ALTER AUDITTRAIL command. Then, re-enter the command, using an appropriate volume name.

## 126

The volume name *volume*, for volume being deleted as a restore audit volume, is invalid.

*volume*

is the name of the volume specified in the command.

**Cause.** An ALTER AUDITTRAIL command specified a restore-audit volume name that was not valid. For example, perhaps the name was too long or contained wild-card characters.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the DELETERESTOREVOL[S] parameter of the ALTER AUDITTRAIL command. Then, re-enter the command, using an appropriate volume name.

## 127

The volume name *volume*, for a volume being added as an active audit volume, is invalid.

*volume*

is the name of the volume specified in the command.

**Cause.** An ADD AUDITTRAIL command specified an active-audit volume name that was not valid. For example, perhaps the name was too long or contained wild-card characters.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the ACTIVEVOL[S] parameter of the ADD AUDITTRAIL command. Then, re-enter the command, using an appropriate volume name.

## 128

The volume name *volume*, for a volume being added as an overflow audit volume, is invalid.

*volume*

is the name of the volume specified in the command.

**Cause.** An ADD AUDITTRAIL command specified an overflow-audit volume name that was not valid. For example, perhaps the name was too long or contained wild-card characters.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the OVERFLOWVOL[S] parameter of the ADD AUDITTRAIL command. Then, re-enter the command, using an appropriate volume name.

## 129

The volume name *volume*, for a volume being added as a restore audit volume, is invalid.

*volume*

is the name of the volume specified in the command.

**Cause.** An ADD AUDITTRAIL command specified a restore-audit volume name that was not valid. For example, perhaps the name was too long or contained wild-card characters.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the RESTOREVOL[S] parameter of the ADD AUDITTRAIL command. Then, re-enter the command, using an appropriate volume name.

## 130

The volume name *volume* is invalid in the context used.

*volume*

is the name of the volume specified in the command.

**Cause.** The command referred to a volume name that was not valid within the context of this command. For example, perhaps the command used a remote volume name where only local names are permitted or specified a name containing wild-card characters where only complete names are permitted.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command. Then, re-enter the command, using an appropriate volume name.

## 131

Subsystem Error: No AuditTrailId was specified.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 132

Subsystem Error: The specified AuditTrailId is invalid.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 133

Subsystem Error: The value of the AuditDump attribute must be ON or OFF.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 134

The value specified for FILESPERVOLUME is invalid.

**Cause.** The command specified a value for the FILESPERVOLUME parameter that did not fall within the proper limit, an integer from 1 through 100. This parameter determines the number of files in the audit trail that reside on each active-audit volume configured.

**Effect.** The command fails.

**Recovery.** Re-enter the command, setting FILESPERVOLUME to an integer from 1 to 100, or omitting this parameter altogether so that the default value of 5 applies.

## 135

The value specified for FILESIZE is invalid.

**Cause.** The command specified a value for the FILESIZE parameter that did not fall within the proper limit, an integer from 1 through 2047. This parameter specifies the size of the audit-trail file, in megabytes.

**Effect.** The command fails.

**Recovery.** Re-enter the command, setting FILESIZE to an integer from 1 to 2047, or omitting this parameter altogether so that the default value of 100 applies.

## 136

The value specified for OVERFLOWTHRESHOLD is invalid.

**Cause.** An ADD AUDITTRAIL or ALTER AUDITTRAIL command specified a value for the OVERFLOWTHRESHOLD parameter that did not fall within the proper limit, an integer from 50 to 100. This parameter determines the percentage of audit trail capacity in use above which TMF begins overflow copying.

**Effect.** The command fails.

**Recovery.** Re-enter the command, setting OVERFLOWTHRESHOLD to an integer from 50 to 100 or omitting this parameter altogether so that the default value of 80 (percent) applies.

## 137

The value specified for BEGINTRANSDISABLE is invalid.

**Cause.** An ADD AUDITTRAIL or ALTER AUDITTRAIL command specified a value for the BEGINTRANSDISABLE parameter that did not fall within the proper limit, an integer from 50 to 100. This parameter determines the percentage of audit-trail capacity that can be consumed before TMF disables new transactions.

**Effect.** The command fails.

**Recovery.** Re-enter the command, setting BEGINTRANSDISABLE to an integer ranging from 50 to 100 or omitting this parameter altogether so that the default value of 90 (percent) applies.

## 138

Subsystem Error: The specified volume type for the audit-trail volume is invalid.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This command should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 139

Subsystem Error: This command can specify only one AuditTrailId.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This command should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 140

A STATUS OPERATIONS command that specified the TYPE parameter also specified an operation number.

**Cause.** A STATUS OPERATIONS command that included the TYPE parameter also referred to a specific TMF operation. When TYPE is included, however, the command must specify the wild-card (\*) character rather than a particular operation number. TYPE then acts as a filter to display all operations of the type indicated.

**Effect.** The command fails.

**Recovery.** Re-enter the command, substituting the wild-card character in place of the operation number.

## 141

A STATUS OPERATIONS command that specified the STATE parameter also specified an operation number.

**Cause.** A STATUS OPERATIONS command that included the STATE parameter also referred to a specific TMF operation. When STATE is included, however, the command must specify the wild-card (\*) character rather than a particular operation number. STATE then acts as a filter to display all operations in the state indicated.

**Effect.** The command fails.

**Recovery.** Re-enter the command, substituting the wild-card character in place of the operation number.



## 142

A STATUS OPERATIONS command that specified the BEGINTIME parameter also specified an operation number.

**Cause.** A STATUS OPERATIONS command that included the BEGINTIME parameter also referred to a specific TMF operation. When BEGINTIME is included, however, the command must specify the wild-card (\*) character rather than a particular operation number. BEGINTIME then acts as a filter to display all operations that began at or after the time indicated.

**Effect.** The command fails.

**Recovery.** Re-enter the command, substituting the wild-card character in place of the operation number.

## 143

The command specified file-set *file-set*, using invalid syntax.

*file-set*

is the file-set specified with invalid syntax.

**Cause.** The command specified a file-set, but did not follow the syntax rules for identifying file-sets.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for file-set names, and re-enter the command using the proper syntax.

## 144

A CANCEL OPERATIONS command attempted to cancel a TMF operation other than a file-recovery or dump-files operation.

**Cause.** A CANCEL OPERATIONS command attempted to cancel an operation that was not a file-recovery or dump-files operation. However, this command can only be used for those two types of operation.

**Effect.** The command fails.

**Recovery.** Use the STATUS OPERATIONS command to display the current operations and their types. Then, re-enter the CANCEL OPERATIONS command, specifying an operation whose type is listed as either Recoveries or DumpFiles.

## 145

Subsystem Error: The SHOTGUN buffer received by the TMFSERVE process did not contain the required token ZTMF-Tkn-OperationNum. Contact the Global Customer Support Center (GCSC) or your service provider.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 146

The command did not specify a valid dump serial number.

**Cause.** A RELOCATE DISKDUMPS command specified a dump serial number less than 64.

**Effect.** The command fails.

**Recovery.** Retry the command, specifying the correct serial number.

## 147

The command specified a wild-card character along with other dump serial numbers.

**Cause.** In a RELOCATE DISKDUMPS command, the wild card character (\*) was used with dump serial numbers. While multiple dump serial numbers may be specified in a single RELOCATE DISKDUMPS command, serial numbers and the wild card character cannot both be specified.

**Effect.** The command fails.

**Recovery.** Retry the command, using either serial numbers or a single wild card character for the *serial-number* parameter; do not mix the wild card character and serial numbers in the *serial-number* parameter.

## 148

The command did not specify a disk media location for the OLDDMLOC option.

**Cause.** A RELOCATE DISKDUMPS command did not specify a disk media location in the OLDDMLOC parameter.

**Effect.** The command fails.

**Recovery.** Retry the command, specifying a valid disk media location in the OLDDMLOC parameter.

## 149

The command specified an invalid disk media location for the OLDDMLOC option. The invalid disk media location was *dmloc*.

*dmloc*

is the invalid disk media location specified in OLDDMLOC parameter.

**Cause.** A RELOCATE DISKDUMPS command specified the disk media location for the OLDDMLOC option in an invalid format.

**Effect.** The command fails.

**Recovery.** Check the syntax for the OLDDMLOC option and re-enter the command.

## 150

The command did not specify a disk media location for the NEWDMLOC option.

**Cause.** A RELOCATE DISKDUMPS command did not specify the NEWDMLOC option.

**Effect.** The command fails.

**Recovery.** Retry the command, specifying a valid disk media location in the NEWDMLOC parameter.

## 151

The command specified an invalid disk media location for the NEWDMLOC option. The invalid disk media location was *dmloc*.

*dmloc*

is the invalid disk media location specified in the NEWDMLOC parameter.

**Cause.** A RELOCATE DISKDUMPS command specified the disk media location for the NEWDMLOC option in an invalid format.

**Effect.** The command fails.

**Recovery.** Check the syntax for the NEWDMLOC option and re-enter the command.

## 152

The command specified a wild-card for the OLDDMLOC option but not for the NEWDMLOC option. The NEWDMLOC option was *dmloc*.

*dmloc*

is the disk media location specified in the NEWDMLOC parameter.

**Cause.** In a RELOCATE DISKDUMPS command, the wild card character (\*) was used for the OLDDMLOC parameter, but not for the NEWDMLOC parameter.

**Effect.** The command fails.

**Recovery.** Retry the command, using either specific volume names only or wild-card characters only for both parameters; do not mix wild card characters and specific volume names in the OLDDMLOC and NEWDMLOC parameters. Specify wildcard characters for OLDDMLOC and NEWDMLOC only when changing the system names of the disk media in the TMF catalog.

## 153

The command specified more than one disk media location for the OLDDMLOC option.

**Cause.** A RELOCATE DISKDUMPS command specified multiple disk media locations in the OLDDMLOC parameter.

**Effect.** The command fails.

**Recovery.** Retry the command, specifying a single value for the OLDDMLOC parameter.

## 154

The command specified more than one disk media location for the NEWDMLOC option.

**Cause.** A RELOCATE DISKDUMPS command specified multiple disk media locations in the NEWDMLOC parameter.

**Effect.** The command fails.

**Recovery.** Retry the command, specifying a single value for the NEWDMLOC parameter.

## 155

TMF found no disk dumps that matched the requested criteria.

**Cause.** TMF found no dumps listed in the catalog that matched the serial numbers or old disk media locations specified in the command.

**Effect.** None.

**Recovery.** Informative message only; no corrective action required.

## 156

The command specified two or more mutually exclusive options.

**Cause.** The ALTER TMF command specified both the SWITCHPROCESS and CONFIGVOL options.

**Effect.** The command fails.

**Recovery.** Retry the command, specifying only one of the options.

## 157

The command omitted at least one required option.

**Cause.** An ALTER TMF command was issued without either a SWITCHPROCESS or CONFIGVOL option.

**Effect.** The command fails.

**Recovery.** Retry the command, specifying one of the options.

## 158

The command must specify only one process-type option.

**Cause.** An ALTER PROCESS command specified more than one process type (for example, FILERECOVERY and BACKOUT).

**Effect.** The command fails.

**Recovery.** Retry the command, specifying only one process type.

## 160

The command option WHEREPHYSVOLIS supplied an invalid volume name.

**Cause.** A RECOVER FILES command specified an invalid volume name in the WHEREPHYSVOLIS parameter. Perhaps the name was too long, contained improper characters, or began with a character other than a dollar sign (\$).

**Effect.** The command fails.

**Recovery.** Verify that the volume name complies with the syntax rules: this name must consist of a dollar sign (\$) followed by one to seven alphanumeric characters, and the first alphanumeric character must be a letter. Then, re-enter the command, specifying a valid volume name.

## 161

The command option TOPHYSVOL supplied an invalid volume name.

**Cause.** A RECOVER FILES command specified an invalid volume name in the TOPHYSVOL parameter. Perhaps the name was too long, contained improper characters, or began with a character other than a dollar sign (\$).

**Effect.** The command fails.

**Recovery.** Verify that the volume name complies with the syntax rules: this name must consist of a dollar sign (\$) followed by one to seven alphanumeric characters, and the first alphanumeric character must be a letter. Then, re-enter the command, specifying a valid volume name.

## 162

The command option TOPHYSVOL allows only one physical volume name.

**Cause.** A RECOVER FILES command specified more than one physical volume name in the TOPHYSVOL parameter.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying just one volume name in the TOPHYSVOL parameter.

## 163

The command specified an invalid RECOVERVDPPHASE1 parameter.

**Cause.** A RECOVER FILES command specified the RECOVERVDPPHASE1 parameter incorrectly.

**Effect.** The command fails.

**Recovery.** Check the command syntax, ensuring the RECOVERVDPPHASE1 keyword is placed properly and spelled correctly. Then, re-enter the command, specifying the keyword correctly.

## 166

The RECOVERVDPPHASE1 option requires that the command also specify the TOPHYSVOL option.

**Cause.** A RECOVER FILES command included the RECOVERVDPPHASE1 option (which recovers the underlying catalog for the virtual disk process) but omitted the TOPHYSVOL option (which indicates the target volume on which to place the catalog).

**Effect.** The command fails.

**Recovery.** Re-enter the command, including both the RECOVERVDPPHASE1 and TOPHYSVOL options.

## 167

The command specified an invalid RECRMOUNT value.

**Cause.** An ALTER BEGINTRANS command contained an invalid value for the RECRMOUNT attribute.

**Effect.** The command fails.

**Recovery.** Retry the command with a valid value for the RECRMOUNT attribute. The valid values are 16 through 16384, inclusive. The value specified will be automatically rounded to the nearest multiple of 16.

## 168

The command specified an invalid RMOPENPERCPU value.

**Cause.** An ALTER BEGINTRANS command contained an invalid value for the RMOPENPERCPU attribute.

**Effect.** The command fails.

**Recovery.** Retry the command with a valid value for the RMOPENPERCPU attribute. The valid values are 16 through 1024, inclusive. The value specified will be automatically rounded to the nearest multiple of 16.

## 169

The command specified an invalid BRANCHESPERRM value.

**Cause.** An ALTER BEGINTRANS command contained an invalid value for the BRANCHESPERRM attribute.

**Effect.** The command fails.

**Recovery.** Retry the command with a valid value for the BRANCHESPERRM attribute. The valid values are 16 through 1024, inclusive. The value specified will be automatically rounded to the nearest multiple of 16.



## 170

The resource manager name *rmname* is invalid.

*rmname*

is the invalid resource manager name specified in the command.

**Cause.** The resource manager name specified in *rmname* is invalid.

**Effect.** The command fails for this resource manager name.

**Recovery.** Retry the command with a valid resource manager name. Valid resource manager names have a maximum length of 128 characters, must be alphanumeric or include the special characters \$ \_ - ^ @ & and must begin with an alphabetic character. In some commands, resource manager names cannot begin with the string ZZ-VOLATILE-.

## 171

This command must specify a resource manager name.

**Cause.** The command requires at least one resource manager name, but did not include that name.

**Effect.** The command fails.

**Recovery.** Although TMF can return a programmatic equivalent of this message to a user program, this message should never appear in a TMFCOM display. Make a note of the command entered and the circumstances surrounding its entry. Then, contact the GCSC or your service provider.

## 172

The command specified the wild-card character where a resource manager name is required.

**Cause.** An ADD RESOURCEMANAGER, DELETE RESOURCEMANAGER, or CLOSE RESOURCEMANAGER command included a wild-card character (\*) where a resource manager name should appear.

**Effect.** The command fails.

**Recovery.** Retry the command, using a valid resource manager name in place of the wild-card character.

## 173

The command specified both a wild-card resource manager name and a specific volume name.

**Cause.** An INFO RESOURCEMANAGER or a STATUS RESOURCEMANAGER command included both a wild-card character (\*) and a resource manager name to identify the resource managers for which to display information. Although TMFCOM accepts either of these entries, it does not accept both in the same request.

**Effect.** The command fails.

**Recovery.** Retry the command, using either a valid resource manager name or the wild-card character, but not both.

## 174

The command did not specify the resource manager owner userid.

**Cause.** An ADD RESOURCEMANAGER command did not specify the userid or user name of the resource manager's owner, but one of these identifiers must be supplied.

**Effect.** The command fails.

**Recovery.** Retry the command, specifying either the proper user-id or user name in the OWNER parameter.

## 175

The command specified an invalid resource manager state.

**Cause.** A STATUS RESOURCEMANAGER command specified an invalid entry in the STATE parameter.

**Effect.** The command fails.

**Recovery.** Retry the command, either omitting the STATE parameter or specifying one of the following states: CREATED, CLOSED, OPEN, RECOVERING, CRASHED, VOLATILE, or PROCESS.

## 176

The command can specify only one resource manager name.

**Cause.** The command specified multiple resource managers, but only one resource manager is allowed.

**Effect.** The command fails.

**Recovery.** Retry the command, specifying only the correct single resource manager.

## 201

ABORT TRANSACTION command failed because of *errcode*.

*errcode*

supplies a numeric code that further defines the error and identifies the subsystem or system procedure that transmitted this code: TMF, the File System, the NEWPROCESS process-control procedure, or the Enform query language and report writer.

**Cause.** While an ABORT TRANSACTION command was in process, the condition indicated by *errcode* occurred.

**Effect.** The command fails.

**Recovery.** Refer to the following manuals for a description of the error code:

| Transmitting Subsystem | Manual                              | Appendix Containing Description |
|------------------------|-------------------------------------|---------------------------------|
| TMF                    | This reference manual               | This appendix                   |
| File system            | <i>Operator Messages Manual</i>     | "File-System Errors" Appendix   |
| NEWPROCESS procedure   | <i>Operator Messages Manual</i>     | "NEWPROCESS Errors" Appendix    |
| Enform subsystem       | <i>Enform Plus Reference Manual</i> | "Error Messages" Appendix       |

For further details about understanding and correcting File System and NEWPROCESS errors, see the *Guardian Procedure Errors and Messages Manual*.

Correct the error condition reported, if necessary, and re-enter the command.

## 202

The TAPEMEDIA "*tape-volume*" is already listed in the TAPEMEDIA specification.

*tape-volume*

is the tape volume name.

**Cause.** The command specified the tape volume named *tape-volume* more than once.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying *tape-volume* only once.

## 208

TMFCOM does not recognize this command.

**Cause.** This command is not defined in TMF.

**Effect.** The command fails.

**Recovery.** Check the command against those defined in this manual, or enter HELP ALL or HELP \* to display a list of all TMFCOM commands. Then, select and enter a valid command.

## 215

TMF must be stopped before this command can be used.

**Cause.** This command should only be issued when TMF is not running.

**Effect.** The command fails.

**Recovery.** Enter a STOP TMF command to shut down TMF in an orderly way. Then, enter the command that resulted in this error message.

## 236

The ALTER BEGINTRANS command specified an invalid AUTOABORT parameter. AUTOABORT must indicate either a *timeout* value in the proper range or OFF.

**Cause.** The command specified an invalid value for the AUTOABORT parameter.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying either an appropriate *timeout* value for the AUTOABORT parameter, or 0 to turn the autoabort feature off. The *timeout* value can range from 20 seconds to one of the following maximum values: 21474836 seconds, 357913 minutes, or 5965 hours. For more information about the timeout value, see the ALTER BEGINTRANS command description in [Section 3, TMFCOM Commands](#).

## 249

TMF must be configured. Configuration is performed through system generation.

**Cause.** This command cannot be entered before TMF is configured.

**Effect.** The command fails.

**Recovery.** Ask your system manager to configure TMF through system generation. If necessary, also start TMF. Then, re-enter this command.

## 253

This command can only be used when TMF is ACTIVE or STOPPING.

**Cause.** You entered a command that can only be issued while TMF is running.

**Effect.** The command fails.

**Recovery.** Start TMF and then re-enter this command.

## 254

The user ID SUPER.SUPER (255,255) must be used to execute this command. The user ID of this user is *groupid,userid*.

*groupid*

is your user group ID.

*userid*

is your user ID.

**Cause.** This command can only be issued by a person logged on under the super ID (user ID 255,255).

**Effect.** The command fails.

**Recovery.** Ask someone with the super ID to enter this command on your behalf, or enter another command.

## 255

A user ID in the SUPER group (255,\*) must be used to execute this command. The user ID of this user is *groupid,userid*.

*groupid*

is your user group ID.

*userid*

is your user ID.

**Cause.** This command can be issued only by a member of the super user group (user ID 255,*n*).

**Effect.** The command fails.

**Recovery.** Ask a member of the super user group to enter this command on your behalf, or select another command.

## 256

Only the owner of the TMF configuration files (user ID *groupid1*, *userid1*), or someone with the SUPER.SUPER ID (user ID 255,255), can execute this command. The user ID of this user is *groupid2*,*userid2*.

*groupid1*

is the user group ID of the owner of the TMF configuration files.

*userid1*

is the user ID of the owner of the TMF configuration files.

*groupid2*

is your user group ID.

*userid2*

is your user ID.

**Cause.** This command can only be issued by the owner of the TMF configuration files, indicated by *groupid1*, *userid1*, or by a person logged on under the super ID (user ID 255,255).

**Effect.** The command fails.

**Recovery.** Ask the owner of the configuration files or the user with the super ID to enter this command on your behalf, or select another command.

## 257

Only someone with the user ID *groupid1,userid1*, or someone with the SUPER.SUPER ID (user ID 255,255), can execute this command. The user ID of this user is *groupid2,userid2*.

*groupid1*

is the user group ID required to execute the command.

*userid1*

is the user ID required to execute the command.

*groupid2*

is your user group ID.

*userid2*

is your user ID.

**Cause.** This command can only be issued by a person logged on under the user ID *groupid1,userid1* or the super ID (255,255).

**Effect.** The command fails.

**Recovery.** Ask the person with the user ID *groupid1,userid1* or the super ID to enter this command on your behalf, or select another command.



## 258

Subsystem Error: Unable to PROCESS\_CREATE\_ the process *process-name* using *filename* because of *errcode*.

*process-name*

is the name of the process that could not be created.

*filename*

is the name of the object file for this process.

*errcode*

supplies a numeric code that further defines the error and identifies the subsystem or system procedure that transmitted this code: TMF, the File System, the NEWPROCESS process-control procedure, or the Enform query language and report writer.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the information supplied in *process-name*, *filename*, and *errcode*. Then, contact the GCSC or your service provider.

## 259

Subsystem Error: Unable to OPEN the TMF process *process-name* (*filename*) because of *errcode*.

*process-name*

is the name of the process that could not be opened.

*filename*

is the name of the object file for this process.

*errcode*

supplies a numeric code that further defines the error and identifies the subsystem or system procedure that transmitted this code: TMF, the File System, the NEWPROCESS process-control procedure, or the Enform query language and report writer.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the information supplied in *process-name*, *filename*, and *errcode*. Then, contact the GCSC or your service provider.

## 260

Subsystem Error: Unable to WRITEREAD a startup message to TMF process *process-name* (*filename*) because of *errcode*.

*process-name*

is the name of the process that could not receive the message.

*filename*

is the name of the object file for this process.

*errcode*

supplies a numeric code that further defines the error and identifies the subsystem or system procedure that transmitted this code: TMF, the File System, the NEWPROCESS process-control procedure, or the Enform query language and report writer.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the information supplied in *process-name*, *filename*, and *errcode*. Then, contact the GCSC or your service provider.

## 261

Subsystem Error: Unable to create a process name for a TMF process.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 263

The TAPEMEDIA named "*tape-volume*" already exists in the catalog.

*tape-volume*

is the name of the existing tape volume.

**Cause.** This command attempted to add a tape volume named *tape-volume* to the TMF catalog, but a tape volume with this identity is already defined there.

**Effect.** The command fails.

**Recovery.** None required.

## 264

The catalog contains no TAPEMEDIA matching the mediaset *tape-volume*.

*tape-volume*

is the name of the nonexistent tape volume.

**Cause.** This command attempted to modify or delete a tape volume named *tape-volume* in the TMF catalog, but a tape volume with this identity is not defined there.

**Effect.** The command fails.

**Recovery.** Re-enter the command, using the name of an existing tape volume.

## 265

The command specified a TAPEMEDIA named *tape-volume*, but this is not the MEDIUM type required.

*tape-volume*

is the name of the referenced tape volume.

**Cause.** This command attempted to modify or delete a tape volume named *tape-volume*, but *tape-volume* erroneously refers to a disk device.

**Effect.** The command fails.

**Recovery.** Re-enter the command, using a proper tape-volume name.

## 266

The command could not label the tape because MEDIASRV returned error *errcode*. See the *DSM/Tape Catalog Messages Manual* for more information about this error.

### *errcode*

supplies a numeric code that further defines the error and identifies the subsystem or system procedure that transmitted this code: TMF, the File System, the NEWPROCESS process-control procedure, or the Enform query language and report writer.

**Cause.** The tape-handling process could not write a label to the tape.

**Effect.** The tape remains unlabeled.

**Recovery.** See the *DSM/Tape Catalog Messages Manual* for more information about correcting this error.

## 268

The command could not label the tape because of error *errcode* on communicating with MEDIASRV. If this error persists, contact the Global Customer Support Center (GCSC) or your service provider.

### *errcode*

supplies a numeric code that further defines the error and identifies the subsystem or system procedure that transmitted this code: TMF, the File System, the NEWPROCESS process-control procedure, or the Enform query language and report writer.

**Cause.** The tape-handling process could not write a label to the tape.

**Effect.** The tape remains unlabeled.

**Recovery.** This message should never appear. Make a note of the command entered and the information supplied in *errcode*. Then, contact the GCSC or your service provider.

## 269

Labeled-tape support is not configured. One or more audit trails are configured for automatic audit dumping, so labeled-tape support must be enabled for this system. Configuration is performed through system generation.

**Cause.** Audit dumping to tape cannot take place because labeled-tape support is not enabled.

**Effect.** Audit dumping to tape remains disabled.

**Recovery.** Ask your system manager to configure labeled-tape support, using the system generation program.

## 270

The TMF configuration files were last initialized under a TMF release that is not upwardly compatible with this release. TMF must be reinitialized to run on this release.

**Cause.** The configuration files were last initialized under a release of TMF that is not compatible with this one.

**Effect.** TMF is inoperable.

**Recovery.** Ask your system manager to reinitialize your TMF system configuration to run on this TMF release.

## 272

The RESOLVE TRANSACTION command failed because of *errcode*.

*errcode*

supplies a numeric code that further defines the error and identifies the subsystem or system procedure that transmitted this code: TMF, the File System, the NEWPROCESS process-control procedure, or the Enform query language and report writer.

**Cause.** While a RESOLVE TRANSACTION command was in process, the condition indicated by *errcode* occurred.

**Effect.** The command fails.

**Recovery.** Refer to the following manuals for a description of the error code:

| Transmitting Subsystem | Manual                              | Appendix Containing Description |
|------------------------|-------------------------------------|---------------------------------|
| TMF                    | This reference manual               | This appendix                   |
| File system            | <i>Operator Messages Manual</i>     | "File-System Errors" Appendix   |
| NEWPROCESS procedure   | <i>Operator Messages Manual</i>     | "NEWPROCESS Errors" Appendix    |
| Enform subsystem       | <i>Enform Plus Reference Manual</i> | "Error Messages" Appendix       |

For further details about understanding and correcting File System and NEWPROCESS errors, see the *Guardian Procedure Errors and Messages Manual*.

Correct the error condition reported, if necessary, and re-enter the command.

## 275

The command attempted to add to an existing dump in the catalog, but the *dump type* (ONLINEDUMP or AUDITDUMP) specified did not match that of the existing dump.

**Cause.** An ADD DUMPS command attempted to add a record for an existing dump in the TMF catalog, but the dump type specified (ONLINEDUMP or AUDITDUMP) did not match that denoted by the existing dump's serial number.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying the correct dump type.

## 276

The command attempted to add an existing audit dump to the TMF catalog.

**Cause.** An ADD DUMPS command attempted to add to the TMF catalog an audit dump record that was already entered there.

**Effect.** None.

**Recovery.** None required.

## 277

The command attempted to add an existing dump to the catalog, but the MAT specified did not match that of the existing dump.

**Cause.** An ADD DUMPS command attempted to add to the TMF catalog a dump whose audit-trail file name did not match that noted in the command.

**Effect.** The command fails.

**Recovery.** Verify the audit-trail file name and re-enter the command.

## 278

The "*part:copy*" specification of the TAPEMEDIA *tape-volume* duplicates a portion of the dump already specified.

*part*

identifies the portion of the dump stored on this tape volume.

*copy*

identifies the copy number of the dump stored on this tape volume.

*tape-volume*

is the tape-volume name.

**Cause.** The command specified, in the *part:copy* subparameter, a portion of the dump already specified in this command.

**Effect.** The command fails.

**Recovery.** Re-enter the command with this portion of the dump specified only once.



## 279

The "*part:copy*" specification of the TAPEMEDIA *tape-volume* is out of sequence with other portions of the dump.

*part*

identifies the portion of the dump stored on this tape volume.

*copy*

identifies the copy number of the dump stored on this tape volume.

*tape-volume*

is the tape-volume name.

**Cause.** The command specified, in the *part:copy* subparameter, a portion of the dump that is out of sequence with the other portions.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying the dump portion in proper sequence.

## 280

The command attempted to open the Process Configuration File, but that file has not yet been created. This file will be created automatically once the TMP starts.

**Cause.** The command attempted to open the Process Configuration File before that file was created.

**Effect.** The command fails.

**Recovery.** Once the TMP process starts, the Process Configuration File is created and you can re-enter the command.

## 281

The TAPEMEDIA *tape-volume* is already recorded as part of another dump in the catalog.

*tape-volume*

is the tape-volume name.

**Cause.** The command attempted to add a tape volume already defined as part of another dump in the TMF catalog.

**Effect.** The command fails.

**Recovery.** None required.

## 282

The file *filename* is already recorded as part of this dump.

*filename*

is the name of the file.

**Cause.** The command attempted to add a file already defined as part of this dump in the TMF catalog.

**Effect.** The command fails.

**Recovery.** None required.

## 283

Subsystem Error: The volume that the file *filename* resides on is not configured as a data volume, or the virtual volume that the file resides on contains one or more physical volumes that are not enabled for TMF.

*filename*

is the name of the referenced file.

**Cause.** For a file on a data volume, a TMF error occurred. For a file on a virtual volume, a required physical volume within the NonStop Storage Management Foundation (SMF) storage pool was not enabled for TMF when an ADD DUMPS command attempted to add the file to the TMF catalog.

**Effect.** The command fails.

**Recovery.** For a file on a data volume, this message should never appear. Make a note of the command entered and the information supplied in *filename*. Then, contact the GCSC or your service provider. For a file on a virtual volume, enable the physical volumes; if you cannot enable the volumes, isolate or quarantine the physical volumes in SMF, so that they are unavailable for file placement. Then, re-enter the ADD DUMPS command.

## 285

No files listed in the TMF catalog match the file set *filename*.

*filename*

identifies the file set.

**Cause.** An INFO DUMPS command either referred to the wrong file set or to a file set for which no dumps have been performed.

**Effect.** The command fails.

**Recovery.** Either re-enter the command, specifying a different file set, or perform dumps to populate the file set.

## 290

The Guardian procedure ENFORMSTART returned an error. The error was *errcode*.

*errcode*

supplies a numeric code that further defines the error and identifies the subsystem and system procedure that transmitted this code: the Guardian procedure ENFORMSTART.

**Cause.** While a command was in process, the condition indicated by *errcode* occurred.

**Effect.** The command fails.

**Recovery.** For details about this error, refer to the *Guardian Procedure Errors and Messages Manual*. Correct the error condition reported, if necessary, and re-enter the command.

## 291

The Guardian procedure ENFORMRECEIVE returned an error. The error was *errcode*.

*errcode*

supplies a numeric code that further defines the error and identifies the subsystem and system procedure that transmitted this code: the Guardian procedure ENFORMRECEIVE.

**Cause.** While a command was in process, the condition indicated by *errcode* occurred.

**Effect.** The command fails.

**Recovery.** For details about this error, refer to the *Guardian Procedure Errors and Messages Manual*. Correct the error condition reported, if necessary, and re-enter the command.

## 295

Unable to open the catalog because of *errcode*.

*errcode*

supplies a numeric code that further defines the error and identifies the subsystem or system procedure that transmitted this code: TMF, the File System, the NEWPROCESS process-control procedure, or the Enform query language and report writer.

**Cause.** While the command attempted to open the TMF catalog, the condition indicated by *errcode* occurred.

**Effect.** The command fails.

**Recovery.** Refer to the following manuals for a description of the error code:

| Transmitting Subsystem | Manual                              | Appendix Containing Description |
|------------------------|-------------------------------------|---------------------------------|
| TMF                    | This reference manual               | This appendix                   |
| File system            | <i>Operator Messages Manual</i>     | "File-System Errors" Appendix   |
| NEWPROCESS procedure   | <i>Operator Messages Manual</i>     | "NEWPROCESS Errors" Appendix    |
| Enform subsystem       | <i>Enform Plus Reference Manual</i> | "Error Messages" Appendix       |

For further details about understanding and correcting File System and NEWPROCESS errors, see the *Guardian Procedure Errors and Messages Manual*.

Correct the error condition reported, if necessary, and re-enter the command.

## 296

Unable to determine catalog status because of *errcode*.

*errcode*

supplies a numeric code that further defines the error and identifies the subsystem or system procedure that transmitted this code: TMF, the File System, the NEWPROCESS process-control procedure, or the Enform query language and report writer.

**Cause.** While the command attempted to obtain the status of the TMF catalog, the condition indicated by *errcode* occurred.

**Effect.** The command fails.

**Recovery.** Refer to the following manuals for a description of the error code:

| Transmitting Subsystem | Manual                              | Appendix Containing Description |
|------------------------|-------------------------------------|---------------------------------|
| TMF                    | This reference manual               | This appendix                   |
| File system            | <i>Operator Messages Manual</i>     | "File-System Errors" Appendix   |
| NEWPROCESS procedure   | <i>Operator Messages Manual</i>     | "NEWPROCESS Errors" Appendix    |
| Enform subsystem       | <i>Enform Plus Reference Manual</i> | "Error Messages" Appendix       |

For further details about understanding and correcting File System and NEWPROCESS errors, see the *Guardian Procedure Calls Reference Manual*.

Correct the error condition reported, if necessary, and re-enter the command.

## 297

Unable to obtain scratch reels because of *errcode*.

*errcode*

supplies a numeric code that further defines the error and identifies the subsystem or system procedure that transmitted this code: TMF, the File System, the NEWPROCESS process-control procedure, or the Enform query language and report writer.

**Cause.** While the command attempted to obtain scratch reels for dumps, the condition indicated by *errcode* occurred.

**Effect.** The command fails.

**Recovery.** Refer to the following manuals for a description of the error code:

| Transmitting Subsystem | Manual                              | Appendix Containing Description |
|------------------------|-------------------------------------|---------------------------------|
| TMF                    | This reference manual               | This appendix                   |
| File system            | <i>Operator Messages Manual</i>     | "File-System Errors" Appendix   |
| NEWPROCESS procedure   | <i>Operator Messages Manual</i>     | "NEWPROCESS Errors" Appendix    |
| Enform subsystem       | <i>Enform Plus Reference Manual</i> | "Error Messages" Appendix       |

For further details about understanding and correcting File System and NEWPROCESS errors, see the *Guardian Procedure Errors and Messages Manual*.

Correct the error condition reported, if necessary, and re-enter the command.

## 298

Unable to start the specified dump because of *errcode*.

*errcode*

supplies a numeric code that further defines the error and identifies the subsystem or system procedure that transmitted this code: TMF, the File System, the NEWPROCESS process-control procedure, or the Enform query language and report writer.

**Cause.** While the command attempted to start a dump, the condition indicated by *errcode* occurred.

**Effect.** The command fails.

**Recovery.** Refer to the following manuals for a description of the error code:

| Transmitting Subsystem | Manual                              | Appendix Containing Description |
|------------------------|-------------------------------------|---------------------------------|
| TMF                    | This reference manual               | This appendix                   |
| File system            | <i>Operator Messages Manual</i>     | "File-System Errors" Appendix   |
| NEWPROCESS procedure   | <i>Operator Messages Manual</i>     | "NEWPROCESS Errors" Appendix    |
| Enform subsystem       | <i>Enform Plus Reference Manual</i> | "Error Messages" Appendix       |

For further details about understanding and correcting File System and NEWPROCESS errors, see the *Guardian Procedure Errors and Messages Manual*.

Correct the error condition reported, if necessary, and re-enter the command.



## 299

The DELETE CATALOG command returned *errcode*. Please see the log for possible further explanation.

*errcode*

supplies a numeric code that further defines the error and identifies the subsystem or system procedure that transmitted this code: TMF, the File System, the NEWPROCESS process-control procedure, or the Enform query language and report writer.

**Cause.** While a DELETE CATALOG command attempted to clear the TMF catalog, the condition indicated by *errcode* occurred.

**Effect.** The command fails.

**Recovery.** Refer to the EMS event log and the following manuals for a description of the error code:

| Transmitting Subsystem | Manual                              | Appendix Containing Description |
|------------------------|-------------------------------------|---------------------------------|
| TMF                    | This reference manual               | This appendix                   |
| File system            | <i>Operator Messages Manual</i>     | "File-System Errors" Appendix   |
| NEWPROCESS procedure   | <i>Operator Messages Manual</i>     | "NEWPROCESS Errors" Appendix    |
| Enform subsystem       | <i>Enform Plus Reference Manual</i> | "Error Messages" Appendix       |

For further details about understanding and correcting File System and NEWPROCESS errors, see the *Guardian Procedure Errors and Messages Manual*.

Correct the error condition reported, if necessary, and re-enter the command.

## 300

Unable to acquire the specified TAPEMEDIA, *tape-volume*, because of *errcode*.

*tape-volume*

is the tape volume that could not be accessed.

*errcode*

supplies a numeric code that further defines the error and identifies the subsystem or system procedure that transmitted this code: TMF, the File System, the NEWPROCESS process-control procedure, or the Enform query language and report writer.

**Cause.** While the command attempted to access a tape volume, the condition indicated by *errcode* occurred.

**Effect.** The command fails.

**Recovery.** Refer to the following manuals for a description of the error code:

| Transmitting Subsystem | Manual                              | Appendix Containing Description |
|------------------------|-------------------------------------|---------------------------------|
| TMF                    | This reference manual               | This appendix                   |
| File system            | <i>Operator Messages Manual</i>     | "File-System Errors" Appendix   |
| NEWPROCESS procedure   | <i>Operator Messages Manual</i>     | "NEWPROCESS Errors" Appendix    |
| Enform subsystem       | <i>Enform Plus Reference Manual</i> | "Error Messages" Appendix       |

For further details about understanding and correcting File System and NEWPROCESS errors, see the *Guardian Procedure Errors and Messages Manual*.

Correct the error condition reported, if necessary, and re-enter the command.

## 301

Unable to end the specified dump because of *errcode*.

*errcode*

supplies a numeric code that further defines the error and identifies the subsystem or system procedure that transmitted this code: TMF, the File System, the NEWPROCESS process-control procedure, or the Enform query language and report writer.

**Cause.** While the command attempted to terminate a dump in progress, the condition indicated by *errcode* occurred.

**Effect.** The command fails.

**Recovery.** Refer to the following manuals for a description of the error code:

| Transmitting Subsystem | Manual                              | Appendix Containing Description |
|------------------------|-------------------------------------|---------------------------------|
| TMF                    | This reference manual               | This appendix                   |
| File system            | <i>Operator Messages Manual</i>     | "File-System Errors" Appendix   |
| NEWPROCESS procedure   | <i>Operator Messages Manual</i>     | "NEWPROCESS Errors" Appendix    |
| Enform subsystem       | <i>Enform Plus Reference Manual</i> | "Error Messages" Appendix       |

For further details about understanding and correcting File System and NEWPROCESS errors, see the *Guardian Procedure Errors and Messages Manual*.

Correct the error condition reported, if necessary, and re-enter the command.

## 302

Unable to add the specified TAPEMEDIA, *tape-volume*, because of *errcode*.

*tape-volume*

is the tape volume that could not be added.

*errcode*

supplies a numeric code that further defines the error and identifies the subsystem or system procedure that transmitted this code: TMF, the File System, the NEWPROCESS process-control procedure, or the Enform query language and report writer.

**Cause.** While the command attempted to add a tape volume to the TMF catalog, the condition indicated by *errcode* occurred.

**Effect.** The command fails.

**Recovery.** Refer to the following manuals for a description of the error code:

| Transmitting Subsystem | Manual                              | Appendix Containing Description |
|------------------------|-------------------------------------|---------------------------------|
| TMF                    | This reference manual               | This appendix                   |
| File system            | <i>Operator Messages Manual</i>     | "File-System Errors" Appendix   |
| NEWPROCESS procedure   | <i>Operator Messages Manual</i>     | "NEWPROCESS Errors" Appendix    |
| Enform subsystem       | <i>Enform Plus Reference Manual</i> | "Error Messages" Appendix       |

For further details about understanding and correcting File System and NEWPROCESS errors, see the *Guardian Procedure Errors and Messages Manual*.

Correct the error condition reported, if necessary, and re-enter the command.

## 303

Unable to add the specified file, *filename*, because of *errcode*.

*filename*

is the file that could not be added.

*errcode*

supplies a numeric code that further defines the error and identifies the subsystem or system procedure that transmitted this code: TMF, the File System, the NEWPROCESS process-control procedure, or the Enform query language and report writer.

**Cause.** While the command attempted to add a file to the TMF system, the condition indicated by *errcode* occurred.

**Effect.** The command fails.

**Recovery.** Refer to the following manuals for a description of the error code:

| Transmitting Subsystem | Manual                              | Appendix Containing Description |
|------------------------|-------------------------------------|---------------------------------|
| TMF                    | This reference manual               | This appendix                   |
| File system            | <i>Operator Messages Manual</i>     | "File-System Errors" Appendix   |
| NEWPROCESS procedure   | <i>Operator Messages Manual</i>     | "NEWPROCESS Errors" Appendix    |
| Enform subsystem       | <i>Enform Plus Reference Manual</i> | "Error Messages" Appendix       |

For further details about understanding and correcting File System and NEWPROCESS errors, see the *Guardian Procedure Errors and Messages Manual*.

Correct the error condition reported, if necessary, and re-enter the command.

## 304

Unable to find the specified TAPEMEDIA, *tape-volume*, because of *errcode*.

*tape-volume*

is the tape volume that could not be located.

*errcode*

supplies a numeric code that further defines the error and identifies the subsystem or system procedure that transmitted this code: TMF, the File System, the NEWPROCESS process-control procedure, or the Enform query language and report writer.

**Cause.** While the command attempted to locate a tape volume in the TMF catalog, the condition indicated by *errcode* occurred.

**Effect.** The command fails.

**Recovery.** Refer to the following manuals for a description of the error code:

| Transmitting Subsystem | Manual                              | Appendix Containing Description |
|------------------------|-------------------------------------|---------------------------------|
| TMF                    | This reference manual               | This appendix                   |
| File system            | <i>Operator Messages Manual</i>     | "File-System Errors" Appendix   |
| NEWPROCESS procedure   | <i>Operator Messages Manual</i>     | "NEWPROCESS Errors" Appendix    |
| Enform subsystem       | <i>Enform Plus Reference Manual</i> | "Error Messages" Appendix       |

For further details about understanding and correcting File System and NEWPROCESS errors, see the *Guardian Procedure Errors and Messages Manual*.

Correct the error condition reported, if necessary, and re-enter the command.

## 305

Unable to find the specified file, *filename*, because of *errcode*.

*filename*

is the file that could not be located.

*errcode*

supplies a numeric code that further defines the error and identifies the subsystem or system procedure that transmitted this code: TMF, the File System, the NEWPROCESS process-control procedure, or the Enform query language and report writer.

**Cause.** While the command attempted to locate a file in the TMF system, the condition indicated by *errcode* occurred.

**Effect.** The command fails.

**Recovery.** Refer to the following manuals for a description of the error code:

| Transmitting Subsystem | Manual                              | Appendix Containing Description |
|------------------------|-------------------------------------|---------------------------------|
| TMF                    | This reference manual               | This appendix                   |
| File system            | <i>Operator Messages Manual</i>     | "File-System Errors" Appendix   |
| NEWPROCESS procedure   | <i>Operator Messages Manual</i>     | "NEWPROCESS Errors" Appendix    |
| Enform subsystem       | <i>Enform Plus Reference Manual</i> | "Error Messages" Appendix       |

For further details about understanding and correcting File System and NEWPROCESS errors, see the *Guardian Procedure Errors and Messages Manual*.

Correct the error condition reported, if necessary, and re-enter the command.

## 306

Unable to find the specified dump because of *errcode*.

*errcode*

supplies a numeric code that further defines the error and identifies the subsystem or system procedure that transmitted this code: TMF, the File System, the NEWPROCESS process-control procedure, or the Enform query language and report writer.

**Cause.** While the command attempted to locate a dump, the condition indicated by *errcode* occurred.

**Effect.** The command fails.

**Recovery.** Refer to the following manuals for a description of the error code:

| Transmitting Subsystem | Manual                              | Appendix Containing Description |
|------------------------|-------------------------------------|---------------------------------|
| TMF                    | This reference manual               | This appendix                   |
| File system            | <i>Operator Messages Manual</i>     | "File-System Errors" Appendix   |
| NEWPROCESS procedure   | <i>Operator Messages Manual</i>     | "NEWPROCESS Errors" Appendix    |
| Enform subsystem       | <i>Enform Plus Reference Manual</i> | "Error Messages" Appendix       |

For further details about understanding and correcting File System and NEWPROCESS errors, see the *Guardian Procedure Errors and Messages Manual*.

Correct the error condition reported, if necessary, and re-enter the command.



## 307

Unable to alter the specified TAPEMEDIA, *tape-volume*, because of *errcode*.

*tape-volume*

is the tape volume that could not be altered.

*errcode*

supplies a numeric code that further defines the error and identifies the subsystem or system procedure that transmitted this code: TMF, the File System, the NEWPROCESS process-control procedure, or the Enform query language and report writer.

**Cause.** While the command attempted to change the entries for a tape volume in the TMF catalog, the condition indicated by *errcode* occurred.

**Effect.** The command fails.

**Recovery.** Refer to the following manuals for a description of the error code:

| Transmitting Subsystem | Manual                              | Appendix Containing Description |
|------------------------|-------------------------------------|---------------------------------|
| TMF                    | This reference manual               | This appendix                   |
| File system            | <i>Operator Messages Manual</i>     | "File-System Errors" Appendix   |
| NEWPROCESS procedure   | <i>Operator Messages Manual</i>     | "NEWPROCESS Errors" Appendix    |
| Enform subsystem       | <i>Enform Plus Reference Manual</i> | "Error Messages" Appendix       |

For further details about understanding and correcting File System and NEWPROCESS errors, see the *Guardian Procedure Errors and Messages Manual*.

Correct the error condition reported, if necessary, and re-enter the command.

## 308

Unable to delete the specified TAPEMEDIA, *tape-volume*, because of *errcode*.

*tape-volume*

is the tape volume that could not be deleted.

*errcode*

supplies a numeric code that further defines the error and identifies the subsystem or system procedure that transmitted this code: TMF, the File System, the NEWPROCESS process-control procedure, or the Enform query language and report writer.

**Cause.** While the command attempted to remove a tape volume from the TMF catalog, the condition indicated by *errcode* occurred.

**Effect.** The command fails.

**Recovery.** Refer to the following manuals for a description of the error code:

| Transmitting Subsystem | Manual                              | Appendix Containing Description |
|------------------------|-------------------------------------|---------------------------------|
| TMF                    | This reference manual               | This appendix                   |
| File system            | <i>Operator Messages Manual</i>     | "File-System Errors" Appendix   |
| NEWPROCESS procedure   | <i>Operator Messages Manual</i>     | "NEWPROCESS Errors" Appendix    |
| Enform subsystem       | <i>Enform Plus Reference Manual</i> | "Error Messages" Appendix       |

For further details about understanding and correcting File System and NEWPROCESS errors, see the *Guardian Procedure Errors and Messages Manual*.

Correct the error condition reported, if necessary, and re-enter the command.

## 309

Unable to count the number of copies of the specified file, *filename*, because of *errcode*.

*filename*

is the file for which copies could not be counted.

*errcode*

supplies a numeric code that further defines the error and identifies the subsystem or system procedure that transmitted this code: TMF, the File System, the NEWPROCESS process-control procedure, or the Enform query language and report writer.

**Cause.** While the command attempted to dump multiple copies of a file, the condition indicated by *errcode* occurred.

**Effect.** The command fails.

**Recovery.** Refer to the following manuals for a description of the error code:

| Transmitting Subsystem | Manual                              | Appendix Containing Description |
|------------------------|-------------------------------------|---------------------------------|
| TMF                    | This reference manual               | This appendix                   |
| File system            | <i>Operator Messages Manual</i>     | "File-System Errors" Appendix   |
| NEWPROCESS procedure   | <i>Operator Messages Manual</i>     | "NEWPROCESS Errors" Appendix    |
| Enform subsystem       | <i>Enform Plus Reference Manual</i> | "Error Messages" Appendix       |

For further details about understanding and correcting File System and NEWPROCESS errors, see the *Guardian Procedure Errors and Messages Manual*.

Correct the error condition reported, if necessary, and re-enter the command.

## 310

Unable to alter the specified dump because of *errcode*.

*errcode*

supplies a numeric code that further defines the error and identifies the subsystem or system procedure that transmitted this code: TMF, the File System, the NEWPROCESS process-control procedure, or the Enform query language and report writer.

**Cause.** While the command attempted to alter dump entries in the TMF catalog, the condition indicated by *errcode* occurred.

**Effect.** The command fails.

**Recovery.** Refer to the following manuals for a description of the error code:

| Transmitting Subsystem | Manual                              | Appendix Containing Description |
|------------------------|-------------------------------------|---------------------------------|
| TMF                    | This reference manual               | This appendix                   |
| File system            | <i>Operator Messages Manual</i>     | "File-System Errors" Appendix   |
| NEWPROCESS procedure   | <i>Operator Messages Manual</i>     | "NEWPROCESS Errors" Appendix    |
| Enform subsystem       | <i>Enform Plus Reference Manual</i> | "Error Messages" Appendix       |

For further details about understanding and correcting File System and NEWPROCESS errors, see the *Guardian Procedure Errors and Messages Manual*.

Correct the error condition reported, if necessary, and re-enter the command.

## 311

Unable to delete the specified dump because of *errcode*.

*errcode*

supplies a numeric code that further defines the error and identifies the subsystem or system procedure that transmitted this code: TMF, the File System, the NEWPROCESS process-control procedure, or the Enform query language and report writer.

**Cause.** While the command attempted to remove a dump entry from the TMF catalog, the condition indicated by *errcode* occurred.

**Effect.** The command fails.

**Recovery.** Refer to the following manuals for a description of the error code:

| Transmitting Subsystem | Manual                              | Appendix Containing Description |
|------------------------|-------------------------------------|---------------------------------|
| TMF                    | This reference manual               | This appendix                   |
| File system            | <i>Operator Messages Manual</i>     | "File-System Errors" Appendix   |
| NEWPROCESS procedure   | <i>Operator Messages Manual</i>     | "NEWPROCESS Errors" Appendix    |
| Enform subsystem       | <i>Enform Plus Reference Manual</i> | "Error Messages" Appendix       |

For further details about understanding and correcting File System and NEWPROCESS errors, see the *Guardian Procedure Errors and Messages Manual*.

Correct the error condition reported, if necessary, and re-enter the command.

## 314

Unable to add the specified DISKMEDIA, *filename*, because of *errcode*.

*filename*

is the disk file that could not be added.

*errcode*

supplies a numeric code that further defines the error and identifies the subsystem or system procedure that transmitted this code: TMF, the File System, the NEWPROCESS process-control procedure, or the Enform query language and report writer.

**Cause.** While the command attempted to add a disk file to the TMF configuration, the condition indicated by *errcode* occurred.

**Effect.** The command fails.

**Recovery.** Refer to the following manuals for a description of the error code:

| Transmitting Subsystem | Manual                              | Appendix Containing Description |
|------------------------|-------------------------------------|---------------------------------|
| TMF                    | This reference manual               | This appendix                   |
| File system            | <i>Operator Messages Manual</i>     | "File-System Errors" Appendix   |
| NEWPROCESS procedure   | <i>Operator Messages Manual</i>     | "NEWPROCESS Errors" Appendix    |
| Enform subsystem       | <i>Enform Plus Reference Manual</i> | "Error Messages" Appendix       |

For further details about understanding and correcting File System and NEWPROCESS errors, see the *Guardian Procedure Errors and Messages Manual*.

Correct the error condition reported, if necessary, and re-enter the command.

## 315

The command attempted to add a device for an existing dump in the catalog, but the dump medium (TAPE or DISK) specified did not match that of the existing dump.

**Cause.** The command attempted to add an additional device for a particular dump to the TMF catalog, but the device is not the same type as the devices previously assigned for this dump.

**Effect.** The command fails.

**Recovery.** Re-enter the command, using a device of the same type (either tape or disk) as those devices already assigned for the dump.

## 316

Unable to delete the specified \*.\*.\* because of *errcode*.

*errcode*

supplies a numeric code that further defines the error and identifies the subsystem or system procedure that transmitted this code: TMF, the File System, the NEWPROCESS process-control procedure, or the Enform query language and report writer.

**Cause.** While the command attempted to remove the specified fileset from the TMF configuration, the condition indicated by *errcode* occurred.

**Effect.** The command fails.

**Recovery.** Refer to the following manuals for a description of the error code:

| Transmitting Subsystem | Manual                              | Appendix Containing Description |
|------------------------|-------------------------------------|---------------------------------|
| TMF                    | This reference manual               | This appendix                   |
| File system            | <i>Operator Messages Manual</i>     | "File-System Errors" Appendix   |
| NEWPROCESS procedure   | <i>Operator Messages Manual</i>     | "NEWPROCESS Errors" Appendix    |
| Enform subsystem       | <i>Enform Plus Reference Manual</i> | "Error Messages" Appendix       |

For further details about understanding and correcting File System and NEWPROCESS errors, see the *Guardian Procedure Errors and Messages Manual*.

Correct the error condition reported, if necessary, and re-enter the command.

## 317

The command did not purge some or all of the DISKMEDIA from disk because of *errcode*. See the log for the specific DISKMEDIA. The dump has been deleted from the catalog.

*errcode*

supplies a numeric code that further defines the error and identifies the subsystem or system procedure that transmitted this code: TMF, the File System, the NEWPROCESS process-control procedure, or the Enform query language and report writer.

**Cause.** When the command attempted to delete a dump from the TMF catalog, the dump was successfully removed, but one or more disk devices assigned for the dump could not be removed from the configuration because of the condition indicated by *errcode*.

**Effect.** The command fails.

**Recovery.** Refer to the following manuals for a description of the error code:

| Transmitting Subsystem | Manual                              | Appendix Containing Description |
|------------------------|-------------------------------------|---------------------------------|
| TMF                    | This reference manual               | This appendix                   |
| File system            | <i>Operator Messages Manual</i>     | "File-System Errors" Appendix   |
| NEWPROCESS procedure   | <i>Operator Messages Manual</i>     | "NEWPROCESS Errors" Appendix    |
| Enform subsystem       | <i>Enform Plus Reference Manual</i> | "Error Messages" Appendix       |

See the EMS event log file to discover which devices were not removed. For further details about understanding and correcting File System and NEWPROCESS errors, see the *Guardian Procedure Errors and Messages Manual*.

Correct the error condition reported, if necessary, and re-enter the command.



## 318

The command did not purge some or all of the DISKMEDIA from disk because of *errcode*. See the log for the specific DISKMEDIA. The dump has not been removed from the catalog.

*errcode*

supplies a numeric code that further defines the error and identifies the subsystem or system procedure that transmitted this code: TMF, the File System, the NEWPROCESS process-control procedure, or the Enform query language and report writer.

**Cause.** When the command attempted to delete a dump from the TMF catalog, the dump remained in the catalog. Furthermore, one or more disk devices assigned for the dump could not be removed from the configuration because of the condition indicated by *errcode*.

**Effect.** The command fails.

**Recovery.** Refer to the following manuals for a description of the error code:

| Transmitting Subsystem | Manual                              | Appendix Containing Description |
|------------------------|-------------------------------------|---------------------------------|
| TMF                    | This reference manual               | This appendix                   |
| File system            | <i>Operator Messages Manual</i>     | "File-System Errors" Appendix   |
| NEWPROCESS procedure   | <i>Operator Messages Manual</i>     | "NEWPROCESS Errors" Appendix    |
| Enform subsystem       | <i>Enform Plus Reference Manual</i> | "Error Messages" Appendix       |

See the EMS event log file to discover which devices were not removed. For further details about understanding and correcting File System and NEWPROCESS errors, see the *Guardian Procedure Errors and Messages Manual*.

Correct the error condition reported, if necessary, and re-enter the command.

## 319

The command specified more than one DISKMEDIA for an audit dump.

**Cause.** An ADD DUMPS command identified multiple disks for an audit dump, but only one disk is allowed.

**Effect.** The command fails.

**Recovery.** Re-enter the command with a single DISKMEDIA option.

## 320

The DISKMEDIA *filename* is out of sequence with the other DISKMEDIA listed or with the DISKMEDIA already added into the catalog.

*filename*

is the disk file that is out of sequence.

**Cause.** The command specified a disk volume that was out of sequence with the other disk volumes specified in this command or those already configured in the catalog.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying the disk volume in the proper sequence.

## 323

The specified audit trail already exists.

**Cause.** The command attempted to add an audit trail already defined in the configuration.

**Effect.** None.

**Recovery.** None required.

## 324

Audit trails must be configured in order, from MASTER to AUXILIARY15.

**Cause.** The command attempted to configure an audit trail that would be out of sequence with audit trails already configured. For example, perhaps the command attempted to add AUX01 before MAT or AUX07 before AUX06.

**Effect.** The command fails.

**Recovery.** Issue an INFO AUDITTRAIL \* command to list all audit trails currently configured, and then add the next audit trail in the appropriate sequence.

## 325

This command would result in no active audit trail volumes being configured.

**Cause.** The command attempted to remove all active-audit volumes from the system, but a TMF configuration must have at least one such volume.

**Effect.** The command fails.

**Recovery.** Re-enter the command without specifying all remaining audit volumes; leave at least one such volume in the system.

## 326

This command would result in no restore audit trail volumes being configured when AUDITDUMP is ON.

**Cause.** The command attempted to delete all restore-audit volumes from the system, but a TMF configuration must have at least one such volume.

**Effect.** The command fails.

**Recovery.** Re-enter the command without specifying all remaining restore-audit volumes; leave at least one such volume in the system.

## 327

Volume *volume* is already configured as an active audit volume for another audit trail. A volume can only be active for a single audit trail.

*volume*

is the volume name.

**Cause.** The command attempted to configure as an active-audit volume a volume that has already been configured for another audit trail. However, any audit volume can only be active for one audit trail at a time.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying another audit volume for this audit trail.

## 328

Volume *volume*, which was specified as an active audit volume, already contains files on the ZTMFAT subvolume. This is not allowed.

*volume*

is the volume name.

**Cause.** The command attempted to configure as an active-audit volume a disk volume that already contains files stored in the ZTMFAT subvolume. This subvolume name is reserved for audit files. Volumes with files in the ZTMFAT subvolume cannot be newly added as audit volumes to the configuration.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a different volume.

## 329

File System error *fserrcode* occurred while accessing the AUDINFO file.

*fserrcode*

supplies a numeric code that further defines the error and identifies the subsystem that transmitted this code as the File System.

**Cause.** A File System error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the information supplied in *fserrcode*. Then, contact the GCSC or your service provider.

## 330

File System error *fserrcode* occurred while attempting to add volume *volume* to the audit trail configuration as *at-volume-type*.

*fserrcode*

supplies a numeric code that further defines the error and indicates it was transmitted by the File System.

*volume*

is the name of the volume specified in the command.

*at-volume-type*

is the volume type specified in the command.

**Cause.** While the command attempted to add the volume name *volume* to the TMF configuration as an *at-volume-type*, the condition indicated by *fserrcode* occurred. *at-volume-type* must be ACTIVEVOL, OVERFLOWVOL, or RESTOREVOL, to indicate an active-, overflow-, or restore-audit volume, respectively.

**Effect.** The command fails.

**Recovery.** Refer to the *Operator Messages Manual* for a description of the error code. For additional details about understanding and correcting File System errors, see the *Guardian Procedure Errors and Messages Manual*.

Correct the error condition reported, if necessary, and re-enter the command.

## 331

File System error *fserrcode* occurred while attempting to delete volume *volume* from the audit trail configuration as *at-volume-type*.

*fserrcode*

supplies a numeric code that further defines the error and indicates it was transmitted by the File System.

*volume*

is the name of the volume specified in the command.

*at-volume-type*

is the volume type specified in the command.

**Cause.** While the command attempted to delete the volume named *volume* from the TMF configuration as an *at-volume-type*, the condition indicated by *fserrcode* occurred. *at-volume-type* must be ACTIVEVOL, OVERFLOWVOL, or RESTOREVOL, to indicate an active-, overflow-, or restore-audit volume, respectively.

**Effect.** The command fails.

**Recovery.** Refer to the *Operator Messages Manual* for a description of the error code. For additional details about understanding and correcting File System errors, see the *Guardian Procedure Errors and Messages Manual*.

Correct the error condition reported, if necessary, and re-enter the command.

## 332

Attempt to add volume *volume* failed because an audit trail can only have 16 volumes configured as *at-volume-type*.

*volume*

is the name of the volume specified in the command.

*at-volume-type*

is the volume type specified in the command.

**Cause.** The command attempted to add a 17th volume of type *at-volume-type*, but only up to 16 of this type are permitted. *at-volume-type* must be ACTIVEVOL, OVERFLOWVOL, or RESTOREVOL, to indicate an active-, overflow-, or restore-audit volume, respectively.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a different type of audit volume, or delete one of the present audit volumes of this type and then add another.

## 333

Volume *volume* cannot be added as an active audit volume because it is currently configured as a data volume.

*volume*

is the name of the volume specified in the command.

**Cause.** The command attempted to add a data volume to the configuration as an audit volume, but data volumes cannot be used as audit volumes.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying another volume that is not a data volume.

## 334

Volume *volume* cannot be deleted as *at-volume-type* because it is not configured as that type.

*volume*

is the name of the volume specified in the command.

*at-volume-type*

is the volume type specified in the command.

**Cause.** The command attempted to remove an audit volume of type *at-volume-type*, but the volume is not configured as that type. For example, perhaps the command attempted to delete an ACTIVEVOL named \$VOL1, but \$VOL1 is configured as an OVERFLOWVOL.

**Effect.** The command fails.

**Recovery.** Re-enter the command, making sure that the volume you name is the appropriate type.

## 335

This command cannot add an active volume when data volume service is not started.

**Cause.** An ALTER AUDITTRAIL command was issued with the ADDACTIVEVOL option, but TMF startup has not yet progressed sufficiently to allow processing of the request.

**Effect.** The command fails.

**Recovery.** Wait until the data volume portion of TMF has been started before attempting to add active volumes to the audit trail configuration. Use the STATUS DATAVOLS command to determine if data volume service is started.



## 336

Volume *volume* cannot be deleted as an active audit volume because it holds the CURRENT audit trail file.

*volume*

is the name of the volume specified in the command.

**Cause.** The command attempted to remove the volume that holds the current audit-trail file.

**Effect.** The command fails.

**Recovery.** Re-enter the command, selecting another volume to remove.

## 337

Volume *volume* cannot be deleted as an active audit volume because it holds the NEXT audit trail file.

*volume*

is the name of the volume specified in the command.

**Cause.** The command attempted to remove the volume that contains the file specified as the next file in the audit-trail sequence—the file that will next become the current audit-trail file.

**Effect.** The command fails.

**Recovery.** Re-enter the command, selecting another volume to remove.

## 338

Volume *volume*, being configured as *at-volume-type*, has files residing in the ZTMFAT subvolume that are either old audit trail files or are not audit trail files at all.

*volume*

is the name of the volume specified in the command.

*at-volume-type*

is the volume type specified in the command.

**Cause.** The command attempted to add the volume named *volume* to the configuration as an *at-volume-type* audit volume, but this volume currently contains files stored in the ZTMFAT subvolume. Volumes containing any files at all in their ZTMFAT subvolume cannot be added as audit volumes.

**Effect.** The command fails.

**Recovery.** Remove these files from the volume, or select another volume. Then, re-enter the command.

## 339

The *at-id* audit trail does not exist.

*at-id*

is the audit-trail identifier specified in the command.

**Cause.** The command referred to an audit trail that does not exist.

**Effect.** The command fails.

**Recovery.** Use INFO commands to list all audit trails in your TMF configuration, and select an existing audit trail. Then, re-enter the command, specifying this audit trail.

## 340

An auxiliary audit trail cannot be configured with AUDITDUMP ON unless the master audit trail is also configured with AUDITDUMP ON.

**Cause.** The command attempted to configure an auxiliary audit trail with the AUDITDUMP option set to ON, but the master audit trail has this option set to OFF. If any auxiliary audit trail is to be dumped, the master audit trail must be dumped.

**Effect.** The command fails.

**Recovery.** Reset the AUDITDUMP option to ON for the master audit trail, and then re-enter the command. Or, re-enter the command with the AUDITDUMP option for the auxiliary audit trail set to OFF.

## 341

The number of active audit trail files on volume *volume* cannot be adjusted to the specified FILESPERVOLUME parameter.

*volume*

is the name of the volume specified in the command.

**Cause.** The command specified too many files per volume for an audit trail. The FILESPERVOLUME parameter specifies the number of files in the audit trail that reside on each active-audit volume configured. For this parameter, the minimum number is 1 and the maximum is 100.

**Effect.** The command fails

**Recovery.** Re-enter the command, specifying a value from 5 through 100 for the FILESPERVOLUME parameter.

## 342

Before issuing this command, you must first delete the current TMF configuration.

**Cause.** The command attempted to modify the TMF configuration, but the configuration must be totally removed from the system for this command to be executed.

**Effect.** The command fails.

**Recovery.** If you are ready to delete the TMF configuration, do so (or ask your system manager to do so). Then, re-enter the command.

## 343

You cannot alter the active audit volume configuration for an audit trail unless TMF is started.

**Cause.** The command attempted to change the active volume configuration for an audit trail, but TMF is not yet started.

**Effect.** The command fails.

**Recovery.** Start TMF (or ask your system manager to do so). Then, re-enter the command.

## 346

The resource manager named *rmname* already exists.

*rmname*

is the duplicate resource manager name specified in the command.

**Cause.** The command attempted to add the resource manager specified in *rmname*, but this name already exists in the resource manager directory.

**Effect.** The command fails for this resource manager name.

**Recovery.** Retry the command with a different resource manager name.

## 347

The resource manager named *rmname* does not exist.

*rmname*

is the name of the non-existent resource manager specified in the command.

**Cause.** The command attempted to delete the resource manager specified in *rmname*, but the name does not exist in the resource manager directory.

**Effect.** The command fails for this resource manager name.

**Recovery.** Retry the command with a different resource manager name.

## 348

The resource manager named *rmname* is currently open.

*rmname*

is the name of the open resource manager specified in the command.

**Cause.** The command attempted to delete the resource manager specified in *rmname*, but this resource manager is open at present.

**Effect.** The command fails for this resource manager name.

**Recovery.** Force closure of the resource manager by entering the CLOSE RESOURCEMANAGER command or by causing the gateway process to close the resource manager. Then, retry the command.

## 349

The resource manager named *rmname* has unresolved transactions.

*rmname*

is the name of the resource manager with unresolved transactions that was specified in the command.

**Cause.** The command attempted to delete the resource manager specified in *rmname*, but this resource manager has unresolved transactions pending.

**Effect.** The command fails for this resource manager name.

**Recovery.** The transactions must be resolved before the resource manager can be deleted. This should be done by invoking a gateway process to open the resource manager and resolve the transactions with the relevant foreign resource manager. If this is not possible, issue the RESOLVE TRANSACTION command to force resolution of the transactions. Then, retry the DELETE RESOURCEMANAGER command.

## 350

The resource manager named *rmname* is not currently open.

*rmname*

is the name of the unopen resource manager specified in the command.

**Cause.** The command attempted to close the resource manager specified in *rmname*, but this resource manager is not currently open.

**Effect.** The command fails for this resource manager name.

**Recovery.** Retry the command with another resource manager name, or select another TMFCOM command.

## 351

The resource manager named *rmname* does not match the specified criteria.

*rmname*

is the unmatched resource manager name specified in the command.

**Cause.** A STATUS RESOURCEMANAGER command specified a resource manager name that does not match the filtering criteria supplied in the STATE or PROCESS parameters.

**Effect.** The command fails for this resource manager name.

**Recovery.** Check the filtering criteria supplied in the command against the resource manager name. Retry the command with another resource manager name or different filtering criteria.

## 352

You cannot close a volatile resource manager through TMFCOM.

**Cause.** You tried to close a volatile resource manager through a TMFCOM command.

**Effect.** The command fails.

**Recovery.** To close an open volatile resource manager from your workstation or terminal, first issue a STATUS RESOURCEMANAGER command to determine what process has opened the volatile resource manager. Then, issue an operating system command to stop that process.

## 353

The ALTER BEGINTRANS, AUTOABORT *timer* command failed because the autoabort timer value cannot be less than the disconnect timer value. Use the INFODISCONNECTTMR command in SNOOP to determine the disconnect timer value

*timer*

is the timer value specified in the ALTER BEGINTRANS command.

**Cause.** The ALTER BEGINTRANS command attempted to set the autoabort timer to a value below the disconnect timer value, after the disconnect timer value was changed from its default setting.

**Effect.** The command fails; no values are changed.

**Recovery.** Use the INFODISCONNECTTMR command in SNOOP to determine the current value of the disconnect timer. Then, either use TMFCOM to re-enter the ALTER BEGINTRANS command with an AUTOABORT value greater than or equal to the current disconnect timer value, or use SNOOP to lower the disconnect timer value and then use TMFCOM to re-enter the ALTER BEGINTRANS command with the same AUTOABORT value entered previously.

## 354

File System Error: TMFSERVE could not communicate with the TMP because of File System error *fserrcode*.

*fserrcode*

supplies a numeric code that further defines the error and identifies the subsystem that transmitted this code as the File System.

**Cause.** A File System error occurred.

**Effect.** The command fails.

**Recovery.** Make a note of the command entered and the information supplied in *fserrcode*, and check the state of TMF. If TMF is still operational, re-enter the command. If TMF has crashed, however, contact the GCSC or your service provider.

## 355

Audit trails cannot be added after TMF has been started following a DELETE TMF operation.

**Cause.** An ADD AUDITTRAIL command attempted to add an audit trail after a new TMF configuration had been generated. New audit trails, however, can only be added after the DELETE TMF command purges the old configuration but before the START TMF command generates the new one.

**Effect.** The command fails.

**Recovery.** See your system manager.

## 356

Volume *volume* could not be deleted as *at-volume-type*.

*volume*

is the name of the volume specified in the command.

*at-volume-type*

is the audit volume type specified in the command.

**Cause.** An ALTER AUDITTRAIL command attempted to delete a volume specified as an *at-volume-type*, but this volume was not that type. For example, the command specified DELETEACTIVEVOL (indicating an active-audit volume) when it should have specified DELETEDOVERFLOWVOL (indicating an overflow-audit volume).

**Effect.** The command fails.

**Recovery.** Re-enter the command, using the appropriate deletion parameter. This parameter must be set to DELETEACTIVEVOL, DELETEOVERLOWVOL, or DELETERESTOREVOL.



## 357

Volume *volume*, being configured as *at-volume-type*, does not exist.

*volume*

is the name of the volume specified in the command.

*at-volume-type*

is the audit volume type specified in the command.

**Cause.** An ADD AUDITTRAIL or ALTER AUDITTRAIL command attempted to add a volume named *volume* as an *at-volume-type*, but this volume does not exist.

**Effect.** The command fails.

**Recovery.** Re-enter the command, using the name of an existing volume.

## 358

Volume *volume* was specified as *at-volume-type* twice in the same command. This is invalid.

*volume*

is the name of the volume specified in the command.

*at-volume-type*

is the audit volume type specified in the command.

**Cause.** The command attempted to add or delete a volume named *volume*, specified two or more times in this command as audit volume type *at-volume-type*. However, the same volume can be specified as the same type only once in a command. For example, the command specified ADDACTIVEVOL \$AUDIT3 twice.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying the volume as this audit volume type only once.

## 359

AUDITDUMP could not be set to OFF for the master audit trail because auxiliary audit trails were configured with AUDITDUMP ON.

**Cause.** The command attempted to configure a master audit trail with the AUDITDUMP option set to OFF, but this option is set to ON for the auxiliary audit trail (or trails). This option must be consistent for all audit trails.

**Effect.** The command fails.

**Recovery.** Reset the AUDITDUMP option to ON for the master audit trail, and then re-enter the command. Or, re-enter the command with the AUDITDUMP option for the auxiliary audit trail set to OFF.

## 360

There must be at least two active audit trail files, but this command would result in only one. (The number of audit trail files equals FILESPERVOLUME times the number of active audit volumes.)

**Cause.** An ADD AUDITTRAIL or ALTER AUDITTRAIL command would result in defining only one audit-trail file in the TMF system, but at least two are required. To determine the number of audit-trail files specified, multiply the value given in the FILESPERVOLUME parameter of the command by the number of active-audit volumes currently defined.

**Effect.** The command fails.

**Recovery.** Re-enter the command, increasing the value of FILESPERVOLUME. Alternatively, add more active-audit volumes and then re-enter the command with the previous value.

## 361

Before issuing this command, you must first start TMF.

**Cause.** The command attempted to modify the TMF configuration but requires TMF to be running to take effect.

**Effect.** The command fails.

**Recovery.** Enter (or ask your system manager to enter) the START TMF command to activate TMF. Then, re-enter the previous command.

## 362

Subsystem Error: Status ATFiles specified an illegal ATFileStatus value. Contact the Global Customer Support Center (GCSC) or your service provider.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Contact the GCSC or your service provider.

## 363

Subsystem Error: Status ATFiles specified an illegal ATFileDumpStatus value. Contact the Global Customer Support Center (GCSC) or your service provider.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Contact the GCSC or your service provider.

## 364

The *at-id* audit trail is not configured for dumping.

*at-id*

is the audit-trail identifier specified in the command.

**Cause.** The command indicated an audit trail for which audit dumping was not configured. For example, audit dumping must be configured for audit trails referenced by the following commands: ALTER AUDITDUMP, ALTER AUDITTRAIL, INFO AUDITDUMP, and STATUS AUDITDUMP.

**Effect.** The command fails.

**Recovery.** Verify that your command referred to the right audit trail, or that the audit trail is enabled. Then, re-enter the command with the correct audit trail reference.

## 365

The command specified the same ADDDISKMEDIA or DELDISKMEDIA identifier more than once. The duplicated identifier is: *disk*.

*disk*

is the duplicated identifier.

**Cause.** An ALTER AUDITDUMP command included more than one instance of the same disk medium identifier in the ADDDISKMEDIA or DELDISKMEDIA parameter.

**Effect.** The command fails.

**Recovery.** Re-enter the command with the correct audit-trail identifier specified only once.

## 366

The command specified, in the DELDISKMEDIA parameter, a device that is not configured. The name of the device is: *disk*.

*disk*

is the name of the unconfigured disk.

**Cause.** An ALTER AUDITDUMP command indicated, in the DELDISKMEDIA parameter, an unconfigured disk device.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying the name of a configured device.

## 367

The command specified, in the ADDDISKMEDIA parameter, a device that is already configured. The name of the device is: *disk*.

*disk*

is the name of the configured disk.

**Cause.** An ALTER AUDITDUMP command indicated, in the ADDDISKMEDIA parameter, a configured disk device.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying the name of an unconfigured device.

## 370

Subsystem Error: The backup TMP could not handle the request.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 371

This command requires that TMF is either started or in the process of stopping.

**Cause.** The command attempted to modify the TMF configuration but cannot take effect because TMF is neither started nor in the process of stopping.

**Effect.** The command fails.

**Recovery.** Enter (or ask your system manager to enter) the START TMF command to activate TMF. Then, re-enter the previous command.

## 372

AUDITDUMP cannot be turned OFF because at least one DATAVOL is configured to use the audittrail with RECOVERYMODE ARCHIVE; therefore AUDITDUMP must be ON.

**Cause.** The command specified setting the AUDITDUMP parameter to OFF (preventing audit dumping), but this parameter must remain ON because the RECOVERYMODE option is set to ARCHIVE (indicating that audit-trail files are not retained on disk for recovery purposes). AUDITDUMP ON permits audit dumping to take place.

**Effect.** The command fails.

**Recovery.** Re-enter the command with AUDITDUMP set to ON, or change the RECOVERYMODE attribute to ONLINE for all data volumes using this audit trail.

## 380

```
START TMF operation operation-number begun.
```

*operation-number*

is an integer assigned by TMF to uniquely identify this specific operation at your terminal and in the EMS log file.

**Cause.** A START TMF command has been executed, resulting in startup of the TMF system and transaction processing.

**Effect.** None; informative message only.

**Recovery.** None required.

## 381

```
STOP TMF operation operation-number begun.
```

*operation-number*

is an integer assigned by TMF to uniquely identify this specific operation at your terminal and in the EMS log file.

**Cause.** A STOP TMF command has been executed, terminating the TMF system and transaction processing.

**Effect.** None; informative message only.

**Recovery.** None required.

## 382

```
TMF is not stopped.
```

**Cause.** A START TMF command was issued, but TMF is already started.

**Effect.** None; informative message only.

**Recovery.** None required.

## 383

```
TMF is not started.
```

**Cause.** A STOP TMF command was issued, but TMF has not been started.

**Effect.** None; informative message only.

**Recovery.** None required.

## 385

Subsystem Error: A STOP TMF command specified the ABRUPT option with an illegal value. Contact the Global Customer Support Center (GCSC) or your service provider.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Contact the GCSC or your service provider.

## 386

The state of the specified transaction (*transid*) does not permit the attempted operation.

*transid*

is a transaction identifier.

**Cause.** A RESOLVE TRANSACTION, ABORT TRANSACTION, or DELETE TRANSACTION command was issued for the specified transaction, but the transaction is in a state that does not allow the operation requested.

**Effect.** The command fails.

**Recovery.** Reissue the command when the transaction is in an appropriate state.

## 387

Attempt to add resource manager name *rmname* failed because the configured limit on the total number of resource managers has already been reached.

*rmname*

is the resource manager name specified in the command.

**Cause.** An ADD RESOURCEMANAGER command attempted to add a resource manager after the configured limit on the total number of recoverable resource managers was reached.

**Effect.** The command fails.

**Recovery.** Delete one or more recoverable resource managers using the DELETE RESOURCEMANAGER command, or increase the configured recoverable resource manager limit using the ALTER BEGINTRANS, RECRMOUNT command. Then add the resource manager by reissuing the ADD RESOURCEMANAGER command.

## 390

```
volume -- not added (not accessible).
```

*volume*

is the name of the volume specified in the command.

**Cause.** The command attempted to add a volume but could not do so, probably because the volume is inaccessible.

**Effect.** The command fails.

**Recovery.** Make the volume accessible; bring up the volume or the processor to which it is attached.

## 391

```
volume -- not added (DP rejected startup request, see EMS log).
```

*volume*

is the name of the volume specified in the command.

**Cause.** The command attempted to add a volume but could not do so because the request to start the volume was rejected.

**Effect.** The command fails.

**Recovery.** Examine the EMS log file to determine the reason the command failed. Correct this cause, if possible, and re-enter the command.

## 392

```
volume -- not added (belongs to different config; not overridden).
```

*volume*

is the name of the volume specified in the command.

**Cause.** The command attempted to add a data volume but could not do so because this volume was already assigned to a different TMF configuration.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying the IGNOREPREVIOUSCONFIG option to override this TMF check.



### 393

```
volume -- not added (already configured as a data volume).
```

*volume*

is the name of the volume specified in the command.

**Cause.** The command attempted to add a data volume but could not do so because the volume already exists in the current configuration as a data volume.

**Effect.** None.

**Recovery.** None required.

### 394

```
volume -- not added (already configured as active audit trail volume).
```

*volume*

is the name of the volume specified in the command.

**Cause.** The command attempted to add a data volume but could not do so because the volume already exists in the current configuration as an active-audit trail volume.

**Effect.** None.

**Recovery.** None required.

## 395

ARCHIVE recovery mode requires that the specified audit trail be configured for audit dumping.

**Cause.** The command set the RECOVERYMODE option to ARCHIVE, but the audit trail was not configured for audit dumping.

**Effect.** The command fails.

**Recovery.** Possible recovery actions are as follows:

- If you unintentionally set RECOVERYMODE to ARCHIVE, re-enter the ADD DATAVOLS or ALTER DATAVOLS command with RECOVERYMODE set to ONLINE.
- If you do not really need to alter the data volume configuration, leave it as is with the RECOVERYMODE set to ONLINE.
- If the data volume is pinning an audit-trail file and you are willing to give up volume recovery for that volume in return for freeing the audit trail, delete the data volume from the configuration; you can re-add the volume later if you wish.
- If the audit trail should really be configured for audit dumping, you can use the ALTER AUDITTRAIL command to turn audit dumping on. Then, re-enter the ADD DATAVOLS command.

---

△ **Caution.** Turning on audit dumping can have a significant impact on TMF operations and should only be performed with the consent of your system manager. For further information about audit dumping, see the ADD AUDITTRAIL and ALTER AUDITTRAIL command descriptions in [Section 3, TMFCOM Commands](#).

---

## 396

ARCHIVE recovery mode requires that the MASTER audit trail be configured for audit dumping.

**Cause.** The command set the RECOVERYMODE option to ARCHIVE, but the master audit trail was not configured for audit dumping.

**Effect.** The command fails.

**Recovery.** Possible recovery actions are as follows:

- If you unintentionally set RECOVERYMODE to ARCHIVE, re-enter the ADD DATAVOLS or ALTER DATAVOLS command with RECOVERYMODE set to ONLINE.
- If you do not really need to alter the data volume configuration, leave it as is with the RECOVERYMODE set to ONLINE.
- If the data volume is pinning an audit-trail file and you are willing to give up volume recovery for that volume in return for freeing the audit trail, delete the data volume from the configuration; you can re-add the volume later if you wish.
- If the audit trail should really be configured for audit dumping, you can use the ALTER AUDITTRAIL command to turn audit dumping on. Then, re-enter the ADD DATAVOLS command.

---

△ **Caution.** Turning on audit dumping can have a significant impact on TMF operations and should only be performed with the consent of your system manager. For further information about audit dumping, see the ADD AUDITTRAIL and ALTER AUDITTRAIL command descriptions in [Section 3, TMFCOM Commands](#).

---

## 397

*volume* is not a configured data volume.

*volume*

is the name of the volume specified in the command.

**Cause.** The command specified a volume that was not configured as a data volume.

**Effect.** The command fails.

**Recovery.** If necessary, reissue the command, specifying the name of a configured data volume. If the volume you named is not a configured data volume, but it should be, use the ADD DATAVOLS command to add the volume to the configuration.

## 398

*volume* did not match the selection criteria specified in the command.

*volume*

is the name of the volume specified in the command.

**Cause.** The command specified the volume but supplied additional selection criteria that did not match the volume's attributes. For example, a STATUS DATAVOLS command requested information about a single volume named \$DATA10 but specified that only information about volumes in the STARTED state was to be returned. Because \$DATA10 had not been started, the volume and the selection criteria did not match.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying multiple volumes with the selection criteria, or specifying a single volume but omitting the selection criteria.

## 399

*volume* is not in a valid state for this command.

*volume*

is the name of the volume specified in the command.

**Cause.** The command referred to a volume whose current state prohibited the operation. For example, a DISABLE DATAVOLS command attempted to disable a data volume that was in the starting state, but the command did not specify the ABRUPT option. As another example, the ENABLE DATAVOLS command attempted to enable a data volume that was not in the disabled state.

**Effect.** The command fails.

**Recovery.** Re-enter the command after the volume's state changes or with the appropriate option specified (if applicable).

## 400

```
volume -- not deleted (dirty, ALLOWINCONSISTENCY not  
specified).
```

*volume*

is the name of the volume specified in the command.

**Cause.** A DELETE DATAVOLS command attempted to remove a data volume in the DOWNDIRTY state, but did not specify the ALLOWINCONSISTENCY option that would have permitted this deletion. The DOWNDIRTY state indicates that the volume is potentially inconsistent.

**Effect.** The volume remains in the configuration.

**Recovery.** If you want to override the TMF protective mechanism that prevents DOWNDIRTY volumes from being deleted, re-enter the DELETE DATAVOLS command, specifying ALLOWINCONSISTENCY.

## 420

```
DELETE TMF operation begun.
```

**Cause.** A DELETE TMF operation, which will purge the entire TMF configuration, has begun.

**Effect.** None; informative message only.

**Recovery.** None required.

## 421

```
A STOP TMF command was issued while TMF was stopped.
```

**Cause.** A STOP TMF command attempted to shut down TMF, but TMF was already stopped.

**Effect.** None.

**Recovery.** None required.

## 422

A STOP TMF command was issued while TMF was starting.

**Cause.** A STOP TMF command attempted to shut down TMF while TMF was being started.

**Effect.** The command fails.

**Recovery.** Re-enter the STOP TMF command, this time using the ABRUPT option, or wait until TMF is fully started and then re-enter the command as it was previously entered.

- 
- △ **Caution.** Use STOP TMF with the ABRUPT option with great care: this option causes TMF to crash and forces the next START TMF command to spend more time in execution.
- 

## 423

A STOP TMF command was issued while TMF was stopping.

**Cause.** A STOP command attempted to shut down TMF, but TMF shutdown was already in progress.

**Effect.** None.

**Recovery.** Re-enter the STOP TMF command, this time using the ABRUPT option, or wait until TMF is fully stopped.

- 
- △ **Caution.** Use STOP TMF with the ABRUPT option with great care: this option causes TMF to crash and forces the next START TMF command to spend more time in execution.
- 

## 424

A DELETE TMF command was issued while TMF was starting.

**Cause.** A DELETE TMF command attempted to purge the TMF configuration while TMF was running.

**Effect.** The command fails.

**Recovery.** Enter a STOP TMF command, wait until TMF stops, and then re-enter the DELETE TMF command.

## 425

A CANCEL OPERATIONS command specified an operation that does not exist.

**Cause.** A CANCEL OPERATIONS command requested removal of a non-existent operation.

**Effect.** The command fails.

**Recovery.** Enter a STATUS OPERATIONS \* command to list all operations currently recorded in the system, and check to ensure that you are specifying the correct operation number.

## 426

A CANCEL OPERATIONS command specified an operation that is in the process of being canceled.

**Cause.** A CANCEL OPERATIONS command requested removal of an operation that TMF is presently removing.

**Effect.** None.

**Recovery.** None required.

## 427

A CANCEL OPERATIONS command specified an operation that has been completed.

**Cause.** A CANCEL OPERATIONS command requested removal of an operation that TMF has already completed.

**Effect.** None.

**Recovery.** None required.

## 440

Operation *operation-number* has begun.

*operation-number*

is an integer assigned by TMF to uniquely identify this specific operation at your terminal and in the EMS log file.

**Cause.** The operation specified has begun.

**Effect.** None.

**Recovery.** Informative message only; no corrective action is needed.

## 441

The command specified an invalid *process-type* parameter.

**Cause.** An ALTER PROCESS or INFO PROCESS command specified an illegal value for the *process-type* parameter. For example, perhaps the intended process type was misspelled.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a proper process type. (For a complete list of valid process types, see the command syntax description.)

## 442

Subsystem Error: An ALTER PROCESS command specified the MAKESAVEABEND option with an illegal value. Contact the Global Customer Support Center (GCSC) or your service provider.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Contact the GCSC or your service provider.



## 443

Subsystem Error: An ALTER PROCESS command specified the DEBUG option with an illegal value. Contact the Global Customer Support Center (GCSC) or your service provider.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Contact the GCSC or your service provider.

## 444

Subsystem Error: An ALTER PROCESS command specified the INSPECT option with an illegal value. Contact the Global Customer Support Center (GCSC) or your service provider.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Contact the GCSC or your service provider.

## 445

The command specified an erroneous SWAP value.

**Cause.** An ALTER PROCESS command specified an invalid volume name in the SWAP parameter. This parameter must indicate a local volume name.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a valid volume name.

## 446

The command specified multiple SWAP values.

**Cause.** An ALTER PROCESS command specified multiple volume names in the SWAP parameter. This parameter must indicate a single local volume name.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a single valid volume name.

## 447

The command specified an erroneous TERM value.

**Cause.** An ALTER PROCESS command specified an invalid file identifier in the TERM parameter.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a valid file identifier.

## 448

The command specified an erroneous EXTENDEDSEGSIZE value. The *integer2* value was not a multiple of *integer1*; or *integer1* or *integer2* was greater than 128; or the values supplied were not equal to, or greater than, the values required by the current TMFSERVE process.

**Cause.** An ALTER PROCESS command attempted to alter the configuration attributes for a process type, but specified an invalid extended segment size (*integer1*, *integer2*, or both) in the EXTENDEDSEGSIZE parameter. For the specific EXTENDEDSEGSIZE limits and default values, see the discussion of the [ALTER PROCESS](#) command in Section 3, and [Table C-6, Process Limits and Defaults](#), on page C-3.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a valid size for the extended segment.

## 449

The command specified an erroneous PFSSIZE value. For BACKOUT, FILERECOVERY, and VOLUMERECOVERY, this value must be 1048576. For all other processes, this value must be between 131072 and 1048576, inclusive.

**Cause.** An ALTER PROCESS command specified an invalid process file segment size in the PFSSIZE parameter.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a valid size for the process file segment.

## 450

File system error *fserrcode* occurred while attempting to access the new configuration volume *confvol*.

*fserrcode*

is the File System error number.

*confvol*

is the new configuration volume.

**Cause.** An ALTER TMF command attempted to access the new configuration volume, but could not do so because of a File System error.

**Effect.** The command fails.

**Recovery.** Refer to the *Guardian Procedure Errors and Messages Manual* for a description of, and recovery actions for, the File System error. Determine why the configuration volume could not be accessed, correct the problem, and retry the command.

## 451

An ALTER TMF command was issued when the TMF subsystem was not stopped.

**Cause.** An ALTER TMF command requested a change, but TMF was running.

**Effect.** The command fails.

**Recovery.** Issue a STOP TMF command before issuing the ALTER TMF command.

## 452

An ALTER TMF command specified more than one volume as the configuration volume, but only one volume is allowed.

**Cause.** An ALTER TMF command specified multiple configuration volumes.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying only one configuration volume.

## 453

The command specified could not be executed because an ALTER TMF, CONFIGVOL command was already in progress.

**Cause.** TMFCOM received a command while TMF was performing an ALTER TMF, CONFIGVOL operation.

**Effect.** The command fails.

**Recovery.** Wait until the ALTER TMF, CONFIGVOL operation completes and then re-enter the command.

## 454

The command specified failed because file system error *fserrcode* occurred while accessing the configuration volume file.

*fserrcode*

is the File System error number.

**Cause.** An ALTER TMF command failed while accessing the configuration volume because a File System error occurred.

**Effect.** The command fails.

**Recovery.** Refer to the *Guardian Procedure Errors and Messages Manual* for a description of, and recovery actions for, the File System error. Determine why the problem occurred, correct it, and retry the command.

## 455

The SwitchProcess command cannot complete because the audit trail files are full. The command is rejected. Add an overflowvol to the MAT or increase the FilesPerVolume value.

**Cause.** An ALTER TMF, SWITCHPROCESS command attempted to switch the primary and backup TMPs, but the operation failed because the audit-trail files were full.

**Effect.** The command fails.

**Recovery.** Increase the space available for audit-trail files by adding an overflow-audit volume to the MAT or increasing the FILESPERVOLUME value.

## 460

The INFO DUMPS Query Server reported File System Error *fserrcode* on catalog file *filename*.

*fserrcode*

is the File System error number.

*filename*

is the file on which the error was encountered.

**Cause.** An INFO DUMPS command encountered a File System error on the catalog file noted.

**Effect.** The command fails.

**Recovery.** Refer to the *Guardian Procedure Errors and Messages Manual* for a description of, and recovery actions for, the File System error. Correct the error and retry the INFO DUMPS command.

## 461

The INFO DUMPS Query Server reported File System Error *fserrcode* while attempting to open catalog file *filename*.

*fserrcode*

is the File System error number.

*filename*

is the file that could not be opened.

**Cause.** An INFO DUMPS command encountered a File System error when opening the catalog file noted.

**Effect.** The command fails.

**Recovery.** Refer to the *Guardian Procedure Errors and Messages Manual* for a description of, and recovery actions for, the File System error. Correct the error and retry the INFO DUMPS command.

## 462

The INFO DUMPS Query Server did not find SerialNumber *serial-number* in the catalog.

*serial-number*

is the serial number of the dump.

**Cause.** An INFO DUMPS command could not locate the dump identified by *serial-number*.

**Effect.** The command fails.

**Recovery.** Ensure that you are using the correct dump serial number, or select another dump and re-enter the command.

## 463

The INFO DUMPS Query Server reported Sort Error *sort-errcode1,sort-errcode2* in the catalog.

*sort-errcode1*

supplies a numeric code returned by FastSort that identifies the error.

*sort-errcode2*

supplies a numeric code returned by FastSort that further defines the error.

**Cause.** An INFO DUMPS command received an error from the FastSort software.

**Effect.** The command fails.

**Recovery.** Refer to the error message descriptions in the *FastSort Manual* for corrective action.

## 464

The INFO DUMPS Query Server did not find any files that matched fileset *fileset*.

*fileset*

is the fileset name specified in the command.

**Cause.** An INFO DUMPS command could not locate the dump files identified by *fileset*.

**Effect.** The command fails.

**Recovery.** Ensure that you are using the correct fileset, or select another fileset and re-enter the command.

## 465

The token code *token* requires a ZTMF-TKN FileSet token in the SPI command language. Contact the Global Customer Support Center (GCSC) or your service provider.

*token*

is the token name specified in the command.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Contact the GCSC or your service provider.

## 466

File System Error *fserrcode* occurred between the TMFSERVE and TMFQRY processes.

*fserrcode*

is the File System error number.

**Cause.** During command execution, the TMFSERVE or TMFQRY processes encountered File System error *fserrcode*.

**Effect.** The command fails.

**Recovery.** Refer to the *Guardian Procedure Errors and Messages Manual* for a description of and recovery actions for the File System error. Correct the error and retry the command.

## 501

The SPI *usedlen* *usedlen* was greater than the maximum *maxlen*.

*usedlen*

is the specified size of the SPI buffer.

*maxlen*

is the maximum size permitted for the SPI buffer.

**Cause.** A Subsystem Programmatic Interface (SPI) violation occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the information supplied in *usedlen* and *lengthread*. Then, contact the GCSC or your service provider.

## 502

Subsystem Error: The Shotgun buffer received by the TMFSERVE process had not been initialized with the TMF SSID. The unrecognized subsystem ID was: *subsystem-id*. Contact the Global Customer Support Center (GCSC) or your service provider.

*subsystem-id*

is the unrecognized subsystem ID (SSID).

**Cause.** A Subsystem Programmatic Interface (SPI) violation occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the information supplied in *subsystem-id*. Then, contact the GCSC or your service provider.



## 503

The Shotgun buffer received by the TMFSERVE process contained a minimum server version *minserv* that was greater than the server version *procserv* of the TMFSERVE process. Contact the Global Customer Support Center (GCSC) or your service provider.

*minserv*

is the minimum server version.

*procserv*

is the *procserv* server version.

**Cause.** A Subsystem Programmatic Interface (SPI) violation occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the information supplied in *minserv* and *procserv*. Then, contact the GCSC or your service provider.

## 504

The SPI buffer TMFSERVE received contained an unknown token code: *unknown-token-code*.

*unknown-token-code*

is the unknown token code.

**Cause.** A Subsystem Programmatic Interface (SPI) violation occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the information supplied in *unknown-token-code*. Then, contact the GCSC or your service provider.

## 505

The SPI buffer TMFSERVE received contained an unknown command: *unknown-command*. The object was: *unknown-object*.

*unknown-command*

is the unknown command.

*unknown-object*

is the object referred to in the command.

**Cause.** A Subsystem Programmatic Interface (SPI) violation occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the information supplied in *unknown-command* and *unknown-object*. Then, contact the GCSC or your service provider.

## 506

The SPI buffer TMFSERVE received contained an unknown object-type: *unknown-object*. The command was: *unknown-command*.

*unknown-object*

is the unknown object referred to in the command.

*unknown-command*

is the unknown command.

**Cause.** A Subsystem Programmatic Interface (SPI) violation occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the information supplied in *unknown-object* and *unknown-command*. Then, contact your the GCSC or service provider.

## 507

Subsystem Error: This is an empty response as defined by the ShotGun CLS.

**Cause.** A Subsystem Programmatic Interface (SPI) violation occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 508

Subsystem Error: The context token received by TMSERVE was not valid.

**Cause.** A Subsystem Programmatic Interface (SPI) violation occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 509

The TMP encountered a resource limitation.

**Cause.** While responding to a STATUS TRANSACTIONS command, the TMP encountered a resource limitation.

**Effect.** The command fails.

**Recovery.** This is a transient condition that lasts only briefly and that TMF corrects automatically. Re-enter the command once again.

## 510

BEGINTRANS disabled.

**Cause.** A DISABLE BEGINTRANS command was issued, but transaction processing is already disabled because of a previous instance of this command.

**Effect.** None.

**Recovery.** Informative message only; no corrective action is needed.

## 511

BEGINTRANS was already enabled by an operator command. However, there may be other reasons why BEGINTRANS is disabled. The STATUS BEGINTRANS command will provide that information.

**Cause.** An ENABLE BEGINTRANS command was issued without a previous DISABLE BEGINTRANS command. BEGINTRANS could still be disabled for other reasons, including an audit trail reaching its begin-transaction-disable threshold, or a processor or the TMP reaching its memory threshold.

**Effect.** The command fails.

**Recovery.** Issue a STATUS BEGINTRANS command to determine the current BEGINTRANS state. Then take corrective action, for example, adding more resources or increasing a threshold.

## 512

The command specified an erroneous TRANSCOUNTTHRESH value.

**Cause.** An ALTER BEGINTRANS command specified one or more invalid values in the TRANSCOUNTTHRESH parameter.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying valid values for TRANSCOUNTTHRESH.

## 513

The command specified an erroneous TMFLIBMEMTHRESH value.

**Cause.** An ALTER BEGINTRANS command specified one or more invalid values in the TMFLIBMEMTHRESH parameter.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying valid values for TMFLIBMEMTHRESH.

## 514

The command specified an erroneous TMPMEMTHRESH value.

**Cause.** An ALTER BEGINTRANS command specified one or more invalid values in the TMPMEMTHRESH parameter.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying valid values for TMPMEMTHRESH.

## 515

The command specified an erroneous TRANSPERCPU value.

**Cause.** An ALTER BEGINTRANS command specified one or more invalid values in the TRANSPERCPU parameter.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying valid values for TRANSPERCPU.

## 516

The command specified an erroneous AUTOABORT value.

**Cause.** An ALTER BEGINTRANS command specified an invalid value in the AUTOABORT parameter.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a valid value for AUTOABORT.

## 521

TMF was unable to delete the catalog because of File System error *fserrcode*.

*fserrcode*

is the File System error number.

**Cause.** A DELETE CATALOG command attempted to remove the TMF catalog, but the condition indicated by *fserrcode* occurred during the delete operation.

**Effect.** The command fails.

**Recovery.** For information about understanding and correcting the error, refer to the File System section of the *Operator Messages Manual* or the *Guardian Procedure Errors and Messages Manual*.

Correct the error condition reported and re-enter the command.

## 550

Invalid parameter supplied.

**Cause.** The command contained an invalid parameter.

**Effect.** The command fails.

**Recovery.** Re-enter the command, correcting or omitting the invalid parameter.

## 603

No existing transactions matched the requested criteria.

**Cause.** A STATUS TRANSACTIONS command requested information about transactions in a certain state (ACTIVE, ABORTING, ABORTED, COMMITTED, HUNG, or PREPARED), but no transactions in that state currently exist in the system.

**Effect.** None.

**Recovery.** Informative message only; no corrective action is needed.

## 604

No volumes matched *volume*.

*volume*

is the volume name.

**Cause.** The command referred to a volume that TMF could not locate.

**Effect.** None.

**Recovery.** Informative message only; no corrective action is needed.

## 605

No files matched *filename*.

*filename*

is the file name.

**Cause.** The command referred to a file that TMF could not locate.

**Effect.** None.

**Recovery.** Informative message only; no corrective action is needed.

## 606

Selection set was empty (no data volumes matched the specified criteria).

**Cause.** The command specified selection criteria that were not satisfied by any data volumes. For example, an INFO DATAVOLS command requested information about all configured data volumes that send their audit records to the AUXILIARY03 audit trail, but no such volumes are defined in the system.

**Effect.** None.

**Recovery.** Informative message only; no corrective action is needed.

## 607

```
volume -- deleted (but could not clear configuration
information).
```

*volume*

is the volume name.

**Cause.** A DELETE DATAVOLS command deleted the data volume from the TMF configuration but could not remove the related information from the volume label. Perhaps the files were inconsistent when the volume was deleted.

**Effect.** The configuration information remains in the volume label.

**Recovery.** If you add this volume to the TMF configuration again, enter the ADD DATAVOLS command with the IGNOREPREVIOUSCONFIG option set ON.

## 608

```
Selection set was empty (no resource managers matched the
specified criteria).
```

**Cause.** An INFO RESOURCEMANAGER or a STATUS RESOURCEMANAGER command specified the name of a resource manager that does not exist.

**Effect.** None.

**Recovery.** Informative message only; no corrective action is needed.

## 781

```
Message text for error nn is not available.  File system
error fserrcode.
```

*nn*

is the error number for which message text could not be obtained.

*fserrcode*

is the associated file-system error number.

**Cause.** You received an error for which TMFCOM has no available message text. This action occurred because you are using an older version of TMFCOM that is not compatible with the existing version of TMFSERVE.

**Effect.** The command fails.

**Recovery.** Enter your commands through a newer, compatible version of TMFCOM.



## 950

Subsystem Error: The TMFSERVE process encountered an internal error. Please report *errcode* to the Global Customer Support Center (GCSC) or your service provider.

*errcode*

supplies a numeric code that further defines the error and identifies the subsystem or system procedure that transmitted this code: TMF, the File System, the NEWPROCESS process-control procedure, or the Enform query language and report writer.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the information supplied in *errcode*. Then, contact the GCSC or your service provider.

## 951

ZEMS error *ems-error* was received from EMS distributor process *process-name* in response to a *zems-command-name* ZEMS request.

*ems-error*

is the EMS error number.

*process-name*

is the name of the EMS distributor process.

*zems-command-name*

is the name of the ZEMS command.

**Cause.** TMF received an unexpected error from the EMSDIST process while processing the specified command.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered, the EMS error number, and the ZEMS command name. Then, contact the GCSC or your service provider.

## 952

Guardian error *fserrcode* was received while communicating with the EMSDIST process *process-name*.

*fserrcode*

supplies a numeric code that further defines the File System error.

*process-name*

is the name of the EMS distributor process.

**Cause.** An error occurred as a result of communication with the named EMSDIST process.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the error code. Then, contact the GCSC or your service provider.

## 953

EMS distributor process *process-name* reported the log-file position request is too far in the future / past the end-of-file position on the log file *filename*.

*process-name*

is the name of the EMS distributor process.

*filename*

is the name of the log file.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the file name. Then, contact the GCSC or your service provider.

## 954

The command did not supply the STATE option.

**Cause.** A RESOLVE TRANSACTION command was issued without the STATE parameter.

**Effect.** The command fails.

**Recovery.** Reissue the command, specifying the STATE option with the desired value.

## 955

EMS distributor process *distributor-process-name* reported that File System error *fserrcode* was returned from the *procedure-name* File System procedure while attempting to access collector process *collector-process-name*.

*distributor-process-name*

is the name of the EMS distributor process.

*fserrcode*

supplies a numeric code that further defines the File System error.

*procedure-name*

is the name of the File System procedure.

*collector-process-name*

is the name of the collector process.

**Cause.** A File System error occurred with access to the collector specified in the command.

**Effect.** The command fails.

**Recovery.** Refer to the *Operator Messages Manual* for a description of the error code. For additional details about understanding and correcting File System errors, see the *Guardian Procedures Errors and Messages Manual*.

Correct the error condition reported, if necessary, and re-enter the command.

## 956

EMS distributor process *process-name* reported that File System error *fserrcode* was returned from the *procedure-name* procedure while attempting to access log file *filename*.

*process-name*

is the name of the EMS distributor process.

*fserrcode*

supplies a numeric code that further defines the File System error.

*procedure name*

is the name of the File System procedure.

*filename*

is the name of the log file.

**Cause.** A File System error occurred with access to the log file specified in the command.

**Effect.** The command fails.

**Recovery.** Refer to the *Operator Messages Manual* for a description of the error code. For additional details about understanding and correcting File System errors, see the *Guardian Procedures Errors and Messages Manual*.

Correct the error condition reported, if necessary, and re-enter the command.

## 957

EMS distributor process *process-name* reported that File System error *fserrcode* was returned from the *procedure-name* File System procedure while attempting to load filter file *file-name*.

*process-name*

is the name of the EMS distributor process.

*fserrcode*

supplies a numeric code that defines the File System error.

*procedure-name*

is the name of the File System procedure (either OPEN or READ) that detected the error.

*file-name*

is the name of the filter file.

**Cause.** A File System error occurred while the named filter file was being loaded.

**Effect.** The command fails.

**Recovery.** Refer to the *Operator Messages Manual* for a description of the error code. For additional details about understanding and correcting File System errors, see the *Guardian Procedures Errors and Messages Manual*.

Correct the error condition reported, if necessary, and re-enter the command.

## 958

Unable to create a process name for EMSDLIST.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 959

```
EMS distributor process process-name reported event filter  
error errcode from filter name filter-name while processing  
log file file-name.
```

*process-name*

is the name of the EMS distributor process.

*errcode*

supplies a numeric code that defines the error.

*filter-name*

is the name of the filter.

*file-name*

is the name of the log file.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered, the filter error, the filter name, and the circumstances. Then, contact the GCSC or your service provider.

## 960

```
EMS distributor process process-name has no event-message  
source or log-file source.
```

*process-name*

is the name of the EMS distributor process.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 961

EMS distributor process *process-name* reported the collector's log file *file-name* had a *reason-text* error.

*process-name*

is the name of the EMS distributor process.

*file-name*

is the name of the log file.

*reason-text*

describes the reason for the error.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 962

EMS distributor process *process-name* could not find the next log file after *file-name*.

*process-name*

is the name of the EMS distributor process.

*file-name*

is the name of the log file.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This message was caused by a missing log file. Load the log file and reissue the command.

## 963

The next log file was unavailable or inaccessible because of a *reason-text* error. The last logfile accessed was *file-name*.

*reason-text*

describes the reason for the error.

*file-name*

is the name of the log file.

**Cause.** A TMF error occurred.

**Effect.** The command fails if this message is not listed as a warning.

**Recovery.** This message appeared because the log file was unavailable for the reason specified. Determine the specific cause of the missing log file, make the log file available, and reissue the command.

## 964

The next log file was unavailable or inaccessible because of a *reason-text* error. The last logfile accessed was *file-name1*. The new logfile is *filename2*.

*reason-text*

describes the reason for the error.

*file-name1*

is the name of the last log file accessed.

*file-name2*

is the name of the new log file.

**Cause.** A TMF error occurred.

**Effect.** The command fails if this message is not listed as a warning.

**Recovery.** This message appeared because the log file was unavailable for the reason specified. Determine the specific cause of the missing log file, make the log file available, and reissue the command.



## 965

The RECOVER FILES command specified the RECOVERVDPPHASE1 option, which also requires a valid TOPHYSVOL option.

**Cause.** A RECOVER FILES command specified the RECOVERVDPPHASE1 option, but omitted the TOPHYSVOL option that is also required.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying both the RECOVERVDPPHASE1 and TOPHYSVOL options.

## 966

The DELETE TMF command was encountered in a command file or in the command line. DELETE TMF must be run interactively.

**Cause.** A DELETE TMF command was entered from a command file, or from the parameter line of a TMFCOM command to start a noninteractive session. This command, however, can only be entered interactively—that is, in response to the TMFCOM interactive prompt character.

**Effect.** The command fails.

**Recovery.** Re-enter the command interactively.

## 967

The RECOVER FILES command supplied an incorrect number of parameters in the TOMATPOSITION option.

**Cause.** A RECOVER FILES command specified, in the TOMATPOSITION option, either too many or too few parameter values.

**Effect.** The command fails.

**Recovery.** Re-enter the RECOVER FILES command, including in the TOMATPOSITION option both the *atseqno* (audit-trail file sequence number) and the *rba* (relative byte address) parameters, but no others.

## 1002

Subsystem Error: Guardian error *errcode* was received while communicating with the TMFSERVE process.

*errcode*

supplies a numeric code that further defines the error and identifies the subsystem or system procedure that transmitted this code: TMF, the File System, the NEWPROCESS process-control procedure, or the Enform query language and report writer.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the information supplied in *errcode*. Then, contact the GCSC or your service provider.

## 1003

Subsystem Error: The requested process name is not a valid process name.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 1004

Subsystem Error: The requested system does not exist.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 1005

Subsystem Error: Unable to create a process name for TMFSERVE.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 1006

Subsystem Error: Unable to OPEN the TMFSERVE process because of error *errcode* from OPEN of process *process-name*.

*errcode*

supplies a numeric code that further defines the error and identifies the subsystem or system procedure that transmitted this code: TMF, the File System, the NEWPROCESS process-control procedure, or the Enform query language and report writer.

*process-name*

is the name of the process that could not be opened because of *errcode*.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the information supplied in *errcode* and *process-name*. Then, contact the GCSC or your service provider.

## 1007

Subsystem Error: Unable to WRITEREAD a startup message to the TMFSERVE process *process-name* because of error *errcode* from WRITEREAD.

*process-name*

is the name of the process that could not receive the startup message.

*errcode*

supplies a numeric code that further defines the error and identifies the subsystem or system procedure that transmitted this code: TMF, the File System, the NEWPROCESS process-control procedure, or the Enform query language and report writer.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the information supplied in *process-name* and *errcode*. Then, contact the GCSC or your service provider.

## 1011

Subsystem Error: The buffer returned from the TMFSERVE process was not initialized with the TMF SSID identifier.

**Cause.** A Subsystem Programmatic Interface (SPI) violation occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 1012

Subsystem Error: To communicate with a process on another system, the HOMETERM of this TMFCOM process must be expressible in network format (7 or fewer characters including the dollar sign). The HOMETERM name is *term-name*.

*term-name*

is the home terminal name.

**Cause.** A Subsystem Programmatic Interface (SPI) violation occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the information supplied in *term-name*. Then, contact the GCSC or your service provider.

## 1013

Subsystem Error: To communicate with a process on another system, the name of this TMFCOM process must be expressible in network format (5 or fewer characters including the dollar sign). The process name is *process-name*.

*process-name*

is the process name.

**Cause.** A Subsystem Programmatic Interface (SPI) violation occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the information supplied in *process-name*. Then, contact the GCSC or your service provider.

## 1014

Subsystem Error: There was no TMFSERVE process attached.

**Cause.** A Subsystem Programmatic Interface (SPI) violation occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 1015

Subsystem Error: An error was encountered while converting a timestamp. Error *errcode* was returned from the Guardian CONVERTTIMESTAMP procedure.

*errcode*

supplies a numeric code that further defines the error and identifies the system procedure that transmitted this code as CONVERTTIMESTAMP.

**Cause.** A Subsystem Programmatic Interface (SPI) violation occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the information supplied in *errcode*. Then, contact the GCSC or your service provider.

## 1016

Subsystem Error: The OUT filename supplied is invalid for use in a network environment.

**Cause.** A Subsystem Programmatic Interface (SPI) violation occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 1017

Subsystem Error: The server named *process-name* already exists, so the new options have been ignored.

*process-name*

is the process name.

**Cause.** A Subsystem Programmatic Interface (SPI) violation occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the information supplied in *process-name*. Then, contact the GCSC or your service provider.

## 1018

The RECOVER FILES command specified the RECOVERVDP PHASE1 option with the MAP NAMES option. These are mutually exclusive.

**Cause.** A RECOVER FILES command included both the RECOVERVDP PHASE1 and MAP NAMES options. However, you can enter only one of these in any single command.

**Effect.** The command fails.

**Recovery.** Re-enter the RECOVER FILES command, specifying either RECOVERVDP PHASE1 or MAP NAMES, but not both.

## 1050

The SPI message to TMFSERVE exceeds the message buffer size. For DUMP FILES, INFO DUMPS, or RECOVER FILES commands, there are too many filenames in the command.

**Cause.** A Subsystem Programmatic Interface (SPI) violation occurred.

**Effect.** The command fails.

**Recovery.** For a DUMP FILES, INFO DUMPS, or RECOVER FILES command, you entered too many filenames in the *file-set* or NOT *file-set* list. In this case, re-enter the command with fewer file names. For other commands, this message should never appear; make a note of the command entered and then contact the GCSC or your service provider.

## 1051

A SPI error occurred with SubType *subtype*, and Error *errcode* while constructing the SPI message to TMFSERVE

*subtype*

supplies an internal code indicating the SPI procedure that caused the error.

*errcode*

supplies a numeric code that further defines the error.

**Cause.** A Subsystem Programmatic Interface (SPI) violation occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and then contact the GCSC or your service provider.

## 2000

Exceeded maximum OBEY depth of 4.

**Cause.** A TMFCOM or OBEY command referred to a command file that itself contained one or more OBEY commands nested more than four levels. (When a command file includes an OBEY command, and that OBEY command refers to another command file that also includes an OBEY command, two levels of nesting exist; this chain can only be continued to a total of four levels.)

**Effect.** The command fails.

**Recovery.** Check the commands in the command file, and follow the chain until you determine how many levels of OBEY command nesting exist. Modify this command file and the ones that depend on it so that no more than four levels exist.

## 2001

This error should not be displayed.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This error should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 2002

File system error *fserrcode* occurred on the OPEN of OBEY file *filename*.

*fserrcode*

supplies a numeric code that further defines the error and identifies the subsystem that transmitted this code as the File System.

*filename*

is the name of the OBEY file.

**Cause.** A File System error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the information supplied in *fserrcode*. Then, contact the GCSC or your service provider.



## 2101

History Error: An attempt to add an empty command line to the history buffer. Contact the Global Customer Support Center (GCSC) or your support representative.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This error should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 2103

Expecting RESET.

**Cause.** A HISTORY command did not specify RESET where TMFCOM expected this keyword.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it correctly.

## 2104

Expecting ALL or LAST.

**Cause.** A HISTORY command did not specify either ALL or LAST where TMFCOM expected one of these keywords.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it correctly.

## 2105

Text Not Found.

**Cause.** An FC, ?, or ! command specified a text string that could not be matched in the history buffer to identify a command on which to operate.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command description. Then re-enter the FC, ?, or ! command with a different text string or other appropriate command line identifier.

## 2108

Line Number not in the history buffer.

**Cause.** An FC, ?, or ! command specified a line number that could not be found in the history buffer to identify a command on which to operate.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command description. Then re-enter the FC, ?, or ! command with a different line number or other appropriate command line identifier.

## 2109

History Error: An attempt to add a command line longer than the history buffer size. Contact the Global Customer Support Center (GCSC) or your support representative.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This error should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 2113

Control-Y entered.

**Cause.** The command included a Control-Y character.

**Effect.** The command fails.

**Recovery.** Check the command syntax. Then re-enter the command, using only the appropriate characters or keywords.

## 2116

Missing end quote.

**Cause.** The command did not specify a delimiting end quotation mark in its parameter list.

**Effect.** The command fails.

**Recovery.** Check the command syntax. Then re-enter the command, supplying the end-quote.

## 2202

File system error *fserrcode* occurred on the I/O.

*fserrcode*

supplies a numeric code that further defines the error and identifies the subsystem that transmitted this code as the File System.

**Cause.** A File System error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the information supplied in *fserrcode*. Then, contact the GCSC or your service provider.

## 2205

Unable to create a process name.

**Cause.** TMF could not create a process name in response to the command just entered. Perhaps TMFSERVE was momentarily down.

**Effect.** The command fails.

**Recovery.** Retry the command just entered.

## 2206

File System error *fserrcode* occurred on the OPEN.

*fserrcode*

supplies a numeric code that further defines the error and identifies the subsystem that transmitted this code as the File System.

**Cause.** A File System error occurred. Perhaps TMFSERVE was momentarily down.

**Effect.** The command fails.

**Recovery.** Correct the File System error, and retry the command just entered.

## 2207

File System error *fserrcode* occurred on the WRITEREAD.

*fserrcode*

supplies a numeric code that further defines the error and identifies the subsystem that transmitted this code as the File System.

**Cause.** A File System error occurred.

**Effect.** The command fails.

**Recovery.** Correct the File System error, and retry the command just entered.

## 2208

*text*

*text*

text that explains an error that occurred while trying to start the TMFSERVE process.

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This error should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 2209

The buffer returned was not in SPI format.

**Cause.** A Subsystem Programmatic Interface (SPI) violation occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 2214

No server is currently running.

**Cause.** The required server did not start.

**Effect.** The command fails.

**Recovery.** Correct the problem, and retry the command just entered.

## 2217

The specified server already exists and cannot be altered.

**Cause.** A Subsystem Programmatic Interface (SPI) violation occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered and the circumstances. Then, contact the GCSC or your service provider.

## 2992

No help for specified text.

**Cause.** The HELP command specified a command name or object for which no online help exists.

**Effect.** The command fails.

**Recovery.** Check the spelling of the command name or object. Then, re-enter the command with this item spelled correctly or with another item specified.

## 30nn

Unable to open *filename* : Error *nn*.

*filename*

is the file on which the error was encountered.

*nn*

is the File System error number.

**Cause.** A File System error occurred while a TMFCOM command was opening an input or output file.

**Effect.** TMFCOM fails.

**Recovery.** Refer to the *Guardian Procedure Errors and Messages Manual* for a description of, and recovery actions for, the File System error. Correct the error and retry the operation

## 7000

Expecting a number.

**Cause.** The command did not specify a numeric value where TMFCOM expected one.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it correctly.

## 7004

Unknown system.

**Cause.** The command referenced a system (node) that was not configured in the network.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a system defined in the network, or see your network manager about bringing the desired system online.

## 7005

Illegal number.

**Cause.** The command contained an improperly specified numeric parameter. A positive number was required, but a non-numeric value or a negative number was specified.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying the number correctly.

## 7006

Expecting priority in the range 1-199.

**Cause.** An ALTER PROCESS command did not specify a valid process priority in the PRI option. A process' priority must be from 1 through 199, inclusive.

**Effect.** The command fails.

**Recovery.** Re-enter the command, using a process priority number from 1 through 199.

## 7007

Expecting FILESIZE in the range 1-2047.

**Cause.** An ADD AUDITTRAIL or ALTER AUDITTRAIL command did not specify a valid file size in the FILESIZE option. An audit trail's file size must be from 1 through 2047, inclusive.

**Effect.** The command fails.

**Recovery.** Re-enter the command, using a file size from 1 through 2047.

## 7008

Expecting FILESPERVOLUME in the range 1-100.

**Cause.** An ADD AUDITTRAIL or ALTER AUDITTRAIL command did not specify a valid value in the FILESPERVOLUME option. This option must specify a value from 1 through 100, inclusive.

**Effect.** The command fails.

**Recovery.** Re-enter the command, using a files-per-volume value from 1 through 100.

## 7009

Expecting OVERFLOWTHRESHOLD in the range 50-100.

**Cause.** An ADD AUDITTRAIL or ALTER AUDITTRAIL command did not specify a valid value in the OVERFLOWTHRESHOLD option. This option must specify a value from 50 through 100, inclusive.

**Effect.** The command fails.

**Recovery.** Re-enter the command, using an OVERFLOWTHRESHOLD value from 50 through 100.

## 7010

Expecting BEGINTRANSDISABLE in the range 50-100.

**Cause.** An ADD AUDITTRAIL or ALTER AUDITTRAIL command did not specify a valid value in the BEGINTRANSDISABLE option. This option must specify a value from 50 through 100, inclusive.

**Effect.** The command fails.

**Recovery.** Re-enter the command, using a BEGINTRANSDISABLE value from 50 through 100.

## 7011

Expecting CPU number in the range 0-15.

**Cause.** An ALTER PROCESS command did not specify a valid value in the CPU option, perhaps using a number greater than 15 or a negative value for a processor number. This value must be an integer from 0 through 15, inclusive.

**Effect.** The command fails.

**Recovery.** Re-enter the command, using an integer from 0 through 15 for the processor number.

## 7012

Expecting PFSSIZE in the range 131,072 - 1,048,576.

**Cause.** An ALTER PROCESS command did not specify a valid process file segment size in the PFSSIZE option. This value must be an integer from 131,072 through 1,048,576.

**Effect.** The command fails.

**Recovery.** Re-enter the command, using a valid process file segment size in the PFSSIZE option.

## 7013

Expecting valid tape volume. Must be 1 - 6 numeric characters or 1 -6 alphanumeric characters starting with an alpha character.

**Cause.** An ADD TAPEMEDIA or ALTER TAPEMEDIA command did not specify a valid tape-volume name. This required parameter must begin with either a letter or a digit. If the name begins with a digit, up to five digits can follow. If the name begins with a letter, up to five digits or letters in any combination can follow.

**Effect.** The command fails.

**Recovery.** Re-enter the command, using a valid tape volume name.



## 7014

Expecting audit trail sequence number between 1 and 999999.

**Cause.** A RECOVER FILES command did not specify a valid audit-trail sequence number in the TOMATPOSITION parameter. This sequence number must be the numeric portion of the MAT file name, with leading zeros optional, ranging from 1 through 999999.

**Effect.** The command fails.

**Recovery.** Re-enter the command, using a valid audit trail sequence number.

## 7015

Expecting audit trail RBA number between 0 and 2147483647.

**Cause.** A RECOVER FILES command did not specify a valid relative byte address (rba) in the TOMATPOSITION parameter. This rba must be a number ranging from 0 through 2147483647.

**Effect.** The command fails.

**Recovery.** Re-enter the command, using a valid rba.

## 7016

Expecting TAPEMEDIA copy number between 1 and 2.

**Cause.** An ADD DUMPS command did not specify a valid number of copies in the TAPEMEDIA option. This value must be either 1 or 2.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a valid number of copies.

## 7017

Expecting TAPEMEDIA part number; value must be greater than zero.

**Cause.** An ADD DUMPS command did not specify a valid part number in the TAPEMEDIA parameter, indicating which part of the dump should be stored on the volume. This value must be a number greater than zero.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a valid dump part number.

## 7018

Expecting DENSITY value of 0, 1600, or 6250.

**Cause.** An ADD TAPEMEDIA or SET TAPEMEDIA command did not specify a valid value in the DENSITY option. This value must be 0, 1600, or 6250.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a valid density.

## 7019

Expecting one of 0, 1600, or 6250.

**Cause.** The command did not specify 0, 1600, or 6250 where TMFCOM expected one of these values.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it correctly.

## 7020

Expecting a TAPEMEDIA name. Format is *tape-volume:part:copy*.

**Cause.** An ADD DUMPS or SET DUMPS command did not specify a valid tape volume identification in the TAPEMEDIA parameter. This identification must be in the format *tape-volume:part:copy*.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying the tape volume identification in the correct format.

## 7021

Expecting TAPEMEDIA copy number (1-2).

**Cause.** An ADD DUMPS or SET DUMPS command did not specify a valid number of copies in the TAPEMEDIA option. This value must be either 1 or 2.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a valid number of copies.

## 7022

Expecting ASSIGNED, BAD, RELEASED, or SCRATCH.

**Cause.** An ADD TAPEMEDIA, ALTER TAPEMEDIA, INFO TAPEMEDIA, or SET TAPEMEDIA command did not specify a correct tape status where TMFCOM expected one of the following: ASSIGNED, BAD, RELEASED, or SCRATCH.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it correctly.

## 7023

Expecting TAPEMEDIA part number; value must be greater than zero.

**Cause.** An ADD DUMPS or SET DUMPS command did not specify a valid part number in the TAPEMEDIA parameter, indicating which part of the dump should be stored on the volume. This value must be a number greater than zero.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a valid dump part number.

## 7024

Missing final quote.

**Cause.** A command parameter included a beginning quote mark, but omitted the closing quote mark, in a subparameter.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command. Then, re-enter the command, including the final quote.

## 7025

Illegal character.

**Cause.** The command contained an illegal character.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command. Then, re-enter the command, using legal characters only.

## 7026

```
Missing parameter.
```

**Cause.** The command omitted one or more required parameter strings.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command. Then, re-enter the command, specifying all required strings.

## 7027

```
Expecting a quoted string.
```

**Cause.** The command did not specify a quoted string where TMFCOM expected one.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it correctly.

## 7028

```
Keyword too long.
```

**Cause.** The command contained a keyword that was too long.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command. Then, re-enter the command, specifying the keyword properly.

## 7029

```
Unterminated continuation line.
```

**Cause.** The command spanned multiple lines, but at least one continuation line was not terminated by an ampersand (&).

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying an ampersand at the end of each continuation line.

## 7030

Command line too long.

**Cause.** The command exceeded the number of characters permitted in the command line.

**Effect.** The command fails.

**Recovery.** Re-enter the command, using the continuation indicator (&) to allow the command to span multiple entry lines.

## 7031

Expecting semicolon or return.

**Cause.** The command did not include a delimiting semicolon or terminating return where TMFCOM expected one.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7032

Expecting colon.

**Cause.** The command did not include a required delimiting colon where TMFCOM expected one.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it correctly.

## 7033

Missing parameter.

**Cause.** The command omitted one or more required parameters.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command. Then, re-enter the command, specifying all required parameters.

## 7034

```
Illegal process name.
```

**Cause.** The command specified an illegal process name.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command. Then, re-enter the command, specifying an appropriate process name.

## 7035

```
Illegal DISKMEDIA name.
```

**Cause.** The command specified an illegal name in the DISKMEDIA parameter.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command. Then, re-enter the command, specifying an appropriate DISKMEDIA name.

## 7036

```
Expecting a DISKMEDIA name.
```

**Cause.** The command did not include a disk name in the DISKMEDIA parameter.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command. Then, re-enter the command, specifying an appropriate DISKMEDIA name.

## 7037

```
Expecting process name.
```

**Cause.** The command did not include a process name where TMFCOM expected one.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command. Then, re-enter the command, specifying a process name.

## 7038

Expecting comma or return.

**Cause.** The command did not include a delimiting comma or terminating return where TMFCOM expected one.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7039

Expecting comma or ")".

**Cause.** The command did not include a required delimiting comma or terminating (closing) parenthesis where TMFCOM expected one.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7040

Expecting keyword.

**Cause.** The command did not specify a keyword where TMFCOM expected one, probably as a command object or parameter.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7041

Unrecognized object.

**Cause.** The command specified an unrecognizable object in a parameter, for example a misspelled or erroneous object name.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command. Then, re-enter the command, using the proper object keyword or name.

## 7042

Expecting one of CPU, PRI, SWAP, or TERM. The SWAP and TERM options are validated but are not used.

**Cause.** The command specified an unrecognizable optional parameter, for example a misspelled or erroneous keyword.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7043

Expecting one of ABORT, ADD, ALTER, CANCEL, CLOSE, CMDVOL, DELETE, DISABLE, DISPLAY, DUMP, ENABLE, ENV, EXIT, FC, HELP, HISTORY, INFO, NEXT, OBEY, OBEYVOL, OPEN, RECOVER, RELOCATE, RESET, RESOLVE, SET, SHOW, START, STATUS, STOP, or VOLUME.

**Cause.** The command entered is not supported by TMFCOM.

**Effect.** The command fails.

**Recovery.** Enter the HELP ALL or HELP \* command to display a list of all TMFCOM commands, and select a valid command to enter.

## 7044

This parameter has been specified more than once.

**Cause.** The command contained more than one instance of the same parameter.

**Effect.** The command fails.

**Recovery.** Re-enter the command with only one instance of this parameter.

## 7045

Expecting ON or OFF.

**Cause.** An ALTER PROCESS or ALTER CATALOG command contained a keyword parameter that must be set to either ON or OFF, but attempted to use another value instead. In particular, the MAKESAVEABEND, DEBUG, and INSPECT options in ALTER PROCESS and the ROUNDROBIN option in ALTER CATALOG can only be set to ON or OFF.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command. Then, re-enter the command, specifying the ON or OFF keyword as indicated.



## 7046

Expecting a valid file name.

**Cause.** The command did not specify a valid file name where TMFCOM expected one.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7047

Expecting a valid file name set.

**Cause.** The command did not specify a valid file-set where TMFCOM expected one.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7048

Expecting SERVER.

**Cause.** An OPEN command was entered without the SERVER keyword.

**Effect.** The command fails.

**Recovery.** Re-enter the command, including the SERVER keyword.

## 7049

Expecting TRANS[ACTION[S]].

**Cause.** An ABORT TRANSACTION command did not specify TRANS, TRANSACTION, or TRANSACTIONS where TMFCOM expected one of these keywords.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7050

Expecting AUDITTRAIL, DATAVOL[S], DUMPS, RESOURCEMANAGER, RM, or TAPEMEDIA.

**Cause.** An ADD command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7052

Expecting AUDITDUMP, AUDITTRAIL, BEGINTRANS, CATALOG, DATAVOL[S], DUMPS, PROCESS, TAPEMEDIA, or TMF.

**Cause.** An ALTER command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7053

Expecting OPERATION[S].

**Cause.** A CANCEL command omitted the required OPERATION or OPERATIONS keyword.

**Effect.** The command fails.

**Recovery.** Re-enter the command, including the required keyword.

## 7054

Expecting CATALOG, DATAVOL[S], DUMPS, RESOURCMANAGER, RM, TAPEMEDIA, TMF, or TRANS[ACTION].

**Cause.** A DELETE command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7056

Expecting AUDITDUMP, BEGINTRANS, or DATAVOL[S].

**Cause.** A DISABLE command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7058

Expecting OPERATION[S].

**Cause.** A DISPLAY command omitted the required OPERATION or OPERATIONS keyword.

**Effect.** The command fails.

**Recovery.** Re-enter the command, including the OPERATION or OPERATIONS keyword.

## 7059

Expecting FILES.

**Cause.** A DUMP command omitted the required FILES keyword.

**Effect.** The command fails.

**Recovery.** Re-enter the command, including the FILES keyword.

## 7060

Expecting AUDITDUMP, BEGINTRANS, or DATAVOLS.

**Cause.** An ENABLE command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7062

Expecting AUDITDUMP, AUDITTRAIL, BEGINTRANS, CATALOG, DATAVOL[S], DUMPS, RESOURCEMANAGER, RM, PROCESS, TAPEMEDIA, or TMF.

**Cause.** An INFO command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7064

Expecting AUDITTRAIL.

**Cause.** A NEXT command omitted the required AUDITTRAIL keyword.

**Effect.** The command fails.

**Recovery.** Re-enter the command, including the AUDITTRAIL keyword.

## 7065

Expecting FILES.

**Cause.** A RECOVER command omitted the required FILES keyword.

**Effect.** The command fails.

**Recovery.** Re-enter the command, including the FILES keyword.

## 7066

Expecting DISKDUMPS.

**Cause.** A RELOCATE command did not specify DISKDUMPS, where TMFCOM expected this keyword.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7067

Expecting AUDITTRAIL, DATAVOL[S], DUMPS, or TAPEMEDIA.

**Cause.** A RESET command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7068

Expecting TRANSACTION.

**Cause.** A RESOLVE command omitted the required TRANSACTION keyword.

**Effect.** The command fails.

**Recovery.** Re-enter the command, including the TRANSACTION keyword.

## 7069

Expecting AUDITTRAIL, DATAVOL[S], DUMPS, or TAPEMEDIA.

**Cause.** A SET command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7070

Expecting AUDITTRAIL, DATAVOL[S], DUMPS, or TAPEMEDIA.

**Cause.** A SHOW command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7071

Expecting TMF.

**Cause.** A START command omitted the required TMF keyword.

**Effect.** The command fails.

**Recovery.** Re-enter the command, including the TMF keyword.

## 7072

Expecting AUDITDUMP, AUDITTRAIL, BEGINTRANS, CATALOG, DATAVOL[S], OPERATIONS[S], SERVER, TMF, or TRANS[ACTION[S]].

**Cause.** A STATUS command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7074

Expecting AUDITDUMP, FILESIZE, ACTIVEVOL[S], FILESPERVOLUME, OVERFLOWVOL[S], OVERFLOWTHRESHOLD, RESTOREVOLS[S], BEGINTRANSDISABLE, or "\*".

**Cause.** An ADD AUDITTRAIL, ALTER AUDITTRAIL, RESET AUDITTRAIL, or SET AUDITTRAIL command did not specify a correct keyword or special character where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7075

Expecting AUDITTRAIL, RECOVERYMODE, or "\*".

**Cause.** An ADD DATAVOLS or SET DATAVOLS command did not specify a correct keyword or special character where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7076

Expecting ONLINE or ARCHIVE.

**Cause.** An ADD DATAVOLS or SET DATAVOLS command did not specify the correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7077

Expecting DISKMEDIA, TAPEMEDIA, SERIAL, TYPE, or "\*".

**Cause.** An ADD DUMPS command did not specify a correct keyword or special character where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7078

Expecting DENSITY, DRIVE, LABEL, STATUS, UNLOAD, or "\*".

**Cause.** An ADD TAPEMEDIA command did not specify a correct keyword or special character where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7079

Expecting STATUS.

**Cause.** An ADD TAPEMEDIA or ALTER TAPEMEDIA command did not specify STATUS where TMFCOM expected this keyword.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7080

Expecting STATUS or OBEYFORM.

**Cause.** An INFO DUMPS command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7081

Expecting AUDITDUMP or ONLINEDUMP.

**Cause.** An ADD DUMPS command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7082

Expecting serial number greater than 63.

**Cause.** An ADD DUMPS or RECOVER FILES command specified an invalid serial number in the SERIAL parameter used to identify a dump. This serial number must be 64 or greater.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a dump serial number of 64 or more.

## 7083

Expecting dump file name.

**Cause.** An ADD DUMPS command did not specify a dump file name where TMFCOM expected one.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.



## 7084

Expecting only one DISKMEDIA disk.

**Cause.** The command specified multiple disk volumes in the DISKMEDIA option, but only one disk volume is allowed.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying only a single disk volume.

## 7085

Cannot specify both DISKMEDIA and TAPEMEDIA in a single command.

**Cause.** The command specified both the DISKMEDIA and TAPEMEDIA options, but only one of these is allowed.

**Effect.** The command fails.

**Recovery.** Re-enter the command with either the DISKMEDIA or TAPEMEDIA option, but not both.

## 7086

Must specify either DISKMEDIA and TAPEMEDIA.

**Cause.** The command required either the DISKMEDIA or TAPEMEDIA option, but neither of these was included.

**Effect.** The command fails.

**Recovery.** Re-enter the command with either the DISKMEDIA or TAPEMEDIA option.

## 7087

Expecting AUXnn, AUXILIARYnn, MASTER, or MAT.

**Cause.** The command did not include an audit-trail specification AUXnn, AUXILIARYnn, MASTER, or MAT, where TMFCOM expected one of these elements.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7088

Expecting TMF.

**Cause.** A START or STOP command did not include the TMF object keyword.

**Effect.** The command fails.

**Recovery.** Re-enter the command with the TMF keyword.

## 7092

Expecting OBEYFORM.

**Cause.** An INFO command did not specify OBEYFORM where TMFCOM expected this keyword.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7093

Expecting PROCESS.

**Cause.** An ALTER or INFO command did not specify PROCESS where TMFCOM expected this keyword.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7094

Expecting one of AUDITDUMP, AUDITRESTORE, BACKOUT, CATALOG, COPYAUDITOVFLOW, FILERECOVERY, FILERECOVLIST, ONLINEDUMP, ONLINERESTORE, TMFMON, TMFMON2, TMP, VOLUMERECOVERY, or "\*".

**Cause.** An ALTER PROCESS command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7096

Expecting one of CPUS, PRI, MAKESAVEABEND, DEBUG, INSPECT, PROGRAMFILE, TERM, EXTENDEDSEGSIZE, PFSSIZE, or RESET.

**Cause.** An ALTER PROCESS command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7097

Expecting one of CPUS, PRI, PROGRAMFILE, TERM, EXTENDEDSEGSIZE, PFSSIZE, or SWAP.

**Cause.** An ALTER PROCESS command did not specify a correct RESET option.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7098

Expecting RM or RESOURCEMANAGER.

**Cause.** The command did not specify the required RM or RESOURCEMANAGER object keyword.

**Effect.** The command fails.

**Recovery.** Re-enter the command with one of these two keywords.

## 7099

Expecting CPU numbers.

**Cause.** An ALTER PROCESS command specified the CPUS option, but the required *cpu-list* was not present.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7100

Expecting a CPU number.

**Cause.** An ALTER PROCESS command specified the CPUS option, but no processor number was present.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7101

Expecting a valid device name.

**Cause.** The command did not specify a valid device name where TMFCOM expected one.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7102

Expecting a valid volume name.

**Cause.** The command did not specify a valid volume name where TMFCOM expected one.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7103

Expecting a valid TMF transaction identifier.

**Cause.** The command did not specify a valid transaction identifier where TMFCOM expected one.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7104

Expecting a single TMF transaction identifier without "(...)".

**Cause.** The command specified a single transaction identifier but enclosed it within parentheses. (Use parentheses only when specifying multiple identifiers.)

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7105

Expecting AVOIDHANGING or IGNOREDATAERRORS.

**Cause.** An ABORT TRANSACTION command did not specify either AVOIDHANGING or IGNOREDATAERRORS where TMFCOM expected one of these keywords.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7106

Expecting only one of AVOIDHANGING or IGNOREDATAERRORS.

**Cause.** An ABORT TRANSACTION command specified both the AVOIDHANGING and IGNOREDATAERRORS options, but only one of these is allowed.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying either AVOIDHANGING or IGNOREDATAERRORS, but not both.

## 7107

Expecting AUDITTRAIL, NOT, RECOVERYMODE, or IGNOREPREVIOUSCONFIG.

**Cause.** An ADD DATAVOLS command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7108

Expecting ONLINE or ARCHIVE.

**Cause.** The command did not specify either ONLINE or ARCHIVE where TMFCOM expected one of these keywords.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7109

Expecting a valid BLOCKSIZE value. Must be one of 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, or 52.

**Cause.** The command specified a BLOCKSIZE value that was not a multiple of 4 and within the range 4 through 52.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a proper BLOCKSIZE value.

## 7110

Expecting AUDITTRAIL, OBEYFORM, RECOVERYMODE, or SORT

**Cause.** An INFO DATAVOLS command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7111

Expecting return or ";"

**Cause.** The command did not include a delimiting carriage return or semicolon (;) where TMFCOM expected one.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7112

Expecting "(".

**Cause.** An ALTER PROCESS command was entered with the CPUS option, but the *cpu-list* did not include a leading left-parenthesis.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it with the required left-parenthesis.

## 7113

Expecting ")"

**Cause.** An ALTER PROCESS command was entered with the CPUS option, but the *cpu-list* did not include a trailing right-parenthesis.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it with the required right-parenthesis.

## 7114

Expecting ","

**Cause.** The command did not specify a delimiting comma where TMFCOM expected one.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7115

More than 16 CPU numbers in list.

**Cause.** An ALTER PROCESS command was entered with the CPUS option, but the *cpu-list* specified more than 16 processors.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command. Then, re-enter the command, specifying 16 or fewer processors.

## 7116

More than 16 volume names in list.

**Cause.** The command included a DISKMEDIA parameter that specified more than 16 disk volumes.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command. Then, re-enter the command, specifying 16 or fewer volume names.

## 7117

Unable to use an 8-character volume name as the default.

**Cause.** A VOLUME command specified a disk volume on a node other than the local node.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command. Then, re-enter the command, specifying a local disk volume.

## 7118

Expecting EXTENDEDSEGSIZE integer1 value between 1 and 128. The minimum value for integer1 depends on the process type and software release.

**Cause.** An ALTER PROCESS command specified an *integer1* value for the EXTENDEDSEGSIZE parameter that was outside the range allowed for the process. In general, TMFCOM accepts values between 1 and 128, but the minimum value actually allowed in specific cases depends on the process type and software release.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command. Then, re-enter the command, specifying an *integer1* value in the correct range.



## 7119

Expecting EXTENDEDSEGSIZE integer2 value between 1 and 128. The minimum value for integer2 depends on the process type and software release.

**Cause.** An ALTER PROCESS command specified an *integer2* value for the EXTENDEDSEGSIZE parameter that was outside the range allowed for the process. In general, TMFCOM accepts values between 1 and 128, but the minimum value actually allowed in specific cases depends on the process type and software release.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command. Then, re-enter the command, specifying an *integer2* value in the correct range.

## 7120

EXTENDEDSEGSIZE integers are not valid. Integer2 must be equal to or greater than integer1 and integer2 must be a multiple of integer1.

**Cause.** An ALTER PROCESS command specified invalid EXTENDEDSEGSIZE values for *integer1* and *integer2*; for example, an *integer2* value that was less than the *integer1* value or an *integer2* value that was not a multiple of the *integer1* value.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command. Then, re-enter the command, specifying proper *integer1* and *integer2* values.

## 7121

Illegal volume or subvolume specified.

**Cause.** A VOLUME command specified an illegal volume or subvolume name.

**Effect.** The command fails.

**Recovery.** Verify that the volume or subvolume name complies with the syntax rules listed in [File Names and Process Identifiers](#) on page 3-6. Then, re-enter the command, using a correct volume name.

## 7122

Illegal file name.

**Cause.** The command contained an invalid file name—perhaps one that contained improper characters or was too long.

**Effect.** The command fails.

**Recovery.** Verify that the file name complies with the syntax rules listed in [File Names and Process Identifiers](#) on page 3-6. Then, re-enter the command with a correct file name.

## 7123

Unable to open OUT file.

**Cause.** The command contained an OUT option that specified an invalid file name. Perhaps the file name was too long or contained improper characters.

**Effect.** The command fails.

**Recovery.** Verify that the file name complies with the syntax rules listed in [File Names and Process Identifiers](#) on page 3-6. Then, re-enter the command, using a valid file name.

## 7124

Illegal system name.

**Cause.** The command specified an illegal system (node) name.

**Effect.** The command fails.

**Recovery.** Verify that the system name complies with the syntax rules listed in [File Names and Process Identifiers](#) on page 3-6. Then, re-enter the command, using a correct system name.

## 7125

First character must be alphabetic.

**Cause.** The command contained a subvolume name or file identifier that did not begin with a letter.

**Effect.** The command fails.

**Recovery.** Verify that the subvolume name or file identifier complies with the syntax rules listed in [File Names and Process Identifiers](#) on page 3-6. Then, re-enter the command, ensuring that the subvolume name or file identifier begins with a letter.

## 7126

Comment exceeds maximum length.

**Cause.** The command line contained a comment that was too long. Perhaps the ampersand was not used as a continuation character to spread the command over multiple lines.

**Effect.** The command fails.

**Recovery.** Re-enter the command line, limiting the comment to the required size.

## 7127

Maximum length exceeded.

**Cause.** The command contained a file specifier that was too long.

**Effect.** The command fails.

**Recovery.** Verify that the file specifier complies with the syntax rules listed in [File Names and Process Identifiers](#) on page 3-6. Then, re-enter the command, ensuring that the file specifier does not violate its length limit.

## 7128

No previous command to FC.

**Cause.** An FC command attempted to repeat or correct a previously entered command, but no preceding commands exist in the command buffer; this FC command was the first command entered in this TMFCOM session.

**Effect.** None; informative message only.

**Recovery.** None required.

## 7129

```
Wild-card characters not allowed.
```

**Cause.** The command included a parameter that contained wild-card characters, but such characters are prohibited for this parameter.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it correctly.

## 7130

```
The filename must be restricted.  
[[{volume|*}.]{subvolume|*}.]{file-id|*}
```

**Cause.** The command specified a filename that included an element other than a legal volume name, subvolume name, or file identifier.

**Effect.** The command fails.

**Recovery.** Verify that the volume name, subvolume name, or file identifier complies with the syntax rules listed in [File Names and Process Identifiers](#) on page 3-6. Then, re-enter the command, following these rules.

## 7131

```
Escape character must be followed by a wild-card character.
```

**Cause.** A command parameter included an escape character that was not followed by the required wild-card character.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7132

```
Expecting owner ID: GROUP #, USER #.
```

**Cause.** The command did not specify the owner identity (group and user name or number) where TMFCOM expected it.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7134

Expecting a quote.

**Cause.** The command did not specify a quotation mark (") where TMFCOM expected one.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7135

No concatenated commands are allowed after ?, !, or FC.

**Cause.** In a command line containing multiple commands, a command followed the ?, !, or FC command. (Although other commands permit this, ?, !, and FC do not.)

**Effect.** The command fails.

**Recovery.** Re-enter the commands on separate lines.

## 7136

?, !, and FC must be used interactively, must be the only command on the command line, and may not be used within an OBEY file or with the OUT option.

**Cause.** A ?, !, or FC command was entered non-interactively, or was entered on a command line with one or more other commands, or was encountered in an OBEY file, or was issued with the OUT option. (Commands other than ?, !, and FC are permitted any of these entry options.)

**Effect.** The command fails.

**Recovery.** Re-enter the command without the OBEYFORM or OUT option.

## 7137

Illegal device or volume specified.

**Cause.** A command parameter specified an illegal device or volume name.

**Effect.** The command fails.

**Recovery.** Verify that the device identifier or volume name complies with the syntax rules listed in [File Names and Process Identifiers](#) on page 3-6. Then, re-enter the command, using a correct device identifier or volume name.

## 7138

Illegal default volume specified.

**Cause.** A VOLUME command specified an illegal default volume name.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7140

Expecting a "/".

**Cause.** The command did not specify a slash (/) character where TMFCOM expected one.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command. Then, re-enter the command, specifying the slash delimiter in the correct location.

## 7141

Expecting OUT.

**Cause.** The command attempted to direct command output to an alternate destination, but omitted the OUT keyword where it was required.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the OUT option in [Directing Output to an Alternate Destination](#), on page 2-8. Then, re-enter the command, specifying OUT in the correct location.

## 7142

Expecting a SYSTEM (node) name.

**Cause.** The command did not specify a system (node) name where TMFCOM expected this name.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7145

Expecting a period.

**Cause.** The command did not specify a delimiting period where TMFCOM expected one. Such periods are used, for example, as delimiters in qualified file names.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter the command accordingly.

## 7151

Invalid TMF transaction identifier.

**Cause.** The command contained a TMF transaction identifier that was not valid.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a valid transaction identifier.

## 7152

Invalid year. Must be between 0001 and 4000.

**Cause.** The TIME option of an ADD DUMPS or RECOVER FILES command, or the BEGINTIME option of a DISPLAY OPERATIONS or STATUS OPERATIONS command, did not specify the year as a four-digit value from 0001 through 4000.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a valid year value.

## 7153

Invalid month. Must be JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, or DEC.

**Cause.** The TIME option of an ADD DUMPS or RECOVER FILES command, or the BEGINTIME option of a DISPLAY OPERATIONS or STATUS OPERATIONS command, did not specify the month in the correct form.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a valid month.

## 7154

```
Invalid day.  Must be 01 - 31.
```

**Cause.** The TIME option of an ADD DUMPS or RECOVER FILES command, or the BEGINTIME option of a DISPLAY OPERATIONS or STATUS OPERATIONS command, did not specify the day as a two-digit integer ranging from 01 through 31.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a valid day.

## 7155

```
Invalid date-time field.  Must be  
"Month Day Year, Hr:Min[:Sec] [GMT]",  
"Year Month Day, Hr:Min[:Sec] [GMT]", OR  
"Year-Month-Day, Hr:Min[:Sec] [GMT]".
```

**Cause.** The TIME option of an ADD DUMPS or RECOVER FILES command, or the BEGINTIME option of a DISPLAY OPERATIONS or STATUS OPERATIONS command, did not specify the month, day, and year in the proper form.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying the month, day, and year in the correct format.

## 7156

```
The specified year is not a leap year, but FEB 29 was  
entered.
```

**Cause.** The TIME option of an ADD DUMPS or RECOVER FILES command, or the BEGINTIME option of a DISPLAY OPERATIONS or STATUS OPERATIONS command, specified FEB 29 for a year that was not a leap year.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying FEB 28 or earlier, or a different month or year.



## 7157

The day number is not valid for the month entered.

**Cause.** The TIME option of an ADD DUMPS or RECOVER FILES command, or the BEGINTIME option of a DISPLAY OPERATIONS or STATUS OPERATIONS command, specified an invalid day for the month entered--for example, JUN 31 or SEP 31.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a valid day of the month.

## 7158

Invalid Hour entered. Must be 00 - 23.

**Cause.** The TIME option of an ADD DUMPS or RECOVER FILES command, or the BEGINTIME option of a DISPLAY OPERATIONS or STATUS OPERATIONS command, did not specify the hour as a two-digit integer ranging from 00 through 23.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a valid hour.

## 7159

Invalid Minute entered. Must be 00 - 59.

**Cause.** The TIME option of an ADD DUMPS or RECOVER FILES command, or the BEGINTIME option of a DISPLAY OPERATIONS or STATUS OPERATIONS command, did not specify the minute value as a two-digit integer ranging from 00 through 59.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a valid minute.

## 7160

Invalid Second entered. Must be 00 - 59.

**Cause.** The TIME option of an ADD DUMPS or RECOVER FILES command, or the BEGINTIME option of a DISPLAY OPERATIONS or STATUS OPERATIONS command, did not specify the second value as a two-digit integer ranging from 00 through 59.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a valid second.

## 7161

Expecting GMT.

**Cause.** An ADD DUMPS, RECOVER FILES, DISPLAY OPERATIONS, or STATUS OPERATIONS command did not specify GMT where TMFCOM expected this keyword.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7162

Expecting SERIAL, TYPE, DISKMEDIA, TAPEMEDIA, or TIME.

**Cause.** An ADD DUMPS command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7163

Expecting BLOCKSIZE, COPIES, SYSTEM, VERIFYTAPE, UNLOAD, MEDIUM, ADDDISKMEDIA, DELDISKMEDIA, or RESET

**Cause.** An ALTER AUDITDUMP command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7164

Expecting TAPE or DISK.

**Cause.** The command did not specify either TAPE or DISK where TMFCOM expected one of these keywords.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7165

More than two SYSTEM names were entered.

**Cause.** The command specified more than two system (node) names, but only two are allowed.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7166

Expecting BLOCKSIZE, COPIES, MEDIUM, or SYSTEM.

**Cause.** The command did not specify the correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7167

Expecting AUDITDUMP, ADDACTIVEVOL[S], DELETEACTIVEVOL[S],  
FILESPERVOLUME, ADDOVERFLOWVOL[S],  
DELETEOVERFLOWVOL[S], OVERFLOWTHRESHOLD, ADDRESTOREVOLS[S],  
DELETERESTOREVOL[S], BEGINTRANSDISABLE, or RESET.

**Cause.** An ALTER AUDITTRAIL command did not specify the correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7168

Expecting BEGINTRANSDISABLE, FILESPERVOLUME, or  
OVERFLOWTHRESHHOLD.

**Cause.** A RESET AUDITTRAIL command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7169

Illegal COPIES. Must be 1, 2, or 3.

**Cause.** The command specified an invalid number of dump copies in the COPIES parameter. This parameter can indicate only 1 or 2 parallel copies, or from 1 to 3 serial copies.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a valid number of copies.

## 7170

Expecting SERIES.

**Cause.** The command did not specify SERIES where TMFCOM expected this keyword.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7171

Expecting PARALLEL or SERIES.

**Cause.** The command did not specify either PARALLEL or SERIES where TMFCOM expected one of these keywords.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7172

Expecting AUTOABORT, BRANCHESPERM, RECRMOUNT, RESET, RMOPENPERCPU, TMFLIBMEMTHRESH, TMPMEMTHRESH, TRANSCOUNTTHRESH, or TRANSPERCPU.

**Cause.** An ALTER BEGINTRANS command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7173

Expecting AUTOABORT, BRANCHESPERM, RECRMOUNT, RMOPENPERCPU, TMFLIBMEMTHRESH, TMPMEMTHRESH, TRANSCOUNTTHRESH, or TRANSPERCPU.

**Cause.** A RESET BEGINTRANS command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7174

Expecting HOUR[S], MINUTE[S], or SECOND[S].

**Cause.** The TIME option of an ADD DUMPS or RECOVER FILES command, or the BEGINTIME option of a DISPLAY OPERATIONS or STATUS OPERATIONS command, did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7175

Invalid AUTOABORT HOUR value. Must be between 1 and 5965.

**Cause.** The AUTOABORT option of an ALTER BEGINTRANS command did not specify the HOUR value as an integer ranging from 1 through 5965.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a valid hour value.

## 7176

Invalid AUTOABORT MINUTE value. Must be between 1 and 357913.

**Cause.** The AUTOABORT option of an ALTER BEGINTRANS command did not specify the MINUTE value as an integer ranging from 1 through 357913.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a valid minute value.

## 7177

Invalid AUTOABORT SECOND value. Must be between 20 and 21474836.

**Cause.** The AUTOABORT option of an ALTER BEGINTRANS command did not specify the SECOND value as an integer ranging from 20 through 21474836.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a valid second value.

## 7178

Invalid BRANCHESPERRM value. Must be between 16 and 1024.

**Cause.** An ALTER BEGINTRANS command specified a BRANCHESPERRM value that was outside the range 16 through 1024.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a valid BRANCHESPERRM value.

## 7179

Invalid RECRMOUNT value. Must be between 16 and 16384.

**Cause.** An ALTER BEGINTRANS command specified a RECRMOUNT value that was outside the range 16 through 16384.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a valid BRANCHESPERRM value.

## 7180

Invalid RMOPENPERCPU value. Must be between 16 and 1024.

**Cause.** An ALTER BEGINTRANS command specified a RMOPENPERCPU value that was outside the range 16 through 1024.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a valid RMOPENPERCPU value.

## 7181

Incorrect threshold values. The values must be between 1 and 100, and value1 must be less than value2.

**Cause.** An ALTER BEGINTRANS command specified one or more incorrect threshold values in the TMFLIBMEMTHRESH or the TMPMEMTHRESH parameters, or in both of these parameters. Each of the values must be in the range from 1 to 100, and for each parameter, the first value entered must be less than the second value entered.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a correct TMFLIBMEMTHRESH or TMPMEMTHRESH value.

## 7182

Incorrect TRANSCOUNTTHRESH values. The values must be between 1 and 1073741824, and value1 must be less than value2.

**Cause.** An ALTER BEGINTRANS command specified one or more incorrect values in the TRANSCOUNTTHRESH parameter. Each of the values must be in the range from 1 to 1073741824, and the first value entered must be less than the second value entered.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying correct TRANSCOUNTTHRESH values.

## 7183

Incorrect TRANSPERCPU values. The values must be between 1 and 1073741824.

**Cause.** An ALTER BEGINTRANS command specified one or more incorrect values in the TRANSPERCPU parameter. Each of the values must be in the range from 1 to 1073741824.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying correct TRANSPERCPU values.

## 7184

Expecting RELEASED, RESET, or RETAINDEPTH.

**Cause.** An ALTER CATALOG command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7185

Expecting RETAINDEPTH.

**Cause.** An ALTER CATALOG command did not specify RETAINDEPTH where TMFCOM expected this keyword.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7186

Incorrect RETAINDEPTH value. The value must be between 1 and 99.

**Cause.** An ALTER CATALOG command specified an incorrect value in the RETAINDEPTH parameter. This value must be in the range from 1 to 99.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command. Then, re-enter the command, specifying a valid RETAINDEPTH value.



## 7187

Expecting INVALID, RELEASED, or SERIAL.

**Cause.** An ALTER DUMPS command did not specify the correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7188

Expecting a dump file name.

**Cause.** An ALTER DUMPS command did not specify a dump file name.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7189

Invalid dump file name. Must be a local name, in either of these formats:

```
[[volume.]{subvolume | *}.]{file-id | *} or  
$ZTMFAT.ZTMFAT.{file-id | *}.
```

**Cause.** An ALTER DUMPS command did not specify a valid dump file name in the proper format.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying the dump file name in a valid format.

## 7190

Expecting CONFIGVOL or SWITCHPROCESS.

**Cause.** An ALTER TMF command did not specify either SWITCHPROCESS or CONFIGVOL where TMFCOM expected one of these keywords.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7191

Expecting TMP.

**Cause.** The command did not specify TMP where TMFCOM expected this keyword.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7192

Expecting an operation number.

**Cause.** A CANCEL OPERATIONS or DISPLAY OPERATIONS command did not specify an operation number.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7193

Expecting a single operation number, without "(...)".

**Cause.** A STATUS OPERATIONS command specified a single operation number, but enclosed it within parentheses. (Use parentheses only when specifying multiple identifiers.)

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7194

A "\*" operation number is not valid in this context.

**Cause.** A STATUS OPERATIONS command included the symbol for all operations (\*) in an invalid context--for example, when one or more specific operation numbers also appeared in the command.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command. Then, re-enter the command, including only the \* symbol or omitting it altogether.

## 7195

Invalid operation number. Must be between 1 and 100.

**Cause.** A CANCEL OPERATIONS, DISPLAY OPERATIONS, or STATUS OPERATIONS command specified an operation number that did not fall within the range from 1 through 100.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying an operation number in the correct range.

## 7196

Expecting ALLOWINCONSISTENCY.

**Cause.** A DELETE DATAVOLS command did not specify ALLOWINCONSISTENCY where TMFCOM expected this keyword.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7197

Expecting SERIAL.

**Cause.** A DELETE DUMPS command did not specify SERIAL where TMFCOM expected this keyword.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7198

A wild-card name "\*" must be entered only once, and it must be the only name.

**Cause.** The command specified the wild-card character (\*) as an object name multiple times, or along with other names.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7199

A wild-card value "\*" is not valid in this context.

**Cause.** This command does not allow the wild-card character (\*) in the location entered.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7200

A NOT volume list is only valid with a wild-card "\*" volume name.

**Cause.** This command only allows the wild-card character (\*) as a volume name in the NOT *fileset* parameter list.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7201

Expecting NOT.

**Cause.** The command did not specify NOT where TMFCOM expected this keyword.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7202

Expecting a resource manager name.

**Cause.** An ADD RESOURCEMANGER, CLOSE RESOURCEMANGER, DELETE RESOURCEMANGER, INFO RESOURCEMANGER, or STATUS RESOURCEMANGER command did not specify a resource manager name.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying the resource manager name.

## 7203

A wild-card resource manager name "\*" must be entered only once and it must be the only resource manager name.

**Cause.** The command specified the wild-card character (\*) as a resource manager name multiple times, or along with other resource manager names.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7204

A wild-card resource manager name value "\*" is not valid for this command.

**Cause.** This command does not allow the wild-card character (\*) as a parameter value.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7205

Volatile resource manager name "ZZ-VOLATILE-" is not valid for this command.

**Cause.** An ADD RESOURCEMANGER, CLOSE RESOURCEMANGER, DELETE RESOURCEMANGER, INFO RESOURCEMANGER, or STATUS RESOURCEMANGER command specified the resource manager name ZZ-VOLATILE-, but this name is not permitted.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7206

Invalid resource manager name. Must be 1 to 128 characters long, start with an alphabetic character, and contain alphanumeric or the following characters \$, \_, -, ^, @, &.

**Cause.** An ADD RESOURCEMANGER, CLOSE RESOURCEMANGER, DELETE RESOURCEMANGER, INFO RESOURCEMANGER, or STATUS RESOURCEMANGER command specified an invalid resource manager name.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command. Then, re-enter the command, specifying a valid name.

## 7207

Expecting a username or userid.

**Cause.** The command did not specify a user name or user id where TMFCOM expected one of these elements.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7208

Expecting a userid. Must be (group-id,user-id), where group-id and user-id are numbers between 0 and 255.

**Cause.** The command did not specify a user id where TMFCOM expected one.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7209

Invalid group-id or user-id value. Must be between 0 and 255.

**Cause.** The command did not specify a valid group id or user id, which must be an integer in the range 0 through 255.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command. Then, re-enter the command, specifying a valid identifier.

## 7210

Invalid or non-existent group-name.user-name.

**Cause.** The command did not specify a valid group name or user name.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command. Then, re-enter the command, specifying a valid name.

## 7211

Expecting a CPU,PIN.

**Cause.** A STATUS RESOURCEMANAGER or STATUS TRANSACTIONS command did not specify a *cpu,pin* value where TMFCOM expected one.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7212

Invalid CPU,PIN.

**Cause.** A STATUS RESOURCEMANAGER or STATUS TRANSACTIONS command did not specify a valid *cpu,pin* value.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command. Then, re-enter the command, specifying a valid *cpu,pin* value.

## 7213

Expecting a (CPU,PIN) or process name.

**Cause.** A STATUS RESOURCEMANAGER or STATUS TRANSACTIONS command did not specify a (*cpu,pin*) or *process-name* value where TMFCOM expected one of these values.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7214

Expecting only a process name.

**Cause.** A STATUS RESOURCEMANAGER or STATUS TRANSACTIONS command specified values in addition to the process identifier in the PROCESS parameter.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7215

Invalid process name.

**Cause.** A STATUS RESOURCEMANAGER or STATUS TRANSACTIONS command did not specify a valid process identifier.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command. Then, re-enter the command, specifying a valid process identifier.

## 7216

Expecting one of CREATED, CLOSED, OPEN, RECOVERING, CRASHED, or VOLATILE.

**Cause.** A STATUS RESOURCEMANAGER command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7217

Expecting one of STATE, PROCESS, or DETAIL.

**Cause.** A STATUS RESOURCEMANAGER command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.



## 7218

Expecting one of BRIEF, DETAIL, MEDIUM, OBEYFORM, SERIAL, SORT, STATUS, or TYPE.

**Cause.** An INFO DUMPS command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7219

Expecting TAPE or DISK.

**Cause.** An INFO DUMPS command did not specify either TAPE or DISK where TMFCOM expected one of these keywords.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7220

Expecting ASSIGNED, INVALID, or RELEASED.

**Cause.** An INFO DUMPS command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7221

Expecting BYFILE or BYSERIAL.

**Cause.** An INFO DUMPS command did not specify BYFILE or BYSERIAL where TMFCOM expected one of these keywords.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7222

Expecting AUDITDUMP or ONLINEDUMP.

**Cause.** An INFO DUMPS command did not specify AUDITDUMP or ONLINEDUMP where TMFCOM expected one of these keywords.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7223

Expecting only one item of this type in this context. Lists of multiple items are not allowed.

**Cause.** The command specified multiple items where TMFCOM expected only one item.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7224

Expecting BEGINTIME, COLLECTOR, LOGFILE, or LONG.

**Cause.** A DISPLAY OPERATIONS command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7225

Invalid COPIES count. Must be 1 or 2.

**Cause.** A DUMP FILES command specified an invalid number of dump copies to be made in the COPIES parameter. This parameter can request only 1 or 2 copies.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying either 1 or 2 copies.

## 7226

Expecting PARALLEL.

**Cause.** A DUMP FILES command did not specify PARALLEL where TMFCOM expected this keyword.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7227

Expecting BLOCKSIZE, COPIES, DISKMEDIA, NOT, START, SYSTEM, TAPEMEDIA, UNLOAD, or VERIFYTAPE.

**Cause.** A DUMP FILES command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7228

Expecting NOT, FROMARCHIVE, NOWAIT, TOFIRSTPURGE, SYSTEM, TIME, UNLOAD, TOPHYSVOL, WHEREPHYSVOLSIS, or RECOVERYVDPHASE1.

**Cause.** A RECOVER FILES command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7229

Expecting OLDDMLOC or NEWDMLOC.

**Cause.** A RELOCATE DUMPS command did not specify either OLDDMLOC or NEWDMLOC where TMFCOM expected one of these keywords.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7230

Expecting both OLDDMLOC or NEWDMLOC disk media locations.

**Cause.** A RELOCATE DUMPS command did not specify both OLDDMLOC and NEWDMLOC where TMFCOM expected these keywords.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7231

Expecting a diskmedia location.

**Cause.** A RELOCATE DUMPS command omitted the disk media location in either or both of the OLDDMLOC or NEWDMLOC parameters.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7232

Invalid diskmedia location. Must be [`\Node.`]{`*`|`volume`}.

**Cause.** A RELOCATE DUMPS command specified an invalid disk media location in either or both of the OLDDMLOC or NEWDMLOC parameters. This value must be entered in the format [`node.`]{`*`|`volume`}.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command. Then, re-enter the command, specifying a valid disk media location in the correct format.

## 7233

Expecting NOWAIT, ENABLE, or DISABLE.

**Cause.** A START TMF command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7234

Expecting NOWAIT or ABRUPT.

**Cause.** A STOP TMF command did not specify either NOWAIT or ABRUPT where TMFCOM expected one of these keywords.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7235

Expecting BEGINTRANS.

**Cause.** A START TMF command did not specify BEGINTRANS where TMFCOM expected this keyword.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7236

Expecting DETAIL.

**Cause.** The command did not specify DETAIL where TMFCOM expected this keyword.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7237

Expecting ADDING, STARTING, STARTUPBEGUN, AWAITTRANSRESOLVE, QUEUEDFORRECOVERY, RECOVERING, STARTUPCOMPLETING, STARTED, NOTSTARTED, DOWN, DOWNCLEAN, DOWNDIRTY, SHUTTINGDOWN, DISABLING, DISABLED, DISABLEDCLEAN, DISABLEDDIRTY, UNCONFIGURED, or WRONGDISKMOUNTED.

**Cause.** A STATUS DATAVOLS command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it correctly.

## 7238

Expecting AUDITTRAIL, RECOVERYMODE, SORT, or STATE.

**Cause.** A STATUS DATAVOLS command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it correctly.

## 7239

Expecting BEGINTIME, SORT, STATE, or TYPE.

**Cause.** A STATUS OPERATIONS command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it correctly.

## 7240

Expecting INPROGRESS, FINISHED, or ALL.

**Cause.** A STATUS OPERATIONS command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it correctly.

## 7241

Expecting AUDITRESTORE, DELETETMF, DUMPFILES, RECOVERFILES, STARTTMF, STOPTMF, VOLUMERECOVERY, or ANY.

**Cause.** A STATUS OPERATIONS command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it correctly.

## 7242

Expecting ACTIVE, ABORTING, ABORTED, COMMITTED, ENDING, HUNG, or PREPARED.

**Cause.** A STATUS TRANSACTIONS command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it correctly.

## 7243

Expecting PROCESS, STATE, DETAIL, or BRIEF.

**Cause.** A STATUS TRANSACTIONS command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it correctly.

## 7244

FILESET size exceeds 8192 bytes.

**Cause.** The command specified a file-set size greater than 8192 bytes.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a file-set of 8192 bytes or less in length.

## 7245

The currently opened TMFSERVE process does not recognize the TOMATPOSITION option of the RECOVER FILES command.

**Cause.** The current TMFCOM session is directing its commands to an older version of TMFSERVE in which the TOMATPOSITION option is not available.

**Effect.** The command fails.

**Recovery.** With the OPEN SERVER command, open a TMFSERVE process that incorporates the TOMATPOSITION feature. Then, re-enter the RECOVER FILES command.

## 7246

PROCESS\_CREATE\_ error *error*, *error-detail* on file *program-file-name* occurred while attempting to create an EMS Distributor process. At this time, you can correct the problem and THEN respond to the following query to retry or cancel the operation.

*error*

is the error code returned by the PROCESS\_CREATE\_ procedure.

*error-detail*

is the error-detail code returned by the PROCESS\_CREATE\_ procedure, and further defines the error that occurred. This is usually a file-system error code.

*program-file-name*

is the name of the program file that PROCESS\_CREATE\_ was trying to create.

**Cause.** A PROCESS\_CREATE\_ error occurred.

**Effect.** The command fails.

**Recovery.** Make a note of the command entered and the information supplied in *error*, *error-detail*, and *program-file-name* and look up the PROCESS\_CREATE\_ error. Correct the problem as indicated, and then retry or cancel the operation.



## 7247

File System Error - *fserrcode* occurred when attempting to issue a WRITEHEAD request to the EMS Distributor *process-name*. At this time, you can correct the problem and THEN respond to the following query to retry or cancel the operation.

*fserrcode*

supplies a numeric code that further defines the error and identifies the subsystem that transmitted this code as the File System.

*process-name*

is the name of the distributor process.

**Cause.** A File System error occurred.

**Effect.** The command fails.

**Recovery.** Make a note of the command entered and the information supplied in *fserrcode*, and look up the File System error. Correct the problem as indicated, and then retry or cancel the operation.

## 7248

File System Error - *fserrcode* occurred when attempting to replace the existing EMS filter with the filter object file *filename*. At this time, you can correct the problem and THEN respond to the following query to retry or cancel the operation.

*fserrcode*

supplies a numeric code that further defines the error and identifies the subsystem that transmitted this code as the File System.

*filename*

is the name of the filter object file.

**Cause.** A File System error occurred.

**Effect.** The command fails.

**Recovery.** Make a note of the command entered and the information supplied in *fserrcode*, and look up the File System error. Correct the problem as indicated, and then retry or cancel the operation.

## 7249

In this context, the filename cannot contain a system name.

**Cause.** The command specified a filename that included a system (node) name, but node names are prohibited.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying the file name without the node name.

## 7250

Cannot use both TAPEMEDIA and DISKMEDIA in the same command.

**Cause.** An ADD DUMPS command specified both TAPEMEDIA and DISKMEDIA, but only one of these options is permitted.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying either TAPEMEDIA or DISKMEDIA, but not both.

## 7251

Expecting OBEYFORM or CNFGFORM.

**Cause.** The command did not specify either OBEYFORM or CNFGFORM where TMFCOM expected one of these keywords.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7252

Duplicate Volume Name in list.

**Cause.** The command specified the same volume name two or more times in a parameter list.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying the volume name only once.

## 7253

The filename contains a component (volume, subvolume, or filename) that contains more than 8 characters, including wild-card characters.

**Cause.** The command specified a volume, subvolume, or file identifier that was too long.

**Effect.** The command fails.

**Recovery.** Re-enter the command with filename components of 8 characters or less.

## 7254

The HELP request contains more than 31 non-blank characters.

**Cause.** A HELP command exceeded 31 characters (excluding spaces).

**Effect.** The command fails.

**Recovery.** Re-enter the command, using 31 or fewer non-blank characters.

## 7255

Expecting BYATID or BYNAME.

**Cause.** An INFO DATAVOLS or STATUS DATAVOLS command did not specify either BYATID or BYNAME where TMFCOM expected one of these keywords.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7256

OBEYFORM requires a SERIAL number.

**Cause.** The INFO DUMPS command did not specify a serial number with the OBEYFORM parameter.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7257

MEDIUM, STATUS, NOT, and START are not allowed with OBEYFORM.

**Cause.** The INFO DUMPS command specified an improper keyword with the OBEYFORM parameter; MEDIUM, STATUS, NOT and START are not allowed with OBEYFORM.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7258

OBEYFORM requires a single file-set of "\*\*.\*.\*".

**Cause.** The INFO DUMPS command specified multiple \*.\*.\* file-set indicators with the OBEYFORM parameter.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command. Then, re-enter the command, including only one \*.\*.\* file-set indicator with OBEYFORM.

## 7259

The currently opened TMFSERVE process does not recognize the TOMATPOSITION option of the RECOVER FILES command. Requires TMFSERVE version D46ADW, G07V01 or later.

**Cause.** The current TMFCOM session is directing its commands to an older version of TMFSERVE in which the TOMATPOSITION is not available in the RECOVER FILES command.

**Effect.** The command fails.

**Recovery.** With the OPEN SERVER command, open a TMFSERVE process running TMFSERVE version D46ADW, G07V01, or later. Then, re-enter the RECOVER FILES command.

## 7260

The currently opened TMFSERVE process does not support the START, NOT, CATALOG, or multiple file-sets in the RECOVER FILES or INFO DUMPS command. Requires TMFSERVE version G07AEN or later.

**Cause.** The current TMFCOM session is directing its commands to an older version of TMFSERVE in which the START, NOT, and CATALOG options as well as multiple file-set specifications are not available in the RECOVER FILES command.

**Effect.** The command fails.

**Recovery.** With the OPEN SERVER command, open a TMFSERVE process running TMFSERVE version G07AEN or later. Then, re-enter the RECOVER FILES command.

## 7262

Expecting a positive number.

**Cause.** The command did not specify a positive number where TMFCOM expected one.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7263

Expecting BYNUMBER or BYTIME.

**Cause.** The command did not specify either BYNUMBER or BYTIME where TMFCOM expected one of these keywords.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7264

A node name is not valid in this context.

**Cause.** The command specified a node name where TMFCOM did not expect one.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7265

Expecting RECOVERYMODE.

**Cause.** An ALTER DATAVOLS command did not specify RECOVERYMODE where TMFCOM expected this keyword.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7266

Expecting AUDITTRAIL or RECOVERYMODE.

**Cause.** A SET DATAVOLS command did not specify either AUDITTRAIL or RECOVERYMODE where TMFCOM expected one of these keywords.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7267

Expecting AUDITDUMP, FILESIZE, ACTIVEVOL[S], FILESPERVOLUME, OVERFLOWVOL[S], OVERFLOWTHRESHOLD, RESTOREVOL[S], or BEGINTRANSDISABLE.

**Cause.** A SET AUDITTRAIL command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7268

Expecting DISKMEDIA, TAPEMEDIA, SERIAL, or TYPE.

**Cause.** A SET DUMPS command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7269

Expecting DENSITY, DRIVE, LABEL, STATUS, or UNLOAD.

**Cause.** A SET TAPEMEDIA command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7270

Expecting a START filename with format  
[[*\$volume.*]subvol.]{file-id|\*} or [*volume.*]\*.\*.

**Cause.** The START *start-file* option in a command did not specify a filename of the correct format.

**Effect.** The command fails.

**Recovery.** Re-enter the command with a properly formatted file name.

## 7271

A dump serial number was not specified in either the ADD DUMPS or SET DUMPS command; ",SERIAL serial-num".

**Cause.** The SERIAL option in an ADD DUMPS or SET DUMPS command did not specify a dump serial number.

**Effect.** The command fails.

**Recovery.** Re-enter the command with the dump serial number included.

## 7272

A dump type was not specified in either the ADD DUMPS or SET DUMPS command; ",TYPE [AUDITDUMP | ONLINEDUMP ]".

**Cause.** The TYPE option in an ADD DUMPS or SET DUMPS command did not specify the dump type.

**Effect.** The command fails.

**Recovery.** Re-enter the command, including the dump type, which must be either AUDITDUMP or ONLINEDUMP.

## 7273

The `"/OUT..."` option is not allowed in an OBEY command.

**Cause.** An OBEY command specified the OUT option, directing output to an alternate output device. For the OBEY command, this option is illegal.

**Effect.** The command fails.

**Recovery.** Re-enter the command without the OUT option.

## 7274

The required LABEL value is not specified in the command and has not been set by a SET TAPEMEDIA LABEL...command

**Cause.** An ADD TAPEMEDIA command required a LABEL value, but this value was missing.

**Effect.** The command fails.

**Recovery.** Re-enter the ADD TAPEMEDIA command, or enter a SET TAPEMEDIA command, specifying either LABEL ON or LABEL OFF.

## 7275

Expecting one of BRIEF, CATALOG, DETAIL, NOT, MEDIUM, OBEYFORM, START, SERIAL, SORT, STATUS, or TYPE.

**Cause.** An INFO DUMPS command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.

## 7276

Expecting OWNER.

**Cause.** An ADD RESOURCEMANAGER command did not specify OWNER where TMFCOM expected this keyword.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command and re-enter it accordingly.



## 7277

Expecting MAXRETAINEDATFILES in the range 1-9999.

**Cause.** An ALTER AUDITTRAIL command specified a MAXRETAINEDATFILES *integer* value outside the proper range.

**Effect.** The command fails.

**Recovery.** Re-enter the command, with a MAXRETAINEDATFILES *integer* value from 1 through 9999.

## 7278

The currently opened TMFSERVE process does not support MAXRETAINEDATFILES in ALTER AUDITTRAIL, MAP NAMES and SERIAL in RECOVER FILES, or ROUNDROBIN in ALTER CATALOG. Requires TMFSERVE version G07AFG or later.

**Cause.** An ALTER AUDITTRAIL, RECOVER FILES, or ALTER CATALOG command included an option that is not supported by the version of TMFSERVE that received the command.

**Effect.** The command fails.

**Recovery.** Ensure that TMFCOM opens the appropriate version of TMFSERVE.

## 7279

Expecting AUDITDUMP, ADDACTIVEVOL[S], DELETEACTIVEVOLS[S], FILESPERVOLUME, ADDOVERFLOWVOL[S], DELETEOVERFLOWVOL[S], OVERFLOWTHRESHOLD, ADDRESTOREVOL[S], DELETERESTOREVOL[S], BEGINTRANSDISABLE, MAXRETAINEDATFILES, or RESET.

**Cause.** An ALTER AUDITTRAIL command did not specify the correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command, then re-enter it.

## 7280

Expecting BEGINTRANSDISABLE, FILESPERVOLUME, OVERFLOWTHRESHOLD, or MAXRETAINEDATFILES.

**Cause.** An ALTER AUDITTRAIL command with a RESET option, or a RESET AUDITTRAIL command, did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command, then re-enter it.

## 7281

Expecting RELEASED, RESET, RETAINDEPTH, or ROUNDROBIN.

**Cause.** An ALTER CATALOG command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command, then re-enter it.

## 7282

Invalid MAP NAMES, unexpected end of input.

**Cause.** A RECOVER FILES command included the MAP NAMES option, but did not terminate this option in a valid way. Perhaps a closing parenthesis did not follow the last *new-fileset*, or *new-fileset* contained multiple *file-set* identifiers.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command, then re-enter it.

## 7283

Invalid MAP NAMES, MAP expected.

**Cause.** A RECOVER FILES command did not specify the MAP keyword where TMFCOM expected it in the MAP NAMES option.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command, then re-enter it.

## 7284

```
Invalid MAP NAMES, NAMES expected.
```

**Cause.** A RECOVER FILES command did not specify the NAMES keyword where TMFCOM expected it in the MAP NAMES option.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command, then re-enter it.

## 7285

```
INVALID MAP NAMES, TO expected.
```

**Cause.** A RECOVER FILES command did not specify the TO keyword where TMFCOM expected it in the MAP NAMES option.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command, then re-enter it.

## 7286

```
Invalid MAP NAMES file set.
```

**Cause.** A RECOVER FILES command specified an improperly formed *file-set* in the MAP NAMES option.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for specifying a *file-set*. Then, re-enter the command, specifying the *file-set* correctly.

## 7287

```
Invalid MAP NAMES file set, expected it to be restricted.
```

**Cause.** A RECOVER FILES command specified, in the MAP NAMES option, a *new-file-set* that included a component name (volume name, subvolume name, or file ID) that contained the wild-card character \* (asterisk) among other characters. However, the wild-card character (\*) can only be used in place of a full component name. For example, \$volume1.\*.filea is correct, but \$volume1.sub\*.filea is not.

**Effect.** The command fails.

**Recovery.** Re-enter the command, using the wild-card character properly in *new-fileset*.

## 7288

```
Invalid MAP NAMES, comma or right parenthesis missing.
```

**Cause.** A RECOVER FILES command omitted a comma or right-parenthesis where TMFCOM expected one in the MAP NAMES option.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command, then re-enter it.

## 7289

```
Unexpected error returned by FileNameMap^Scan.
```

**Cause.** A TMF error occurred.

**Effect.** The command fails.

**Recovery.** This message should never appear. Make a note of the command entered. Then, contact the GCSC or your service provider.

## 7290

```
MAP NAMES string exceeds 28K bytes in length.
```

**Cause.** A RECOVER FILES command specified too many characters in the MAP NAMES option. This option cannot include more than 28 KB.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying fewer than 28,672 characters in the MAP NAMES parameter.

## 7291

```
MAP NAMES does not support file sets in the form  
$volume.file-id.
```

**Cause.** A RECOVER FILES command specified an improperly formed *file-set* in the MAP NAMES option. When a *file-set* includes volume name, it must also specify a subvolume name. In this case, the subvolume name was omitted.

**Effect.** The command fails.

**Recovery.** Re-enter the command, specifying a subvolume name, or select another *file-set*.

## 7292

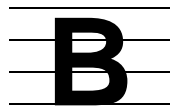
|  |
|--|
| Expecting NOT, FROMARCHIVE, NOWAIT, TOFIRSTPURGE,<br>TOMATPOSITION, SYSTEM, TIME, UNLOAD, TOPHYSVOL,<br>WHEREPHYSVOLIS, RECOVERVDPHASE1, SERIAL, or MAP NAMES. |
|--|

**Cause.** A RECOVER FILES command did not specify a correct keyword where TMFCOM expected one of those listed in the message.

**Effect.** The command fails.

**Recovery.** Check the syntax rules for the command, then re-enter it.





# TMFCOM Command Reference Summary

This appendix presents syntax descriptions for the individual TMFCOM commands, listed in alphabetical order.

| <b>Topic</b>                   | <b>Page</b>                 |
|--------------------------------|-----------------------------|
| <a href="#"><u>AB- AD</u></a>  | <a href="#"><u>B-2</u></a>  |
| <a href="#"><u>AL - CA</u></a> | <a href="#"><u>B-7</u></a>  |
| <a href="#"><u>CL - DE</u></a> | <a href="#"><u>B-15</u></a> |
| <a href="#"><u>DI - FC</u></a> | <a href="#"><u>B-18</u></a> |
| <a href="#"><u>HE - IN</u></a> | <a href="#"><u>B-22</u></a> |
| <a href="#"><u>NE - RE</u></a> | <a href="#"><u>B-27</u></a> |
| <a href="#"><u>SE - SH</u></a> | <a href="#"><u>B-34</u></a> |
| <a href="#"><u>ST - VO</u></a> | <a href="#"><u>B-37</u></a> |

# ABORT TRANSACTION

Backs out (cancels) the effects of an active transaction, or re-attempts to back out an aborting or hung transaction, and releases the locks held by that transaction.

Security: super user group member.

```
ABORT TRANS[ACTION[S]]

    { transaction-id
      { (transaction-id [, transaction-id]...) }

    [, IGNOREDATAERRORS | , AVOIDHANGING]

transaction-id:
    [ node .                               ]
    [ node-number .                         ]
    [ node(tm-flags) .                     ] cpu.seq-num
    [ node-number(tm-flags) .             ]
    [ (tm-flags) .                         ]
```

△ **Caution.** Contact the Global Customer Support Center (GCSC) or your service provider before issuing this command with IGNOREDATAERRORS specified.

Specifying IGNOREDATAERRORS can lead to database corruption and loss of data integrity. Specify IGNOREDATAERRORS only if all efforts to abort a transaction fail and data availability is more important than data integrity.



# ADD AUDITTRAIL

Adds an audit trail to the TMF configuration.

Security: super user group member.

```
ADD AUDITTRAIL {MASTER | MAT
                {AUX[ ILIARY]nn
                }

[,AUDITDUMP {ON | OFF
            }]

[,FILESIZE integer
]

[,ACTIVEVOL[S] {volume
                {(volume [,volume]...)
                }
]

[,FILESPERVOLUME integer
]

[,OVERFLOWVOL[S] {volume
                  {(volume [,volume]...)
                  }
]

[,OVERFLOWTHRESHOLD integer
]

[,RESTOREVOL[S] {volume
                 {(volume [,volume]...)
                 }
]

[,BEGINTRANSDISABLE integer
]
```

# ADD DATAVOLS

Adds one or more data volumes to the TMF configuration, including management attributes such as the audit trail to receive audit records from these volumes.

Security: super user group member.

```
ADD DATAVOL[S] {volume
                 {(volume [,volume]...)
                 * [,NOT volume]
                 * [,NOT (volume[,volume]...)]
                 }

[,AUDITTRAIL {MASTER | MAT
              {AUX[ ILIARY]nn
              }
]

[,RECOVERYMODE {ONLINE | ARCHIVE}
]

[,IGNOREPREVIOUSCONFIG
]
```

**△ Caution.** Volume recovery cannot repair any inconsistencies that exist when a data volume is added with IGNOREPREVIOUSCONFIG specified, even if the data volume is returned to its previous configuration environment.

# ADD DUMPS

Adds dump entries to the TMF catalog, as when reentering online and audit dump entries accidentally deleted from the catalog or recovering a destroyed catalog. There are two forms of the ADD DUMPS command: one for adding an audit dump entry to the catalog, and another for adding one or more online dump entries to the catalog.

Security: super user group member.

## Adding an Audit Dump to the Catalog

To add an audit dump entry to the catalog, use the following syntax:

```
ADD DUMPS audit-trail-file
    ,SERIAL serial-number
    ,TYPE AUDITDUMP
    { ,DISKMEDIA disk
      { ,TAPEMEDIA { tape
                    { (tape [, tape]...) } }
    }

    [,TIME { month[-]day[-]year, hr:min[:sec] [GMT]}
           { day[-]month[-]year, hr:min[:sec] [GMT]} ]

audit-trail-file:
    $ZTMFAT.ZTMFAT.file-id

disk
    [node.]volume . {xx}serial-number . {zz}sequence-number

tape:
    tape-volume:part:copy

month:
    { JAN | FEB | MAR | APR | MAY | JUN | JUL | AUG | SEP | OCT | NOV | DEC }
```

---

△ **Caution.** If recovery is needed, problems could occur if the time listed in the catalog is not the time when the audit dump actually took place; if you specify the TIME option, make sure to enter the date and time completely and accurately.

---

## Adding Online Dump File Names to the Catalog

To add one or more online dump entries to the catalog, use the following syntax:

```
ADD DUMPS {online-dump-file
           {(online-dump-file [,online-dump-file]...)}

           ,SERIAL serial-number

           ,TYPE ONLINEDUMP (master, data)

           {,DISKMEDIA {disk
                        {(disk [,disk]...)}}}
           {,TAPEMEDIA {tape
                        {(tape [,tape]...)}}}

           [,TIME {month[-]day[-]year, hr:min[:sec] [GMT]}
                  {day[-]month[-]year, hr:min[:sec] [GMT]}]

online-dump-file:
    [volume.][subvolume.]file-id

disk:
    [node.]volume.{xx}serial-number.Ffile-number

tape:
    tape-volume:part:copy

month:
    {JAN|FEB|MAR|APR|MAY|JUN|JUL|AUG|SEP|OCT|NOV|DEC}
```

---

△ **Caution.** If recovery is needed, problems could occur if the time listed in the catalog is not the time when the online dump actually took place; if you specify the TIME option, make sure to enter the date and time completely and accurately.

---

# ADD RESOURCEMANAGER

Adds one or more recoverable resource managers to the resource manager directory.

Security: super user group member.

```
ADD {RESOURCEMANAGER | RM} {rm-name
                             {(rm-name [,rm-name]...)}

    ,OWNER {username | userid}

username:
    group-name.user-name

userid:
    (group-id,user-id)
```

# ADD TAPEMEDIA

Adds one or more tape volumes to the TMF catalog and, optionally, labels these volumes.

Security: super user group member.

```
ADD TAPEMEDIA {tape-volume
              {(tape-volume [,tape-volume]...)}

              {tape-labeling-options}
              {tape-status-options}

tape-labeling-options:
    ,LABEL ON [!]

    ,DRIVE [node.]device-name

    [,DENSITY density ]

    [,UNLOAD {ON | OFF}]

tape-status-options:
    ,LABEL OFF

    ,STATUS {ASSIGNED}
            {BAD}
            {RELEASED}
            {SCRATCH }
```

# ALTER AUDITDUMP

Changes the configured attribute values of an audit dump process.

Security: super user group member.

```
ALTER AUDITDUMP {MASTER | MAT }
                {AUX[ ILIARY]nn}

[,BLOCKSIZE {4 | 8 | 12 | 16 | 20 | 24 | 28 | 32 | 36
              40 | 44 | 48 | 52}]

[,RESET BLOCKSIZE

[,COPIES 1 [PARALLEL | SERIES]]
[,COPIES 2 [PARALLEL | SERIES]]
[,COPIES 3 SERIES
[,RESET COPIES

[,SYSTEM {node
          {(node, node)}}]
[,RESET SYSTEM

[,VERIFYTAPE {OFF | ON}]

[,UNLOAD {ON | OFF } ]

[, [MEDIUM {TAPE}
           {DISK}]
  [RESET MEDIUM]]

[,ADDDISKMEDIA {disk
                {(disk [,disk]...)}}]

[,DELDISKMEDIA {disk
                {(disk [,disk]...)}}]

disk:
    [node.]volume[.prefix] | node.volume.prefix
```

**Note.** The following ALTER AUDITDUMP parameters are only valid when MEDIUM is TAPE:

- BLOCKSIZE
- RESET BLOCKSIZE
- COPIES
- RESET COPIES
- SYSTEM
- RESET SYSTEM
- VERIFYTAPE
- UNLOAD

The ADDISKMEDIA and DELDISKMEDIA parameters are only valid when MEDIUM is DISK.

# ALTER AUDITTRAIL

Changes the configured attribute values of an audit trail and can execute while TMF is running.

Security: super user group member.

```
ALTER AUDITTRAIL {MASTER | MAT }
                  {AUX[ILIARY]nn}

[,AUDITDUMP {ON | OFF}]

[,ADDACTIVEVOL[S] {volume
                  {(volume [,volume]...) }
[,DELETEACTIVEVOL[S] {volume
                     {(volume [,volume]...) }

[,FILESPERVOLUME integer]
[,RESET FILESPERVOLUME ]

[,ADDOVERFLOWVOL[S] {volume
                    {(volume [,volume]...) }
[,DELETEOVERFLOWVOL[S] {volume
                       {(volume [,volume]...) }

[,OVERFLOWTHRESHOLD integer]
[,RESET OVERFLOWTHRESHOLD ]

[,ADDRESTOREVOL[S] {volume
                   {(volume [,volume]...) }
[,DELETERESTOREVOL[S] {volume
                      {(volume [,volume]...) }

[,MAXRETAINEDATFILES integer]
[,RESET MAXRETAINEDATFILES ]

[,BEGINTRANSDISABLE integer]
[,RESET BEGINTRANSDISABLE ]
```

- 
- △ **Caution.** Before you decrease the size of an audit trail, determine whether such a change might interrupt transaction processing. TMF does not alert you to the possibility that the audit trail could exceed its begin-transaction-disable threshold when you issue an ALTER AUDITTRAIL command to decrease the files per active volume or delete an active volume from the configuration.

If the audit trail exceeds its begin-transaction-disable threshold, new transactions cannot start. In this case, transaction processing is suspended until audit trail capacity drops below the threshold.

If the overflow threshold is exceeded as a result of reduced audit trail capacity, the first pinned audit trail is copied to the overflow volume.

Be careful not to lower the begin-transaction-disable or overflow threshold below your TMF system's current capacity levels; they will engage precisely in accordance with your request.

---

# ALTER BEGINTRANS

Changes various options related to the automatic disabling and reenabling of transaction processing, control of the TMF autoabort feature, and limits on resource managers and transaction branches used in heterogeneous transaction processing.

Security: super user group member.

```
ALTER BEGINTRANS

{[,TRANSCOUNTTHRESH (integer1, integer2)]
[,RESET TRANSCOUNTTHRESH                ]

[,TMFLIBMEMTHRESH (integer3, integer4) ]
[,RESET TMFLIBMEMTHRESH                ]

[,TMPMEMTHRESH (integer5, integer6)    ]
[,RESET TMPMEMTHRESH                    ]

[,TRANSPERCPU (integer7, integer8)      ]
[,RESET TRANSPERCPU                    ]

[,AUTOABORT timeout | 0                  ]
[,RESET AUTOABORT                        ]

[,RECRMOUNT integer9                     ]
[,RESET RECRMOUNT                       ]

[,RMOPENPERCPU integer10                 ]
[,RESET RMOPENPERCPU                    ]

[,BRANCHESPERM integer11                 ]
[,RESET BRANCHESPERM                    ]}
```

*timeout*:

```
integer { HOUR[S]      }
        { MINUTE[S]   }
        { SECOND[S]   }
```

- 
- △ **Caution.** When AUTOABORT is off, there is no mechanism to abort runaway transactions automatically; if a runaway transaction goes unnoticed, it can pin many audit-trail files, preventing them from being recycled and prohibiting the start of new transactions.
-



# ALTER CATALOG

Changes the configured attribute values of the catalog process that affect the management of online and audit dumps.

Security: super user group member.

```
ALTER CATALOG, attribute [,attribute]  
  
attribute:  
    [RELEASED {OFF | ON}]  
  
    [RETAINDEPTH integer]  
    [RESET RETAINDEPTH ]  
  
    [ROUNDROBIN {OFF | ON}]
```

# ALTER DATAVOLS

Changes the configured attribute values of a data volume.

Security: super user group member.

```
ALTER DATAVOL[S] {volume  
                  {(volume [,volume]...)}  
  
    [,RECOVERYMODE {ONLINE | ARCHIVE}]
```

△ **Caution.** TMF does not guarantee the availability of audit dumps required to perform volume recovery for data volumes configured with RECOVERYMODE ARCHIVE; those audit dumps must still exist in the TMF catalog when the recovery is attempted. Sufficient audit dumps are normally retained if any file on the volume has an online dump recorded in the catalog.

# ALTER DUMPS

Changes dump entries in the TMF catalog.

Security: super user group member.

```
ALTER DUMPS {file-set
              {(file-set [,file-set]...)}}

    [,INVALID {OFF | ON}  ]

    [,RELEASED {OFF | ON} ]

    [,SERIAL serial-number]

file-set:
    For online dumps:
        [[{volume | *}].]{subvolume | *}.]{file-id | *}
    For audit dumps:
        $ZTMFAT.ZTMFAT.{file-id | *}
```

# ALTER PROCESS

Changes the configured attribute values of various TMF process types.

Security: super user group member for all attributes except DEBUG; logon under super ID (user ID 255,255) for DEBUG attribute.

```

ALTER PROCESS {process-type
               {(process-type [, process-type]...) }
               { *
               }

    [,CPUS cpu-list]
    [,RESET CPUS    ]

    [,PRI integer]
    [,RESET PRI    ]

    [,SWAP volume-name]
    [,RESET SWAP    ]

    [,MAKESAVEABEND {ON | OFF}]

    [,DEBUG {OFF | ON}]

    [,INSPECT {ON | OFF}]

    [,PROGRAMFILE file-id]
    [,RESET PROGRAMFILE  ]

    [,TERM file-id]
    [,RESET TERM    ]

    [,EXTENDEDSEGSIZE (integer1, integer2)]
    [,RESET EXTENDEDSEGSIZE          ]

    [,PFSSIZE integer]
    [,RESET PFSSIZE  ]

process-type:
    AUDITDUMP
    AUDITRESTORE
    BACKOUT
    CATALOG
    COPYAUDITOVFLOW
    FILERECOVERY
    FILERECOVLIST
    ONLINEDUMP
    ONLINERESTORE
    TMFMON
    TMFMON2
    TMP
    VOLUMERECOVERY

cpu-list:
    {cpu-num
     {(cpu-num [, cpu-num]...) }
    }

```

## ALTER TAPEMEDIA

Changes the configured attribute values of tape volumes.

Security: super user group member.

```
ALTER TAPEMEDIA { tape-volume
                  { (tape-volume [ ,tape-volume]...) }

,STATUS { ASSIGNED }
        { BAD }
        { RELEASED }
        { SCRATCH }
```

## ALTER TMF

Changes the TMF configuration volume or switches the functions of the TMP's primary and backup processes.

Security: super user group member.

```
ALTER TMF, { CONFIGVOL volume }
           { SWITCHPROCESS TMP }
```

- △ **Caution.** Be very careful when entering the ALTER TMF command with the CONFIGVOL option, because this command deactivates the previous TMF configuration by changing the configuration volume. Because your TMF configuration is thereafter defined on the new configuration volume, it is a good precaution to ensure that this volume is a member of a mirrored disk pair.

To make the previous TMF configuration active again, you must enter another ALTER TMF command, giving the previous configuration volume name in *volume*.

## CANCEL OPERATIONS

Cancels a specific TMF file-recovery or dump-files operation.

Security: super user group member.

```
CANCEL OPERATION[S] operation-number
```

# CLOSE RESOURCEMANAGER

Forces closure of one or more recoverable resource managers that are currently open by a gateway process.

Security: super user group member.

```
CLOSE {RESOURCEMANAGER | RM} {rm-name
                               {(rm-name [ ,rm-name] ...)}}
```

# CMDVOL

Specifies the default volume and subvolume names for expanding unqualified file names other than those in OBEY commands.

Security: any user.

```
CMDVOL [volume
        [ [volume . ]subvolume ]
```

# DELETE CATALOG

Clears from the TMF catalog all online dump, audit dump, and media entries, and resets to 64 the counter used for assigning dump serial numbers.

Security: Logon under super ID (user ID 255,255).

```
DELETE CATALOG
```

- 
- △ **Caution.** Use extreme caution when entering the DELETE CATALOG command, because this command removes TMF's ability to recover your database from previously dumped files.
- 

# DELETE DATAVOLS

Removes one or more data volumes from the TMF configuration.

Security: super user group member.

```
DELETE DATAVOL[S] {volume
                    {(volume [ ,volume] ...)}}
```

[ ,ALLOWINCONSISTENCY]

- 
- △ **Caution.** After a data volume has been deleted with ALLOWINCONSISTENCY selected, volume recovery cannot return it to a consistent state.
-

## DELETE DUMPS

Removes dump entries from the TMF catalog, and attempts to delete the associated dump files from disk.

Security: super user group member.

```
DELETE DUMPS {file-set
              {(file-set [,file-set]...)}} [!]

      [,SERIAL serial-number]

file-set:
  For online dumps:
    [{volume | *}].{subvolume | *}}.{file-id | *}
  For audit dumps:
    $ZTMFAT.ZTMFAT.{file-id | *}
```

## DELETE RESOURCEMANAGER

Deletes one or more recoverable resource managers from the resource manager directory.

Security: super user group member.

```
DELETE {RESOURCEMANAGER | RM} {rm-name
                                {(rm-name [,rm-name]...)}}
```

## DELETE TAPEMEDIA

Removes tape-volume entries from the TMF catalog and also deletes the entries for dumps stored on the removed tape volumes.

Security: super user group member.

```
DELETE TAPEMEDIA {tape-volume
                  {(tape-volume [,tape-volume]...)}}
```

# DELETE TMF

Purges the entire TMF configuration, including volume and file recovery information for the database. Only tape media names are retained in the TMF catalog.

Security: Logon under super ID (user ID 255,255).

```
DELETE TMF
```

- △ **Caution.** Contact the Global Customer Support Center (GCSC) or your service provider before issuing this command.

Use extreme caution when entering the DELETE TMF command, because this command removes TMF's ability to recover database files. This command can leave the database in an inconsistent or damaged state, potentially affecting both local and remote nodes on which shared distributed transactions exist.

# DELETE TRANSACTION

Forcibly removes a transaction from the system.

Security: Logon under super ID (user ID 255,255).

```
DELETE TRANS[ACTION] transaction-id

transaction-id:
    [ node .                               ]
    [ node-number .                         ]
    [ node(tm-flags) .                     ] cpu.seq-num
    [ node-number(tm-flags) .             ]
    [ (tm-flags) .                         ]
```

- △ **Caution.** DELETE TRANSACTION is a very dangerous command, because it can result in a corrupted database. Use this command only with extreme caution. Alternative and safer methods for forcibly deleting a transaction are provided by the RESOLVE TRANSACTION and ABORT TRANSACTION commands.

# DISABLE AUDITDUMP

Suspends an audit dump process, preventing the dumping of audit-trail files.

Security: super user group member.

```
DISABLE AUDITDUMP {MASTER | MAT }
                  {AUX[ ILIARY ]nn}
[ ,ABRUPT]
```

- 
- △ **Caution.** Leaving audit dumps disabled for long durations can cause audit-trail files to accumulate. This accumulation, in turn, can potentially cause TMF to reach the OVERFLOWTHRESHOLD and BEGINTRANSDISABLE limits.
- 

# DISABLE BEGINTRANS

Suspends transaction processing throughout the TMF system.

Security: super user group member.

```
DISABLE BEGINTRANS
```

- 
- △ **Caution.** Suspending transaction processing can cause your transaction processing applications to fail; thus, you may have to stop your applications before issuing the DISABLE BEGINTRANS command. If you must suspend transaction processing, choose a time when the interruption will affect system users the least.
- 

# DISABLE DATAVOLS

Shuts down transaction processing (audited updates) for the specified data volume and places this volume in the disabled state so that it will not be automatically restarted by TMF.

Security: super user group member.

```
DISABLE DATAVOL[S] {volume
                    {(volume [ ,volume ]...)}
[ ,ABRUPT]
```

- 
- △ **Caution.** Leaving data volumes disabled for long durations can cause audit-trail files to accumulate. This accumulation, in turn, can potentially cause TMF to reach the OVERFLOWTHRESHOLD and BEGINTRANSDISABLE limits.
-



# DISPLAY OPERATIONS

Displays the Event Management Service (EMS) messages recorded for a specific TMF operation.

Security: read-access to the EMS log files.

```
DISPLAY OPERATION[S] operation-number

    [,LONG                                ]

    [,BEGINTIME date-time              [GMT]]

    [,LOGFILE filename                  ]
    [,COLLECTOR process-name           ]

date-time:
    {month[-]day[-]year, hr:min[:sec]}
    {day[-]month[-]year, hr:min[:sec]}

month:
    {JAN | FEB | MAR | APR | MAY | JUN | JUL | AUG | SEP | OCT | NOV | DEC}

filename:
    [volume.][subvolume.]file-id
```

---

**Note.** You can specify either the LOGFILE or the COLLECTOR option, but not both.

---

# DUMP FILES

Copies audited files to either tape or disk, producing online dumps. These online dumps might be used later if file recovery is initiated through a RECOVER FILES command.

Security: super user group member.

```

DUMP FILES {file-set
            {(file-set [,file-set]...)}

    [,BLOCKSIZE { 4 | 8 | 12 | 16 | 20 | 24 | 28 | 32 | 36
                  40 | 44 | 48 | 52}]

    [,COPIES { 1 | 2 } [PARALLEL] ]

    [,DISKMEDIA {disk
                 {(disk [,disk]...)} ]

    [,NOT {file-set
           {(file-set [,file-set]...)} ]

    [,START start-file ]

    [,TAPEMEDIA {tape-volume
                 {(tape-volume [,tape-volume]...)} ]

    [,SYSTEM {node
              {(node, node)} ]

    [,UNLOAD { ON | OFF} ]

    [,VERIFYTAPE { ON | OFF} ]

file-set:
    [[volume.]subvolume.]file-id

disk:
    remote-node.volume.prefix
    local-node.volume.prefix
    local-node.volume
    volume
    volume.prefix

start-file:
    {[[volume.]subvolume.]{file-id | *} }
    {[volume.]*.*}

```

- 
- △ **Caution.** Error messages generated by TMF and other subsystems during online dump operations can sometimes have a vital impact that is not always obvious. For example, a disk or tape parity or checksum error can prevent a file from being dumped. For this reason, you should check the individual messages and summary information displayed by the DUMP FILES command to ensure that you are aware of all such errors. In addition, you can use the INFO DUMPS command to verify that all the intended files were dumped.
- 

## ENABLE AUDITDUMP

Permits an audit dump process to resume execution after it has been suspended by the DISABLE AUDITDUMP command, allowing the process to dump audit-trail files once again.

Security: super user group member.

```
ENABLE AUDITDUMP {MASTER | MAT }
                  {AUX[ ILIARY]nn}
```

## ENABLE BEGINTRANS

Resumes transaction processing after it has been suspended by the DISABLE BEGINTRANS command.

Security: super user group member.

```
ENABLE BEGINTRANS
```

## ENABLE DATAVOLS

Reenables transaction processing (audited updates) for the specified data volumes, and makes them subject to automatic restart by TMF. If the data volumes specified are accessible, they are restarted immediately.

Security: super user group member.

```
ENABLE DATAVOL[S] {volume
                   { (volume [,volume],...)
                     * [,NOT volume]
                     * [,NOT (volume[,volume]...)] }
```

## ENV

Displays the current operating environment for your TMFCOM session.

Security: any user.

```
ENV
```

## EXIT

Ends your current TMFCOM session.

Security: any user.

```
EXIT
```

## FC

Permits you to display and repeat or modify the last command entered.

Security: any user.

```
FC [-linenum | linenum | string | "quoted-string"]
```

## HELP

Displays syntax information for TMFCOM commands.

Security: any user.

```
HELP [command          ]  
      [[command] object]  
      [* | ALL          ]
```

## HISTORY

Displays a history of commands entered during the current TMFCOM session, as maintained in the TMFCOM command history buffer.

Security: any user.

```
HISTORY [lines          ]  
         [RESET LAST   ]  
         [RESET ALL    ]
```

## INFO AUDITDUMP

Displays the configured attribute values of audit dump processes.

Security: any user.

```
INFO AUDITDUMP [MASTER | MAT
                [AUX[ILIARY]nn
                [(MASTER | MAT [ ,AUX[ILIARY]nn )]]
                [(AUX[ILIARY]nn [ ,MASTER | MAT )]]
                [(AUX[ILIARY]nn [ ,AUX[ILIARY]nn)]]
                [ *
                ]

[ ,OBEYFORM]
```

## INFO AUDITTRAIL

Displays the configured attribute values of audit trails.

Security: any user.

```
INFO AUDITTRAIL [MASTER | MAT
                 [AUX[ILIARY]nn
                 [(MASTER | MAT [ ,AUX[ILIARY]nn )]]
                 [(AUX[ILIARY]nn [ ,MASTER | MAT )]]
                 [(AUX[ILIARY]nn [ ,AUX[ILIARY]nn)]]
                 [ *
                 ]

[ ,OBEYFORM]
```

## INFO BEGINTRANS

Displays the configured attribute values for the BEGINTRANS object that disable and reenables transaction processing.

Security: any user.

```
INFO BEGINTRANS

[ ,OBEYFORM]
```

# INFO CATALOG

Displays the configured attribute values for the catalog process.

Security: any user.

```
INFO CATALOG
[ ,OBEYFORM]
```

# INFO DATAVOLS

Displays the configured attribute values of data volumes; also filters and then displays volumes based on selected configuration attributes.

Security: any user.

```
INFO DATAVOL[S] [volume
                  [(volume [ ,volume]...)]
                  [ *
                  ]

[ ,config-attrib [ ,config-attrib]...]

[ ,OBEYFORM
                  ]

[ ,SORT {BYAID | BYNAME}
                  ]

config-attrib:
  {AUDITTRAIL {MASTER | MAT }
   {AUX[ILIARY]nn}
  {RECOVERYMODE {ONLINE | ARCHIVE}}}
```

# INFO DUMPS

Displays online and audit dump entries in the TMF catalog.

Security: any user.

```

INFO DUMPS [file-set
            [(file-set [,file-set]...)]

    [,BRIEF | DETAIL | OBEYFORM      ]
    [,MEDIUM {TAPE | DISK}          ]
    [,SERIAL serial-number          ]
    [,SORT {BYFILE | BYSERIAL}       ]
    [,STATUS {ASSIGNED}
             {INVALID}
             {RELEASED}              ]
    [,TYPE {AUDITDUMP | ONLINEDUMP}]
    [,CATALOG [node.]volume.subvolume]
    [,NOT {file-set
           {(file-set [,file-set]...)}
           ]
    [,START start-file              ]

```

*file-set*:

For online dumps:

[[*volume.*]subvolume.]*file-id*

For audit dumps:

\$ZTMFAT.ZTMFAT.*file-id*

*start-file*:

{[[*volume.*]subvolume.]{*file-id* | \*}}

{[*volume.*]\*.\*}

# INFO PROCESS

Displays the configured attribute values for the various TMF process types.

Security: any user.

```
INFO PROCESS [process-type
              [(process-type [,process-type]...)]
              [ *
              ]

    [,OBEYFORM]

process-type:
    AUDITDUMP
    AUDITRESTORE
    BACKOUT
    CATALOG
    COPYAUDITOVFLOW
    FILERECOVERY
    FILERECOVLIST
    ONLINEDUMP
    ONLINERESTORE
    TMFMON
    TMFMON2
    TMP
    VOLUMERECOVERY
```

# INFO RESOURCEMANAGER

Displays the entries for recoverable resource managers currently registered in the resource manager directory.

Security: any user.

```
INFO {RESOURCEMANAGER | RM} [rm-name
                              [(rm-name [,rm-name]...)]
                              [ *
                              ]

    [,OBEYFORM]
```



## INFO TAPEMEDIA

Displays the configured attribute values of tape volumes.

Security: any user.

```
INFO TAPEMEDIA [ tape-volume
                  [ ( tape-volume [ , tape-volume ] ... ) ]
                  [ *
                  ]

    [ , STATUS { ASSIGNED }
              { BAD
              { RELEASED
              { SCRATCH }
              ]

    [ , OBEYFORM ]
```

## INFO TMF

Displays the configured attribute values of TMF.

Security: any user.

```
INFO TMF [ , { CNFGFORM | OBEYFORM } ]
```

## NEXT AUDITTRAIL

Changes audit-trail files for a specified audit trail; stops new audit records from being added to the current audit-trail file and establishes the next file in the audit trail sequence as the current audit-trail file.

Security: super user group member.

```
NEXT AUDITTRAIL { MASTER | MAT }
                 { AUX[ ILIARY ] nn }
```

## OBEY

Executes a series of commands entered in a command file.

Security: any user.

```
OBEY [ node. ] [ volume. ] [ subvolume. ] file-id
```

# OBEYVOL

Specifies the default volume and subvolume names for expanding an unqualified command file name used in an OBEY command.

Security: any user.

```
OBEYVOL [ [ node. ] volume. ] subvolume ]
```

# OPEN SERVER

Specifies the TMFSERVE process with which your TMFCOM process communicates to control TMF.

Security: any user.

```
OPEN SERVER { node
              { [ node. ] server }
      [ , CPU cpu-num      ]
      [ , PRI integer      ]
      [ , SWAP volume      ]
      [ , TERM [ node. ] term ]
```

# RECOVER FILES

Initiates file recovery: restoring a database to a consistent state.

Security: super user group member.

```
RECOVER FILES {file-set
               {(file-set [,file-set]...)}

[,NOT {file-set
       {(file-set [,file-set]...)}      ]

[,FROMARCHIVE ]

[,NOWAIT      ]

[,TOFIRSTPURGE]

[,SERIAL serial-number]

[,SYSTEM node]

[,TIME {month[-]day[-]year, hr:min[:sec]  [GMT]}
       {day[-]month[-]year, hr: min[:sec] [GMT]}}
```

[,TOMATPOSITION (*atseqno*, *rba*)]

[,UNLOAD {ON | OFF}]

[,TOPHYSVOL {*volume*}]

[,WHEREPHYSVOLIS {*volume*
 {(*volume* [,*volume*]...)} ]

[,RECOVERVDPPHASE1]

[,MAP NAME[S] {*old-filesset-list* TO *new-filesset*}
 {(*old-filesset-list* TO *new-filesset*
 [,*old-filesset-list* TO *new-filesset*]...)}]

*file-set*:  
 [[*volume*.]*subvolume*.]*file-id*

*month*:  
 {JAN | FEB | MAR | APR | MAY | JUN | JUL | AUG | SEP | OCT | NOV | DEC}

- 
- △ **Caution.** When you issue a RECOVER FILES command to initiate file recovery to the timestamp, and the time listed in the TMF catalog is not the time that the online dump actually took place, you risk destruction of your database because you will probably not recover all the data you need.

HP strongly recommends that you direct output from the RECOVER FILES command to a disk file or a printer by using the standard OUT option; doing so enables you to save the entire output for future reference. If you omit the OUT option, the output might be lost because the default is the terminal from which you issue the command, and the terminal has limited memory.

Before recovering a file that presently exists in the system, consider backing up that file to preserve its current content and status.

The TOPHYSVOL, WHEREPHYSVOLIS, and RECOVERVDPHASE1 options are intended for use with NonStop Storage Management Foundation (SMF) files only. Before you issue a RECOVER FILES command with any of these options, it is vital that you read [Options for the SMF Product](#) on page 3-212.

---

# RELOCATE DISKDUMPS

Changes the disk media locations specified in the TMF catalog.

Security: super user group member.

```
RELOCATE DISKDUMPS {serial-number
                    {(serial-number [,serial-number]...)}
                    {*}}

,OLDDMLOC diskmedia-loc, NEWDMLOC diskmedia-loc
```

# RESET AUDITTRAIL

Resets audit trail attributes to their default values, canceling the effects of any previous SET AUDITTRAIL commands.

Security: any user.

```
RESET AUDITTRAIL {attribute [,attribute]...}
                 {*}
```

*attribute*:

|                     |   |
|---------------------|---|
| [AUDITDUMP          | ] |
| [FILESIZE           | ] |
| [ACTIVEVOL[S]       | ] |
| [FILESPERVOLUME     | ] |
| [OVERFLOWVOL[S]     | ] |
| [OVERFLOWTHRESHOLD] |   |
| [RESTOREVOL[S]      | ] |
| [BEGINTRANSDISABLE] |   |

# RESET DATAVOLS

Resets data volume attributes to their default values, canceling the effects of any previous SET DATAVOLS commands.

Security: any user.

```
RESET DATAVOL[S] {attribute [,attribute]...}
                  {*}
```

*attribute*:

|                |
|----------------|
| [AUDITTRAIL]   |
| [RECOVERYMODE] |

# RESET DUMPS

Removes (clears) online and audit dump specifications, canceling the effects of previous SET DUMPS commands.

Security: any user.

```
RESET DUMPS {attribute [, attribute]...}  
            {  
              *  
            }
```

```
attribute:  
    [DISKMEDIA]  
    [SERIAL    ]  
    [TAPEMEDIA]  
    [TYPE      ]
```

---

**Note.** You must establish values for the SERIAL and TYPE entries and for either the DISKMEDIA or the TAPEMEDIA entry. Default values are not allowed for these entries. Therefore, if you enter a RESET DUMPS command for any of these entries, you must subsequently respecify them in a new ADD DUMPS or SET DUMPS command.

---

# RESET TAPEMEDIA

Removes certain tape volume attributes, or resets others to their defaults, canceling the effects of previous SET TAPEMEDIA commands.

Security: any user.

```
RESET TAPEMEDIA {attribute [, attribute]...}  
                {  
                  *  
                }
```

```
attribute:  
    [DENSITY  ]  
    [DRIVE    ]  
    [LABEL    ]  
    [STATUS   ]  
    [UNLOAD   ]
```

# RESOLVE TRANSACTION

Causes an unresolved distributed transaction to either commit or back out its changes. To fully resolve the transaction, this command must be issued on all nodes on which the transaction resides, except for the transaction's home node.

Security: super user group member.

```
RESOLVE TRANS[ACTION[S]]

    { transaction-id
      { (transaction-id [, transaction-id] ...) }
    { , STATE { ABORTED | COMMITTED } }

transaction-id:
    [ node . ]
    [ node-number . ]
    [ node ( tm-flags ) . ] cpu.seq-num
    [ node-number ( tm-flags ) . ]
    [ ( tm-flags ) . ]
```

---

△ **Caution.** If you are uncertain about the results or ramifications of the RESOLVE TRANSACTION command, contact the Global Customer Support Center (GCSC) or your service provider before issuing this command.

Using the RESOLVE TRANSACTION command to resolve a transaction that is in any state at the home node other than aborted, aborting, or committed, will result in an inconsistent database.

---

# SET AUDITTRAIL

Establishes attribute values for audit trails created by subsequent ADD AUDITTRAIL commands issued during the same TMFCOM session.

Security: any user.

```
SET AUDITTRAIL attribute [,attribute]...

attribute:
    [AUDITDUMP {ON | OFF}]

    [FILESIZE integer ]

    [ACTIVEVOL[S] {volume
                   {(volume [,volume] ...) } } ]

    [FILESPERVOLUME integer]

    [OVERFLOWVOL[S] {volume
                     {(volume [,volume]...) } } ]

    [OVERFLOWTHRESHOLD integer]

    [RESTOREVOL[S] {volume
                    {(volume [,volume]...) } } ]

    [BEGINTRANSDISABLE integer]
```

# SET DATAVOLS

Establishes attribute values for data volumes added by subsequent ADD DATAVOLS commands issued during the same TMFCOM session.

Security: any user.

```
SET DATAVOL[S] attribute [,attribute]...

attribute:
    AUDITTRAIL {MASTER | MAT }
               {AUX[ILIARY]nn}

    RECOVERYMODE {ONLINE | ARCHIVE}
```



# SET DUMPS

Establishes specifications for online and audit dump entries added to the catalog by subsequent ADD DUMPS commands issued during the same TMFCOM session.

Security: any user.

```
SET DUMPS attribute [,attribute]...

attribute:
    [DISKMEDIA disk ]
    [DISKMEDIA (disk [,disk]...)]

    [TAPEMEDIA tape-volume:part:copy]

    [SERIAL serial-number]

    [TYPE AUDITDUMP ]
    [TYPE ONLINEDUMP (master, data)]
```

# SET TAPEMEDIA

Establishes attribute values for tape volumes added by subsequent ADD TAPEMEDIA commands issued during the same TMFCOM session.

Security: any user.

```
SET TAPEMEDIA attribute [,attribute] ...

attribute:
    [DENSITY density]

    [DRIVE [node.]tape-drive]

    [LABEL {ON [!] | OFF}]

    [STATUS {ASSIGNED}
             {BAD}
             {RELEASED}
             {SCRATCH } ]

    [UNLOAD {ON | OFF}]
```

# SHOW AUDITTRAIL

Displays the attribute values established by SET AUDITTRAIL commands in the current TMFCOM session.

Security: any user.

```
SHOW AUDITTRAIL [attribute [,attribute]...]
                [ * ]
```

```
attribute:
    [AUDITDUMP           ]
    [FILESIZE            ]
    [ACTIVEVOL[S]        ]
    [FILESPELVOLUME      ]
    [OVERFLOWVOL[S]      ]
    [OVERFLOWTHRESHOLD   ]
    [RESTOREVOL[S]       ]
    [BEGINTRANSDISABLE   ]
```

# SHOW DATAVOLS

Displays the attribute values established by SET DATAVOLS commands in the current TMFCOM session.

Security: any user.

```
SHOW DATAVOL[S] [attribute [,attribute]...]
                 [ * ]
```

```
attribute:
    [AUDITTRAIL]
    [RECOVERYMODE]
```

# SHOW DUMPS

Displays the attribute values established by SET DUMPS commands in the current TMFCOM session.

Security: any user.

```
SHOW DUMPS [attribute [, attribute]...]
           [ * ]
```

```
attribute:
    [DISKMEDIA]
    [TAPEMEDIA]
    [SERIAL    ]
    [TYPE      ]
```

# SHOW TAPEMEDIA

Displays the attribute values established by SET TAPEMEDIA commands in the current TMFCOM session.

Security: any user.

```
SHOW TAPEMEDIA [attribute [, attribute]...]
               [ * ]
```

```
attribute:
    [DENSITY ]
    [DRIVE   ]
    [LABEL   ]
    [STATUS  ]
    [UNLOAD  ]
```

# START TMF

Starts TMF transaction processing, optionally displaying the EMS events for the start operation as this operation progresses.

Security: super user group member.

```
START TMF [ ,NOWAIT ]
          [ , {ENABLE | DISABLE} BEGINTRANS ]
```

- △ **Caution.** Verify the status of the stop operation after using a STOP TMF command with the NOWAIT option, but before entering the START command. Enter the DISPLAY OPERATIONS command to verify that the shutdown was complete; otherwise, damage to your database can occur during the restart. In the DISPLAY OPERATIONS command, use the operation number returned by the STOP TMF command-acceptance message or the appropriate date-time specification.

If you use the START TMF command with the NOWAIT option, also use DISPLAY OPERATIONS to verify the status of the start operation.

In addition, if multiple users are accessing your TMFSERVE process through multiple TMFCOM processes, be sure to communicate your intentions to stop or restart TMF.

# STATUS AUDITDUMP

Displays the current status of audit dump processes.

Security: any user.

```
STATUS AUDITDUMP [MASTER | MAT ]
                  [AUX[ ILIARY]nn ]
                  [(MASTER | MAT [,AUX[ ILIARY]nn]) ]
                  [(AUX[ ILIARY]nn [,MASTER | MAT]) ]
                  [(AUX[ ILIARY]nn [,AUX[ ILIARY]nn]) ]
                  [ * ]
```

# STATUS AUDITTRAIL

Displays the current audit trail activity.

Security: any user.

```
STATUS AUDITTRAIL [MASTER | MAT ]
                  [AUX[ ILIARY]nn ]
                  [(MASTER | MAT [,AUX[ ILIARY]nn]) ]
                  [(AUX[ ILIARY]nn [,MASTER | MAT]) ]
                  [(AUX[ ILIARY]nn [,AUX[ ILIARY]nn]) ]
                  [ * ]

[ ,DETAIL]
```

# STATUS BEGINTRANS

Displays the current status of TMF transaction processing, indicating whether this feature is enabled or disabled.

Security: any user.

```
STATUS BEGINTRANS
```

# STATUS CATALOG

Displays the current activity of the catalog process, indicating the availability of the TMF catalog.

Security: any user.

```
STATUS CATALOG
```

# STATUS DATAVOLS

Displays the current status of data volumes.

Security: any user.

```
STATUS DATAVOL[S] [volume
                    [(volume [,volume]...)]
                    [ *
                    ]

    [,STATE {ADDING
             {STARTING
             {STARTUPBEGUN
             {AWAITTRANSRESOLVE
             {QUEUEDFORRECOVERY
             {RECOVERING
             {STARTUPCOMPLETING
             {STARTED
             {NOTSTARTED
             {DOWN
             {DOWNCLEAN
             {DOWNDIRTY
             {SHUTTINGDOWN
             {DISABLING
             {DISABLED
             {DISABLEDCLEAN
             {DISABLEDDIRTY
             {UNCONFIGURED
             {WRONGDISKMOUNTED } ]

    [,AUDITTRAIL {MASTER | MAT }
                 {AUX[ILIARY]nn } ]

    [,RECOVERYMODE {ONLINE | ARCHIVE} ]

    [,SORT {BYATID | BYNAME}          ]
```

# STATUS OPERATIONS

Displays the status of TMF operations that were recently completed or are now in progress.

Security: any user.

```
STATUS OPERATION[S]
  [operation-number] ]
  [(operation-number [,operation-number]...)]
  [ * ]

  [,TYPE {AUDITRESTORE }
         {DELETETMF }
         {DUMPFILES }
         {RECOVERFILES }
         {STARTTMF }
         {STOPTMF }
         {VOLUMERECOVERY}
         {ANY } ]

  [,STATE {INPROGRESS }
          {FINISHED }
          {ALL } ]

  [,BEGINTIME date-time [GMT] ]

  [,SORT {BYNUMBER | BYTIME} ]

date-time:
  {month[-]day[-]year,hr:min[:sec]}
  {day[-]month[-]year,hr:min[:sec]}

month:
  {JAN | FEB | MAR | APR | MAY | JUN | JUL | AUG | SEP | OCT | NOV | DEC}
```

# STATUS RESOURCEMANAGER

Displays the current status of both recoverable and volatile resource managers.

Security: any user.

```
STATUS {RESOURCEMANAGER | RM} [rm-name
                                [rm-name [,rm-name]...]
                                [*]
                                ]

    [, STATE {CREATED
              {CLOSED
              {OPEN
              {RECOVERING
              {CRASHED
              {VOLATILE } ]

    [, PROCESS {process
                {(process [,process]...)
                {(cpu,pin)
                {((cpu,pin) [, (cpu,pin)]...) } ]

    [,DETAIL ]
```

# STATUS SERVER

Displays status and other information about the currently open TMFSERVE process.

Security: any user.

```
STATUS SERVER
```

# STATUS TMF

Displays the current activity of TMF.

Security: any user.

```
STATUS TMF
```

# STATUS TRANSACTIONS

Displays transaction identifiers and status.

Security: any user.

```
STATUS TRANS[ACTION[S]]

[transaction-id
[(transaction-id [,transaction-id]...) ]

[,PROCESS {process
            {process [,process]... }
            {cpu, pin}
            {(cpu, pin) [, (cpu, pin)]...} ]

[,STATE {ACTIVE
          {ABORTING
          {ABORTED
          {COMMITTED
          {ENDING
          {HUNG
          {PREPARED } ]

[,DETAIL | BRIEF ]

transaction-id:
[node. ]
[node-number. ]
[node(tm-flags). ]cpu.seq-num
[node-number(tm-flags). ]
[(tm-flags). ]
```

## STOP TMF

Shuts down TMF, terminating transaction processing and displaying the EMS events for the stop operation as it progresses.

Security: super user group member.

```
STOP TMF [,NOWAIT]

[,ABRUPT]
```

△ **Caution.** If multiple users are managing your TMF system, be sure to communicate your intentions to stop or restart TMF.



# VOLUME

Specifies the default volume and subvolume names for expanding all unqualified file names.

Security: any user.

```
VOLUME [ volume ]  
        [ [volume.] subvolume ]
```

## !

Re-executes a command previously entered in the current session.

Security: any user.

```
! [-linenum | linenum | string | "quoted-string"]
```

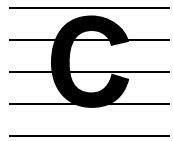
## ?

Displays a command previously entered in the current session.

Security: any user.

```
? [-linenum | linenum | string | "quoted-string"]
```





# TMF Configuration Limits and Defaults

The tables in this appendix summarize various limits and defaults for TMF.

- △ **Caution.** Some of the limits and defaults described in this appendix may have changed since this manual was published with RVU H06.03. If such changes were made, they will be described in *TMF Supplement for Limits and Defaults Changed after RVU H06.03*. If that Supplement currently exists, please read it for new information about limits and defaults.

The following types of limits and defaults are included:

| Topic  | Page                |
|--|---------------------|
| <a href="#">TMF Global Limits and Defaults</a>                 | <a href="#">C-1</a> |
| <a href="#">Audit Trail Limits and Defaults</a>                | <a href="#">C-2</a> |
| <a href="#">Transaction Limits and Defaults</a>                | <a href="#">C-2</a> |
| <a href="#">Audit Dump and Online Dump Limits and Defaults</a> | <a href="#">C-3</a> |
| <a href="#">Resource Manager Limits and Defaults</a>           | <a href="#">C-3</a> |
| <a href="#">Process Limits and Defaults</a>                    | <a href="#">C-3</a> |
| <a href="#">Default Program File Names for Processes</a>       | <a href="#">C-5</a> |

**Table C-1. TMF Global Limits and Defaults**

| Parameter  | Limits                                       | Default                           |
|--|--|-----------------------------------|
| Autoabort threshold  | 20 through 21474836 seconds (about 8 months) | 2 hours                           |
| Master audit trail capacity used by a transaction before it is automatically aborted | Always 45%                                   | 45%                               |
| TMFLIB memory threshold  | 1 through 100%                               | Disable at 90%<br>Reenable at 85% |
| TMP extended memory threshold  | 1 through 100%                               | Disable at 95%<br>Reenable at 90% |

**Table C-2. Audit Trail Limits and Defaults**

| <b>Parameter</b>                          | <b>Limits</b>   | <b>Default</b> |
|---|---|----------------|
| Active-audit volumes per audit trail      | 1 through 16  | None           |
| Audit dumping feature                     |   | On             |
| Audit-trail files per active-audit volume | 1 through 100. Active-audit volumes multiplied by files per active-audit volume must be 2 or greater. | 5              |
| Audit-trail file size                     | 1 through 2047 MB per audit-trail file, or the maximum disk size (if less than 2047 MB)               | 100 MB each    |
| Audit trails per node                     | 1 through 16 (1 master and 1 through 15 auxiliary)  | 1              |
| Begin-transaction-disable threshold       | 50% through 100%  | 90%            |
| Overflow-audit volumes per audit trail    | 0 through 16  | None           |
| Overflow threshold                        | 50% through 100%  | 80%            |
| Restore-audit volumes per audit trail     | 1 through 16  | None           |

**Table C-3. Transaction Limits and Defaults**

| <b>Parameter</b>  | <b>Limits</b>   | <b>Default</b>                      |
|---|---|-------------------------------------|
| Active transactions in a NonStop system   | 1 through 1,073,741,824                                   | Disable at 1600<br>Reenable at 1500 |
| Active transactions in a processor  | 1 through 1,073,741,824. The number must be a power of 2. | 1024                                |
| Active transactions per process (as specified by the TFILE transaction pseudo-file) | 1 through 1000  | 1                                   |
| Option to enable transactions when starting TMF                                     |   | Enabled                             |
| TMF operation number  | 1 through 100   | None                                |

**Table C-4. Audit Dump and Online Dump Limits and Defaults**

| Parameter  | Limits       | Default |
|--|--------------|---------|
| Number of parallel audit dump copies   | 1 through 2  | 1       |
| Number of serial audit dump copies   | 1 through 3  | 1       |
| Number of disk volumes that can be added to an audit dump configuration when audit dumping is directed to disk media | 1 through 16 | None    |
| Number of tape or disk volumes in a DUMP FILES command   | 1 through 16 | None    |

**Table C-5. Resource Manager Limits and Defaults**

| Parameter   | Limits   | Default |
|---|--|---------|
| Resource manager name   | Maximum length: 128 characters                 | None    |
| Maximum number of registered recoverable resource managers per node   | 16 through 16384<br>(must be a multiple of 16) | 256     |
| Maximum number of open resource managers per processor  | 16 through 1024<br>(must be a multiple of 16)  | 128     |
| Maximum number of concurrently pushed (exported) or pulled (imported) transaction branches per resource manager | 16 through 1024<br>(must be a multiple of 16)  | 128     |

**Table C-6. Process Limits and Defaults** (page 1 of 3)

| Parameter                   | Limits                                  | Default                                      |
|-----------------------------|---|--|
| Audit dump process type     |   |  |
| Number of processes         |   | 0 if not needed; 1 per audit trail as needed |
| Extended segment size       | 0.5 through 128 MB<br>(127.5 MB actual) | 0.5 MB                                       |
| Audit overflow process type |   |  |
| Number of processes         |   | 0 if not needed; 1 per audit trail as needed |
| Extended segment size       | 0.5 through 128 MB<br>(127.5 MB actual) | 0.5 MB                                       |

**Table C-6. Process Limits and Defaults** (page 2 of 3)

| <b>Parameter</b>                | <b>Limits</b>                           | <b>Default</b>                          |
|---------------------------------|---|---|
| Audit restore process type      |   |   |
| Number of processes             |   | 0 if not needed;<br>1 or more as needed |
| Extended segment size           | 0.5 through 128 MB<br>(127.5 MB actual) | 0.5 MB                                  |
| Backout process type            |   |   |
| Number of processes             |   | Always 4                                |
| Extended segment size           | 5 through 128 MB<br>(127.5 MB actual)   | 5 MB                                    |
| Catalog process type            |   |   |
| Number of processes             |   | 1 primary and<br>1 backup process       |
| File recovery process type      |   |   |
| Number of processes             |   | 0 if not needed;<br>1 or more as needed |
| Extended segment size           | 40 through 128 MB<br>(127.5 MB actual)  | 40 MB                                   |
| File recovery list process type |   |   |
| Number of processes             |   | 0 if not needed;<br>1 or more as needed |
| Extended segment size           | 1 through 128 MB<br>(127.5 actual)      | 1 MB                                    |
| Online dump process type        |   |   |
| Number of processes             |   | 0 if not needed;<br>1 or more as needed |
| Extended segment size           | 0.5 through 128 MB<br>(127.5 MB actual) | 0.5 MB                                  |
| Online restore process type     |   |   |
| Number of processes             |   | 0 if not needed;<br>1 or more as needed |
| Extended segment size           | 0.5 through 128 MB<br>(127.5 MB actual) | 0.5 MB                                  |
| TMFMON process type             |   |   |
| Number of processes             |   | 1 per processor                         |
| Extended segment size           | 12 through 128 MB<br>(127.5 actual)     | 12 MB                                   |

**Table C-6. Process Limits and Defaults** (page 3 of 3)

| Parameter  | Limits   | Default                              |
|--|--|--------------------------------------|
| TMFMON2 process type   |  |                                      |
| Number of processes  |  | 1 per processor                      |
| Extended segment size  | Always 0.25 MB   | 0.25 MB                              |
| TMP process type   |  |                                      |
| Number of processes  |  | 1 primary and<br>1 backup process    |
| Extended segment size  | See discussion, <a href="#">EXTENDEDSEGIZE Option Considerations for the TMP</a> on page 3-87.         |                                      |
| Volume recovery process type   |  |                                      |
| Number of processes  | 0 through 8. (If the system has fewer than 8 processors, the upper limit is the number of processors.) | 0 if not needed; 1 or more as needed |
| Extended segment size  | 40 through 128 MB (127.5 actual)   | 40 MB                                |
| Number of data volumes assigned to each volume recovery process's work queue | 1 through 100  | As many as available (up to 100)     |
| Delay between the TMP's attempts to restart a failed TMF process             | 30 seconds through 5 minutes   | 30 seconds (at first attempt)        |

**Table C-7. Default Program File Names for Processes**

| TMF Process        | Default Program File Name |
|--------------------|---------------------------|
| Audit dump         | \$SYSTEM.SYSnn.TMFDR      |
| Audit overflow     | \$SYSTEM.SYSnn.TMFDR      |
| Audit restore      | \$SYSTEM.SYSnn.TMFDR      |
| Backout            | \$SYSTEM.SYSnn.TMFBOUT    |
| Catalog            | \$SYSTEM.SYSnn.TMFCTLG    |
| File recovery      | \$SYSTEM.SYSnn.TMFFRCV    |
| File recovery list | \$SYSTEM.SYSnn.TMFFRLS    |
| Online dump        | \$SYSTEM.SYSnn.TMFDR      |
| Online restore     | \$SYSTEM.SYSnn.TMFDR      |
| TMFMON2            | \$SYSTEM.SYSnn.TMFMON2    |
| TMP                | \$SYSTEM.SYSnn.TMFTMP     |
| Volume recovery    | \$SYSTEM.SYSnn.TMFVRCV    |





# Index

## A

### ABORT TRANSACTION

command [3-10/3-13](#), [B-2](#)

Aborted transactions [3-298](#)

Aborting transactions [3-10/3-13](#), [3-229](#), [3-297](#)

### ABRUPT parameter

DISABLE AUDITDUMP

command [3-116](#), [3-117](#)

DISABLE DATAVOLS

command [3-119/3-120](#)

STOP TMF command [3-305/3-306](#)

Active audit volumes [3-16](#), [3-53](#), [3-219](#), [3-233](#), [3-247](#)

See also Audit volumes

Active transactions [3-12](#), [3-297](#)

### ACTIVEVOLS parameter

ADD AUDITTRAIL command [3-15](#)

RESET AUDITTRAIL command [3-219](#)

SET AUDITTRAIL command [3-233](#)

SHOW AUDITTRAIL command [3-247](#)

ADD AUDITTRAIL command [3-14/3-19](#), [B-3](#)

ADD DATAVOLS command [3-20/3-23](#), [B-3](#)

ADD DUMPS command [3-24/3-34](#), [B-4](#)

### ADD RESOURCEMANAGER

command [3-36/3-38](#), [B-6](#)

ADD TAPEMEDIA command [3-39/3-42](#), [4-1/4-2](#), [B-6](#)

ADDACTIVEVOLS parameter, ALTER AUDITTRAIL command [3-52/3-53](#)

ADDDISKMEDIA parameter, ALTER AUDITDUMP command [3-46/3-48](#)

Adding audit dumps [3-25/3-27](#)

Adding audit trails [3-14/3-19](#)

Adding data volume state [3-271](#)

Adding data volumes [3-20/3-23](#)

Adding dumps [3-24/3-34](#)

Adding online dumps [3-28/3-31](#)

Adding resource managers [3-36/3-38](#)

Adding tape volumes [3-39/3-42](#)

ADDOVERFLOWVOLS parameter, ALTER AUDITTRAIL command [3-54](#)

ADDRESTOREVOLS parameter, ALTER AUDITTRAIL command [3-54](#)

ALL parameter, HELP command [3-145](#)

ALLOWINCONSISTENCY parameter, DELETE DATAVOLS command [3-103](#)

ALTER AUDITDUMP command [3-43/3-50](#), [4-4](#), [B-7](#)

ALTER AUDITTRAIL command [3-51/3-60](#), [B-8](#)

ALTER BEGINTRANS command [3-61/3-66](#), [B-10](#)

ALTER CATALOG command [3-67/3-70](#), [B-11](#)

ALTER DATAVOLS command [3-71/3-72](#), [B-11](#)

ALTER DUMPS command [3-73/3-74](#), [B-12](#)

ALTER PROCESS command [3-74/3-91](#), [B-12](#)

ALTER TAPEMEDIA command [3-92/3-93](#), [B-14](#)

ALTER TAPEMOUNT command, in MEDIACOM subsystem [4-2](#)

ALTER TMF command [3-94/3-95](#), [B-14](#)

ARCHIVE recovery mode [3-21](#), [3-22](#), [3-71](#), [3-236](#)

ASSIGNED tape volume status [3-40](#), [3-92](#), [3-110](#), [3-168](#), [3-179](#), [3-244](#)

### Audit dump process

See also Audit dumps

changing attributes of [3-43/3-50](#)

disabling [3-116/3-117](#)

displaying attribute values of [3-150/3-151](#)

displaying status of [3-259/3-260](#)

enabling [3-135](#)

purpose of [3-172](#)

resuming [3-135](#)

suspending [3-116/3-117](#)

## Audit dumps

See also Audit dump process

adding [3-25/3-27](#)

disk volumes for [3-25](#), [3-46](#), [3-49](#),  
[3-127](#), [3-224](#), [3-238/3-240](#), [3-252](#)

file recovery from [3-201/3-214](#)

performing [3-14](#), [3-22](#), [3-51](#), [3-232](#)

serial numbers of [3-25](#), [3-106](#)

tape volumes for [3-26](#), [3-44](#), [3-46](#),  
[3-224](#), [3-240](#), [3-252](#)

verifying [3-45](#)

## Audit restore process [3-172](#)

### Audit trails

adding [3-14/3-19](#)

changing attributes of [3-51/3-60](#)

default attribute values for [3-219/3-221](#)

disk volumes for [3-15](#), [3-52](#), [3-219](#),  
[3-233](#)

displaying attribute values  
of [3-152/3-155](#), [3-247/3-249](#)

displaying status of [3-261/3-266](#)

identifying [3-20](#), [3-25](#), [3-43](#), [3-116](#)

number of files on [3-247](#)

resetting attributes of [3-219/3-221](#)

restriction on usage of [3-22](#)

setting attributes of [3-232/3-235](#)

size of files for [3-15](#), [3-219](#), [3-233](#),  
[3-247](#)

switching files for [3-189](#)

### Audit volumes

See also Active audit volumes

disk volumes for [3-247](#)

number of files on [3-15](#), [3-16](#), [3-53](#),  
[3-58](#), [3-219](#), [3-233](#)

overflow [3-16](#), [3-18](#), [3-54](#), [3-233](#), [3-247](#)

restore [3-17](#), [3-18](#), [3-54](#), [3-220](#), [3-234](#),  
[3-248](#)

### AUDITDUMP object

ALTER command [3-43/3-50](#), [4-4](#), [B-7](#)

DISABLE command [3-116/3-117](#), [B-18](#)

ENABLE command [3-135](#), [B-21](#)

### AUDITDUMP object (continued)

INFO command [3-150/3-151](#), [B-23](#)

STATUS command [3-259/3-260](#), [B-38](#)

### AUDITDUMP parameter

ADD AUDITTRAIL command [3-14](#)

ALTER AUDITTRAIL command [3-51](#),  
[3-56](#)

RESET AUDITTRAIL command [3-219](#)

SET AUDITTRAIL command [3-232](#)

SHOW AUDITTRAIL command [3-247](#)

### AUDITDUMP process type [3-80](#)

### AUDITRESTORE process type [3-80](#)

### AUDITTRAIL object

ADD command [3-14/3-19](#), [B-3](#)

ALTER command [3-51/3-60](#), [B-8](#)

INFO command [3-152/3-155](#), [B-23](#)

NEXT command [3-189](#), [B-27](#)

RESET command [3-219/3-221](#), [B-31](#)

SET command [3-232/3-235](#), [B-34](#)

SHOW command [3-247/3-249](#), [B-36](#)

STATUS command [3-261/3-266](#), [B-38](#)

### AUDITTRAIL parameter

ADD DATAVOLS command [3-20](#)

RESET DATAVOLS command [3-222](#)

SET DATAVOLS command [3-236](#)

SHOW DATAVOLS command [3-250](#)

Audit-trail files, retaining on restore audit  
volumes [3-54](#), [3-56/3-58](#)

audit-trail-file parameter, ADD DUMPS  
command [3-25](#)

Autoabort feature [3-63](#)

AUTOABORT parameter, ALTER

BEGINTRANS command [3-63](#)

### Auxiliary audit trails

adding [3-14/3-19](#)

identifying [3-20](#), [3-43](#), [3-51](#), [3-116](#)

### AUXILIARY parameter

ADD AUDITTRAIL command [3-14](#)

ALTER AUDITDUMP command [3-43](#),  
[3-135](#)

ALTER AUDITTRAIL command [3-51](#)

AUXILIARY parameter (continued)  
 DISABLE AUDITDUMP  
 command [3-116](#)  
 INFO AUDITDUMP command [3-150](#)  
 INFO AUDITTRAIL command [3-152](#)  
 NEXT AUDITTRAIL command [3-189](#)  
 STATUS AUDITDUMP  
 command [3-259](#)  
 STATUS AUDITTRAIL command [3-261](#)  
 STATUS DATAVOLS command [3-273](#)  
 AVOIDHANGING parameter, ABORT  
 TRANSACTION command [3-11/3-12](#)  
 Awaittransresolve data volume state [3-271](#)  
 AWAITTRANSRESOLVE state [3-274](#)

## B

Backout process [3-172](#)  
 BACKOUT process type [3-80](#)  
 BAD tape volume status [3-41](#), [3-92](#), [3-179](#),  
[3-244](#)  
 Beginning file recovery [3-201/3-214](#)  
 BEGINTIME parameter  
 DISPLAY OPERATIONS  
 command [3-121](#)  
 STATUS OPERATIONS  
 command [3-277](#)  
 BEGINTRANS object  
 ALTER command [3-61/3-66](#), [B-10](#)  
 DISABLE command [3-118](#), [B-18](#)  
 ENABLE command [3-136](#), [B-21](#)  
 INFO command [3-156/3-158](#), [B-23](#)  
 STATUS command [3-268](#), [B-38](#)  
 BEGINTRANS option, displaying status  
 of [3-268](#)  
 BEGINTRANSACTION system library  
 procedure [3-118](#)  
 BEGINTRANSDISABLE parameter  
 ADD AUDITTRAIL command [3-17](#)  
 ALTER AUDITTRAIL command [3-55](#)  
 RESET AUDITTRAIL command [3-220](#)

BEGINTRANSDISABLE  
 parameter (continued)  
 SET AUDITTRAIL command [3-234](#)  
 SHOW AUDITTRAIL command [3-248](#)  
 BLOCKSIZE parameter  
 ALTER AUDITDUMP command [3-44](#),  
[4-4](#)  
 DUMP FILES command [3-126](#), [4-4](#)  
 BRANCHESPERM parameter, ALTER  
 BEGINTRANS command [3-64](#)  
 BREAK key, interrupting TMFCOM with [2-3](#)  
 BRIEF parameter  
 INFO DUMPS command [3-166](#), [3-167](#)  
 STATUS TRANSACTIONS  
 command [3-298](#)

## C

CANCEL OPERATIONS  
 command [3-96/3-97](#), [B-14](#)  
 Canceling operations [3-96/3-97](#)  
 Catalog  
 See also Catalog object  
 clearing entries from [3-102](#)  
 contents of [3-69](#)  
 CATALOG object  
 ALTER command [3-67/3-70](#), [B-11](#)  
 DELETE command [3-102](#), [B-15](#)  
 INFO command [3-159/3-160](#), [B-24](#)  
 STATUS command [3-269](#), [B-38](#)  
 CATALOG parameter, INFO DUMPS  
 command [3-168](#)  
 Catalog process  
 changing attributes of [3-67/3-70](#)  
 displaying attribute values  
 of [3-159/3-160](#)  
 displaying status of [3-269](#)  
 purpose of [3-172](#)  
 CATALOG process type [3-80](#)  
 Changing audit trail attributes [3-51/3-60](#)  
 Changing audit-dump process  
 attributes [3-43/3-50](#)

Changing BEGINTRANS options [3-61/3-66](#)  
 Changing catalog process attributes [3-67/3-70](#)  
 Changing data volume attributes [3-71/3-72](#)  
 Changing dump entries [3-73/3-74](#)  
 Changing extended segment size [3-85](#)  
 Changing process attributes [3-75/3-91](#)  
 Changing tape volume attributes [3-92/3-93](#)  
 Changing the configuration volume [3-94](#)  
 Clearing catalog entries [3-102](#)  
 CLOSE RESOURCEMANAGER command [3-98/3-99](#), [B-15](#)  
 Closed resource managers [3-283](#)  
 Closing resource managers [3-98/3-99](#)  
 CMDVOL command [3-100/3-101](#), [3-193](#), [3-310](#), [B-15](#)  
 CNFGFORM parameter, INFO TMF command [3-182](#)  
 COLLECTOR parameter, DISPLAY OPERATIONS command [3-122](#)  
 Command files  
     executing commands in [3-190/3-191](#)  
     formatting output for [3-150](#), [3-152](#), [3-156](#), [3-159](#), [3-161](#), [3-166](#), [3-173](#), [3-176](#), [3-179](#), [3-182](#)  
     purpose of [2-5](#), [3-141](#)  
 Command interface [1-4](#)  
 Command option, TMFCOM run command [2-2](#)  
 command parameter, HELP command [3-145](#)  
 Commands  
     See also Individual command names  
     alternate output for [2-8/2-9](#)  
     comments in [2-6](#)  
     continuing [2-7](#)  
     correcting [3-142/3-144](#)  
     CPU numbers in [3-9](#)  
     device names in [3-8](#)  
     disk file names in [3-7](#)  
     examples overview [3-6](#)  
     file-ids in [3-7](#)

Commands (continued)  
     help in entering [2-10/2-11](#)  
     interactively entering [2-2/2-4](#)  
     listing all [2-12](#)  
     multiple [2-7/2-8](#)  
     named process identifiers in [3-9](#)  
     node names in [3-7](#)  
     nondisk device file names in [3-8](#)  
     noninteractively entering [2-4](#)  
     output from [3-6](#)  
     partially qualified file names in [2-9](#)  
     pins (process identification numbers) in [3-9](#)  
     reading from a command file [2-5](#)  
     security restrictions overview [3-2](#)  
     state requirement for entry of [3-2](#)  
     subvolume names in [3-7](#)  
     summary of [B-1/B-43](#)  
     syntax overview of [3-2](#)  
     unnamed process identifiers in [3-9](#)  
     usage guidelines overview of [3-6](#)  
     volume names in [3-7](#)  
 Committed transactions [3-229](#), [3-298](#)  
 CONFIG parameter, ALTER TMF command [3-94](#)  
 Configuration volume, changing [3-94](#)  
 config-attrib, INFO DATAVOLS command [3-161](#)  
 COPIES parameter  
     ALTER AUDITDUMP command [3-44/3-45](#), [4-4](#)  
     DUMP FILES command [3-127](#), [4-4](#)  
 COPYAUDITOVFLOW process type [3-81](#)  
 Correcting commands [3-142/3-144](#)  
 CPU numbers [3-9](#)  
     See also cpu parameter  
 cpu parameter  
     ABORT TRANSACTION command [3-11](#)  
     DELETE TRANSACTION command [3-113](#)

cpu parameter (continued)  
RESOLVE TRANSACTION  
command [3-229](#)  
STATUS TRANSACTIONS  
command [3-297](#)  
CPU parameter, OPEN SERVER  
command [3-194](#)  
CPUS parameter, ALTER PROCESS  
command [3-77](#), [3-84](#)  
Crashed resource managers [3-283](#)  
crash-count parameter, RESOLVE  
TRANSACTION command [3-113](#)  
Created resource managers [3-283](#)

## D

Data errors [3-11](#)

Data volumes

- adding [3-20/3-23](#)
- changing attributes of [3-71/3-72](#)
- default attribute values for [3-222/3-223](#)
- deleting [3-103/3-104](#)
- disabling [3-119/3-120](#)
- displaying attribute values  
of [3-161/3-163](#), [3-250/3-251](#)
- displaying status of [3-270/3-275](#)
- enabling [3-137/3-138](#)
- resetting attributes of [3-222/3-223](#)
- setting attributes of [3-236/3-237](#)
- states of [3-270/3-273](#)

DATAVOLS object

- ADD command [3-20/3-23](#), [B-3](#)
- ALTER command [3-71/3-72](#), [B-11](#)
- DELETE command [3-103/3-104](#), [B-15](#)
- DISABLE command [3-119/3-120](#), [B-18](#)
- ENABLE command [3-137/3-138](#), [B-21](#)
- INFO command [3-161/3-163](#), [B-24](#)
- RESET command [3-222/3-223](#), [B-31](#)
- SET command [3-236/3-237](#), [B-34](#)
- SHOW command [3-250/3-251](#), [B-36](#)
- STATUS command [3-270/3-275](#), [B-39](#)

DEBUG debugger [3-78](#)

DEBUG parameter, ALTER PROCESS  
command [3-78](#)

Default subvolume names,  
specifying [3-100/3-101](#), [3-192/3-193](#),  
[3-309/3-310](#)

Default volume names,  
specifying [3-100/3-101](#), [3-192/3-193](#),  
[3-309/3-310](#)

DELDISKMEDIA parameter, ALTER  
AUDITDUMP command [3-48](#)

DELETE CATALOG command [3-102](#), [B-15](#)

DELETE DATAVOLS  
command [3-103/3-104](#), [B-15](#)

DELETE DUMPS command [3-105/3-107](#),  
[B-16](#)

DELETE RESOURCEMANAGER  
command [3-108/3-109](#), [B-16](#)

DELETE TAPEMEDIA command [3-110](#),  
[B-16](#)

DELETE TMF command [3-111/3-112](#), [B-17](#)

DELETE TRANSACTION  
command [3-113/3-115](#), [B-17](#)

DELETEACTIVEVOLS parameter, ALTER  
AUDITTRAIL command [3-52/3-53](#)

DELETEOVERFLOWVOLS parameter,  
ALTER AUDITTRAIL command [3-54](#)

DELETERESTOREVOLS parameter,  
ALTER AUDITTRAIL command [3-54](#)

Deleting data volumes [3-103/3-104](#)

Deleting dumps [3-105/3-107](#)

Deleting resource managers [3-108/3-109](#)

Deleting tape volumes [3-110](#)

Deleting TMF configuration [3-111/3-112](#)

Deleting transactions [3-113/3-115](#)

DENSITY parameter

- ADD TAPEMEDIA command [3-40](#),  
[4-1/4-2](#)

- RESET TAPEMEDIA command [3-226](#)

- SET TAPEMEDIA command [3-243](#),  
[4-1/4-2](#)

- SHOW TAPEMEDIA command [3-254](#)



- DETAIL parameter
  - INFO DUMPS command [3-166](#)
  - STATUS AUDITTRAIL command [3-261](#)
  - STATUS RESOURCEMANAGER command [3-284](#)
  - STATUS TRANSACTIONS command [3-298](#)
- Device names [3-8](#)
- Direct file names [3-212](#)
- DISABLE AUDITDUMP command [3-116/3-117](#), [B-18](#)
- DISABLE BEGINTRANS command [3-118](#), [3-136](#), [B-18](#)
- DISABLE DATAVOLS command [3-119/3-120](#), [B-18](#)
- DISABLE parameter, START TMF command [3-256](#)
- Disabled data volume state [3-272](#)
- Disabledclean data volume state [3-272](#)
- Disableddirty data volume state [3-273](#)
- Disabling audit dump processes [3-116/3-117](#)
- Disabling data volume state [3-272](#)
- Disabling data volumes [3-119/3-120](#)
- Disabling transaction processing [3-118](#)
- DISKDUMPS object, RELOCATE command [3-217/3-218](#), [B-31](#)
- DISKMEDIA parameter
  - ADD DUMPS command [3-25](#), [3-29](#), [3-33](#)
  - DUMP FILES command [3-127/3-128](#), [3-130/3-132](#), [3-134](#)
  - RESET DUMPS command [3-224](#)
  - SET DUMPS command [3-238/3-240](#)
  - SHOW DUMPS command [3-252](#)
- DISPLAY OPERATIONS command [3-121/3-124](#), [B-19](#)
- Displaying audit dump process attribute values [3-150/3-151](#)
- Displaying audit dump process status [3-259/3-260](#)
- Displaying audit trail attribute values [3-152/3-155](#), [3-247/3-249](#)
- Displaying audit trail status [3-261/3-266](#)
- Displaying BEGINTRANS attribute values [3-156/3-158](#)
- Displaying BEGINTRANS option status [3-268](#)
- Displaying catalog process attribute values [3-159/3-160](#)
- Displaying catalog process status [3-269](#)
- Displaying data volume attribute values [3-161/3-163](#), [3-250/3-251](#)
- Displaying data volume status [3-270/3-275](#)
- Displaying default volume and subvolume [B-22](#)
- Displaying dump attribute values [3-252/3-253](#)
- Displaying dump entries [3-165/3-171](#)
- Displaying history of commands [3-147/3-149](#)
- Displaying operations [3-121/3-124](#)
- Displaying operations status [3-276/3-281](#)
- Displaying previously entered commands [3-314/3-315](#)
- Displaying process attributes [3-172/3-175](#)
- Displaying resource manager attributes [3-176/3-178](#)
- Displaying resource manager status [3-282/3-288](#)
- Displaying session environment [3-139/3-140](#)
- Displaying tape volume attributes [3-179/3-180](#), [3-254/3-255](#)
- Displaying TMF attributes [3-182/3-185](#)
- Displaying TMF status [3-291/3-295](#)
- Displaying TMFSERVE server status [3-289/3-290](#)
- Displaying transaction status [3-296/3-304](#)
- Distributed transactions, resolving [3-228/3-231](#), [3-257](#)
- Down data volume state [3-272](#)
- Downclean data volume state [3-272](#)
- Downdirty data volume state [3-272](#)
- DRIVE parameter
  - ADD TAPEMEDIA command [3-40](#)
  - RESET TAPEMEDIA command [3-226](#)

DRIVE parameter (continued)

SET TAPEMEDIA command [3-243](#)

SHOW TAPEMEDIA command [3-254](#)

DUMP FILES command [3-125/3-134](#), [4-4](#), [B-20](#)

Dumps

adding [3-24/3-34](#)

audit, performing [3-14](#), [3-22](#), [3-232](#)

changing catalog entries for [3-73/3-74](#)

clearing specifications for [3-224/3-225](#)

deleting [3-105/3-107](#)

displaying attribute values of [3-252/3-253](#)

displaying catalog entries for [3-165/3-171](#)

identifying files for [3-105](#)

online, performing [3-22](#)

performing [3-51](#), [3-125/3-134](#), [3-201/3-214](#)

removing specifications for [3-224/3-225](#)

setting specifications for [3-238/3-242](#)

DUMPS object

ADD command [3-24/3-34](#), [B-4](#)

ALTER command [3-73/3-74](#), [B-12](#)

DELETE command [3-105/3-107](#), [B-16](#)

INFO command [3-165/3-171](#), [B-25](#)

RESET command [3-224/3-225](#), [B-32](#)

SET command [3-238/3-242](#), [B-35](#)

SHOW command [3-252/3-253](#), [B-36](#)

## E

EMS collector process [3-122](#)

EMS logfile [3-122](#)

EMS messages [3-121/3-122](#), [3-256](#), [3-305](#)

ENABLE AUDITDUMP command [3-135](#), [B-21](#)

ENABLE BEGINTRANS command [3-118](#), [3-136](#), [B-21](#)

ENABLE DATAVOLS command [3-137/3-138](#), [B-21](#)

ENABLE parameter, START TMF command [3-256](#)

Enabling audit dump processing [3-135](#)

Enabling data volumes [3-137/3-138](#)

Enabling transaction processing [3-136](#)

Ending TMFCOM sessions [3-141](#)

Ending transactions [3-298](#)

ENV command [3-139/3-140](#), [B-22](#)

Error codes in messages [A-2](#)

Error identifiers [A-3](#)

Error information capture [A-4](#)

Errors, data [3-11](#)

Executing commands in command files [3-190/3-191](#)

Executing previously entered commands [3-311/3-313](#)

EXIT command [3-141](#), [B-22](#)

Extended segment size

TMFMON [3-88](#)

TMP [3-87](#)

EXTENDEDSEGSIZE parameter, ALTER PROCESS command [3-79](#)

## F

Failure data capture [A-4](#)

FC command [3-142/3-144](#), [B-22](#)

File names [3-6](#)

direct [3-212](#)

logical [3-212](#)

physical [3-212](#)

File names in messages [A-3](#)

File recovery list process [3-172](#)

File recovery process [3-73](#), [3-172](#)

File recovery, initiating [3-201/3-214](#)

FILERECOVERY process type [3-81](#)

FILERECOVLIST process type [3-81](#)

FILES object

DUMP command [3-125/3-134](#), [4-4](#), [B-20](#)

RECOVER command [3-201](#), [B-29](#)

## FILESIZE parameter

ADD AUDITTRAIL command [3-15](#),  
[3-58](#)

RESET AUDITTRAIL command [3-219](#)

SET AUDITTRAIL command [3-233](#)

SHOW AUDITTRAIL command [3-247](#)

## FILESPERVOLUME parameter

ADD AUDITTRAIL command [3-16](#),  
[3-18](#)

ALTER AUDITTRAIL command [3-54](#)

RESET AUDITTRAIL command [3-219](#)

SET AUDITTRAIL command [3-233](#)

SHOW AUDITTRAIL command [3-247](#)

file-ids [3-7](#)

## file-set parameter

ALTER DUMPS command [3-73](#)

DELETE DUMPS command [3-105](#)

DUMP FILES command [3-125](#), [3-128](#)

INFO DUMPS command [3-165](#), [3-168](#)

RECOVER FILES command [3-202](#)

Finished data operation state [3-277](#)FROMARCHIVE parameter, RECOVER FILES command [3-203](#)**G**Gateway processes [3-37](#)

## GMT parameter

ADD DUMPS command [3-27](#), [3-31](#)

DISPLAY OPERATIONS  
command [3-122](#)

STATUS OPERATIONS  
command [3-278](#)

**H**HELP command [3-145/3-146](#), [B-22](#)Heterogeneous transactions [3-37/3-38](#),  
[3-61](#)HISTORY command [3-147/3-149](#), [B-22](#)Home node [3-10](#), [3-113](#), [3-228](#), [3-296](#)Home terminals [3-78](#), [3-195](#)Hung transactions [3-11](#), [3-12](#), [3-298](#)**I**Identifying dump files [3-105](#)IGNOREDATAERRORS parameter, ABORT TRANSACTION command [3-11](#)IGNOREPREVIOUSCONFIG parameter, ADD DATAVOLS command [3-21](#)

## IN option

TMFCOM run command [2-1](#), [2-5](#)

INFO AUDITDUMP command [3-150/3-151](#),  
[B-23](#)

INFO AUDITTRAIL command [3-152/3-155](#),  
[B-23](#)

INFO BEGINTRANS  
command [3-156/3-158](#), [B-23](#)

INFO CATALOG command [3-159/3-160](#),  
[B-24](#)

INFO DATAVOLS command [3-161/3-163](#),  
[B-24](#)

INFO DUMPS command [3-165/3-171](#),  
[B-25](#)

INFO PROCESS command [3-172/3-175](#),  
[B-26](#)

INFO RESOURCEMANAGER  
command [3-176/3-178](#), [B-26](#)

INFO TAPEMEDIA command [3-179/3-180](#),  
[B-27](#)

INFO TMF command [3-182/3-185](#), [B-27](#)

Inprogress data operation state [3-277](#)

INSPECT debugger [3-78](#)

INSPECT parameter, ALTER PROCESS  
command [3-78](#)

INSPECT save-abend files [3-78](#)

INVALID parameter, ALTER DUMPS  
command [3-73](#), [3-74](#)

INVALID tape volume status [3-168](#)

**L**

LABEL OFF parameter, ADD TAPEMEDIA  
command [3-40](#)



LABEL ON parameter  
    ADD TAPEMEDIA command [3-39](#)  
    SET TAPEMEDIA command [3-244](#)  
LABEL parameter  
    RESET TAPEMEDIA command [3-226](#)  
    SHOW TAPEMEDIA command [3-254](#)  
Labeling tape volumes [3-39](#), [3-40](#), [3-226](#)  
Log file, EMS [3-122](#)  
LOGFILE parameter, DISPLAY  
OPERATIONS command [3-122](#)  
Logical file names [3-212](#)  
LONG parameter, DISPLAY OPERATIONS  
command [3-121](#)

## M

MAKESAVEABEND parameter, ALTER  
PROCESS command [3-78](#)  
MAP NAMES parameter, RECOVER FILES  
command [3-206/3-208](#), [3-211/3-212](#)  
Master audit trails  
    See also Auxiliary audit trails  
    adding [3-14/3-19](#)  
    identifying [3-20](#), [3-43](#), [3-51](#), [3-116](#)  
MASTER parameter  
    ADD AUDITTRAIL command [3-14](#)  
    ALTER AUDITDUMP command [3-43](#),  
    [3-135](#)  
    ALTER AUDITTRAIL command [3-51](#)  
    DISABLE AUDITDUMP  
    command [3-116](#)  
    INFO AUDITDUMP command [3-150](#)  
    INFO AUDITTRAIL command [3-152](#)  
    NEXT AUDITTRAIL command [3-189](#)  
    STATUS AUDITDUMP  
    command [3-259](#)  
    STATUS AUDITTRAIL command [3-261](#)  
    STATUS DATAVOLS command [3-273](#)  
MAXRETAINEDATFILES parameter,  
ALTER AUDITTRAIL command [3-54](#),  
[3-56/3-58](#)  
MEDIUM parameter, ALTER AUDITDUMP  
command [3-46](#)

Message numbers [A-2](#)  
Message types [A-2](#)  
Messages [A-1/A-233](#)  
Monitor process [3-172](#)

## N

NEWDMLOC parameter, RELOCATE  
DISKDUMPS command [3-217](#)  
NEXT AUDITTRAIL command [3-53](#), [3-189](#),  
[B-27](#)  
Node names [3-7](#)  
node parameter  
    ABORT TRANSACTION  
    command [3-10](#)  
    DELETE TRANSACTION  
    command [3-113](#)  
    OBEY command [3-190](#)  
    OBEYVOL command [3-192](#)  
    OPEN SERVER command [3-194](#)  
    RESOLVE TRANSACTION  
    command [3-228](#)  
    STATUS TRANSACTIONS  
    command [3-296](#)  
node-number parameter  
    ABORT TRANSACTION  
    command [3-10](#)  
    DELETE TRANSACTION  
    command [3-113](#)  
    RESOLVE TRANSACTION  
    command [3-228](#)  
    STATUS TRANSACTIONS  
    command [3-296](#)  
Notstarted data volume state [3-272](#)  
NOWAIT option [3-96](#), [3-123](#)  
NOWAIT parameter  
    RECOVER FILES command [3-203](#)  
    START TMF command [3-256](#)  
    STOP TMF command [3-305/3-306](#)

# O

OBEY command [2-5](#), [3-141](#), [3-190/3-191](#), [B-27](#)

OBEY files

See command files

OBEYFORM parameter

INFO AUDITDUMP command [3-150](#)

INFO AUDITTRAIL command [3-152](#)

INFO BEGINTRANS command [3-156](#)

INFO DATAVOLS command [3-159](#), [3-161](#)

INFO DUMPS command [3-32](#), [3-166](#)

INFO PROCESS command [3-173](#)

INFO RECOURCEMANAGER  
command [3-176](#)

INFO TAPEMEDIA command [3-179](#)

INFO TMF command [3-182](#)

OBEYVOL command [3-101](#), [3-190](#), [3-192/3-193](#), [3-310](#), [B-28](#)

object parameter, HELP command [3-145](#)

OLDDMLOC parameter, RELOCATE  
DISKDUMPS command [3-217](#)

Online dump process [3-172](#)

Online dumps

adding [3-28/3-31](#)

disk volumes for [3-29](#), [3-127](#), [3-131](#), [3-224](#), [3-238/3-240](#), [3-252](#)

file recovery from [3-201/3-214](#)

performing [3-22](#), [3-125/3-134](#)

serial numbers for [3-28](#), [3-74](#), [3-224](#), [3-240](#), [3-252](#)

tape volumes for [3-30](#), [3-127](#), [3-131](#)

verifying [3-129](#)

Online help [3-145/3-146](#)

ONLINE recovery mode [3-21](#), [3-22](#), [3-71](#), [3-222](#), [3-236](#)

ONLINEDUMP process type [3-82](#)

ONLINERESTORE process type [3-82](#)

online-dump-file parameter, ADD DUMPS  
command [3-28](#)

Open resource managers [3-283](#)

OPEN SERVER command [3-194/3-200](#), [B-28](#)

Opening servers [3-194/3-200](#)

Opening TMFSERVE process [3-194/3-200](#)

Operations

canceling [3-96/3-97](#)

displaying [3-121/3-124](#)

displaying status of [3-276/3-281](#)

OPERATIONS object

CANCEL command [3-96/3-97](#), [B-14](#)

DISPLAY command [3-121/3-124](#), [B-19](#)

STATUS command [3-276/3-281](#), [B-40](#)

operation-number parameter

CANCEL OPERATIONS  
command [3-96](#)

DISPLAY OPERATIONS  
command [3-121](#), [3-123](#)

STATUS OPERATIONS  
command [3-276](#)

OUT option, TMFCOM run command [2-2](#), [2-8/2-9](#)

Overflow audit volumes [3-16](#), [3-18](#), [3-54](#), [3-219](#), [3-233](#), [3-247](#)

Overflow process [3-172](#)

OVERFLOWTHRESHOLD parameter

ADD AUDITTRAIL command [3-17](#)

ALTER AUDITTRAIL command [3-54](#)

RESET AUDITTRAIL command [3-220](#)

SET AUDITTRAIL command [3-234](#)

SHOW AUDITTRAIL command [3-247](#)

OVERFLOWVOLS parameter

ADD AUDITTRAIL command [3-16](#)

RESET AUDITTRAIL command [3-219](#)

SET AUDITTRAIL command [3-233](#)

SHOW AUDITTRAIL command [3-247](#)

# P

Parallel tape dumps [3-44](#), [3-127](#)

Pattern matching in file-sets [3-126](#), [3-166](#), [3-202](#)

Performing dumps [3-125/3-134](#)

PFSSIZE parameter, ALTER PROCESS command [3-79](#)

Physical file names [3-212](#)

Pins (process identifications numbers) [3-9](#)

Prepared transaction state [3-298](#)

Prepared transactions [3-229](#), [3-298](#)

PRI parameter

- ALTER PROCESS command [3-77](#), [3-84](#)
- OPEN SERVER command [3-194](#)

Process identification numbers (pins)

- See Pins

PROCESS object

- ALTER command [3-75/3-91](#), [B-12](#)
- INFO command [3-172/3-175](#), [B-26](#)

PROCESS parameter

- STATUS RESOURCEMANAGER command [3-284](#)
- STATUS TRANSACTIONS command [3-297](#)

Processes

- changing attributes of [3-75/3-91](#)
- CPUs for [3-77](#), [3-194](#)
- displaying attributes of [3-172/3-175](#)
- extended data segments for [3-79](#)
- file segments for [3-79](#)
- home terminals for [3-78](#), [3-195](#)
- priority of [3-77](#), [3-194](#)
- program files for [3-78](#)
- save-abend files for, INSPECT [3-78](#)
- swap volumes for [3-77](#), [3-194](#)

process-type parameter

- ALTER PROCESS command [3-76](#)
- INFO PROCESS command [3-172](#)

PROGRAMFILE parameter, ALTER PROCESS command [3-78](#)

## Q

Queuedforrecovery data volume state [3-271](#)

## R

RECOVER FILES command [3-201/3-214](#), [B-29](#)

Recoverable resource managers [3-37](#)

Recovering data volume state [3-271](#)

Recovering resource managers [3-283](#)

RECOVERVDPPHASE1 parameter, RECOVER FILES command [3-206](#)

RECOVERYMODE parameter

- ADD DATAVOLS command [3-21](#), [3-22](#)
- ALTER DATAVOLS command [3-71](#), [3-72](#)
- RESET DATAVOLS command [3-222](#)
- SET DATAVOLS command [3-236](#)
- SHOW DATAVOLS command [3-250](#)

RECRMOUNT parameter, ALTER BEGINTRANS command [3-63](#)

Registering resource managers [3-37](#)

RELEASED parameter

- ALTER CATALOG command [3-67](#)
- ALTER DUMPS command [3-73](#)

RELEASED tape volume status [3-41](#), [3-67](#), [3-92](#), [3-168](#), [3-179](#), [3-245](#)

RELOCATE DISKDUMPS command [3-217/3-218](#), [B-31](#)

Requesting online help [3-145/3-146](#)

RESET AUDITTRAIL command [3-219/3-221](#), [B-31](#)

RESET DATAVOLS command [3-222/3-223](#), [B-31](#)

RESET DUMPS command [3-224/3-225](#), [B-32](#)

RESET TAPEMEDIA command [3-226/3-227](#), [B-32](#)

RESETMAXRETAINEDATFILES parameter, ALTER AUDITTRAIL command [3-54](#)

Resetting audit trail attributes [3-219/3-221](#), [3-222/3-223](#)

Resetting dump specifications [3-224/3-225](#)

Resetting tape volume attributes [3-226/3-227](#)

Resident Monitor process [3-172](#)

**RESOLVE TRANSACTION**

command [3-228/3-231](#), [3-257](#), [B-33](#)

Resolving distributed transactions [3-228/3-231](#), [3-257](#)

Resource manager files [3-37](#)

Resource manager registration [3-37](#)

Resource managers [3-37](#)

adding [3-36/3-38](#)

closing [3-98/3-99](#)

deleting [3-108/3-109](#)

displaying attribute values of [3-176/3-178](#)

displaying status of [3-282/3-288](#)

**RESOURCEMANAGER object**

ADD command [3-36/3-38](#), [B-6](#)

CLOSE command [3-98/3-99](#), [B-15](#)

DELETE command [3-108/3-109](#), [B-16](#)

INFO command [3-176/3-178](#), [B-26](#)

STATUS command [3-282/3-288](#), [B-41](#)

Restarting TMF processes [3-88](#)

Restore audit volumes [3-17](#), [3-18](#), [3-54](#), [3-220](#), [3-234](#), [3-248](#)

**RESTOREVOLS parameter**

ADD AUDITTRAIL command [3-17](#)

RESET AUDITTRAIL command [3-220](#)

SET AUDITTRAIL command [3-234](#)

SHOW AUDITTRAIL command [3-248](#)

Resuming audit dump processing [3-135](#)

RETAINDEPTH parameter, ALTER CATALOG command [3-49](#), [3-67](#), [3-131](#), [3-132](#)

RMOPENPERCPU parameter, ALTER BEGINTRANS command [3-63/3-64](#)

**rm-name parameter**

ADD RESOURCEMANAGER command [3-36](#)

CLOSE RESOURCEMANAGER command [3-98](#)

DELETE RESOURCEMANAGER command [3-108](#)

INFO RESOURCEMANAGER command [3-176](#)

**rm-name parameter (continued)**

STATUS RESOURCEMANAGER command [3-282](#)

ROUNDROBIN parameter, ALTER CATALOG command [3-68](#), [3-69/3-70](#)

**S**

Save-abend files, INSPECT [3-78](#)

SCRATCH tape volume status [3-41](#), [3-67](#), [3-92](#), [3-179](#), [3-245](#)

**Security**

See Commands, security restrictions overview

**seq-num parameter**

ABORT TRANSACTION command [3-11](#)

DELETE TRANSACTION command [3-114](#)

RESOLVE TRANSACTION command [3-229](#)

STATUS TRANSACTIONS command [3-297](#)

**SERIAL parameter**

ADD DUMPS command [3-25](#), [3-28](#)

ALTER DUMPS command [3-74](#)

DELETE DUMPS command [3-106](#)

INFO DUMPS command [3-167](#)

RECOVER FILES command [3-204](#), [3-210](#)

RESET DUMPS command [3-224](#)

SET DUMPS command [3-240](#)

SHOW DUMPS command [3-252](#)

Serial tape dumps [3-44](#), [3-127](#)

serial-number parameter, RELOCATE DISKDUMPS command [3-217](#)

**SERVER object**

OPEN command [3-194/3-200](#), [B-28](#)

STATUS command [3-289/3-290](#), [B-41](#)

server parameter, OPEN SERVER command [3-194](#)

Servers, opening [3-194/3-200](#)

## Sessions

ending [3-141](#)

starting [2-1](#)

SET AUDITTRAIL command [3-232/3-235](#), [B-34](#)

SET DATAVOLS command [3-236/3-237](#), [B-34](#)

SET DUMPS command [3-238/3-242](#), [B-35](#)

SET TAPEMEDIA command [3-243/3-246](#), [4-1/4-2](#), [B-35](#)

Setting audit trail attributes [3-232/3-235](#)

Setting data volume attributes [3-236/3-237](#)

Setting dump specifications [3-238/3-242](#)

Setting tape volume attributes [3-243/3-246](#)

SHOW AUDITTRAIL  
command [3-247/3-249](#), [B-36](#)

SHOW DATAVOLS command [3-250/3-251](#), [B-36](#)

SHOW DUMPS command [3-252/3-253](#), [B-36](#)

SHOW TAPEMEDIA  
command [3-254/3-255](#), [B-37](#)

Shutdown [3-120](#)

Shuttingdown data volume state [3-272](#)

SORT parameter

INFO DATAVOLS command [3-162](#)

STATUS DATAVOLS command [3-273](#)

STATUS OPERATIONS  
command [3-278](#)

SORT parameter, INFO DUMPS  
command [3-167](#)

Sorting dump information [3-167](#)

Specifying default subvolume  
names [3-100/3-101](#), [3-192/3-193](#),  
[3-309/3-310](#)

Specifying default volume  
names [3-100/3-101](#), [3-192/3-193](#),  
[3-309/3-310](#)

START parameter, DUMP FILES  
command [3-128](#), [3-130](#), [3-134](#)

START parameter, INFO DUMPS  
command [3-169](#)

START TMF command [3-256/3-258](#), [B-37](#)

Started data volume state [3-271](#)

Starting data volume state [3-271](#)

Starting TMF [3-256/3-258](#)

Starting TMFCOM sessions [2-1](#)

Startupbegun data volume state [3-271](#)

Startupcompleting data volume state [3-271](#)

STATE parameter

RESOLVE TRANSACTION  
command [3-229](#)

STATUS DATAVOLS command [3-270](#)

STATUS OPERATIONS  
command [3-277](#)

STATUS RESOURCEMANAGER  
command [3-283](#)

STATUS TRANSACTIONS  
command [3-297/3-298](#)

STATUS AUDITDUMP  
command [3-259/3-260](#), [B-38](#)

STATUS AUDITTRAIL  
command [3-261/3-266](#), [B-38](#)

STATUS BEGINTRANS command [3-268](#),  
[B-38](#)

STATUS CATALOG command [3-269](#), [B-38](#)

STATUS DATAVOLS  
command [3-270/3-275](#), [B-39](#)

STATUS OPERATIONS  
command [3-276/3-281](#), [B-40](#)

STATUS parameter

ADD TAPEMEDIA command [3-40](#)

ALTER TAPEMEDIA command [3-92](#)

INFO DUMPS command [3-168](#)

INFO TAPEMEDIA command [3-179](#)

RESET TAPEMEDIA command [3-226](#)

SET TAPEMEDIA command [3-244](#)

SHOW TAPEMEDIA command [3-254](#)

STATUS RESOURCEMANAGER  
command [3-282/3-288](#), [B-41](#)

STATUS SERVER command [3-289/3-290](#),  
[B-41](#)

STATUS TMF command [3-291/3-295](#), [B-41](#)

STATUS TRANSACTIONS  
command [3-296/3-304](#), [B-42](#)

STOP TMF command [3-305/3-308](#), [B-42](#)

Stopping TMF [3-305/3-308](#)



Stopping TMF processes [3-88](#)  
 Subvolume names [3-7](#)  
 subvolume parameter  
     CMDVOL command [3-100](#)  
     NEXT AUDITTRAIL command [3-190](#)  
     OBEY command [3-190](#)  
     OBEYVOL command [3-192](#)  
     VOLUME command [3-309](#)  
 Suppressing display of EMS  
 messages [3-256](#), [3-305](#)  
 Suspending audit dump  
 processes [3-116/3-117](#)  
 SWAP parameter  
     ALTER PROCESS command [3-77](#),  
     [3-84](#)  
     OPEN SERVER command [3-194](#)  
 Switching audit trail files [3-189](#)  
 Switching the primary and backup TMP  
 process [3-95](#)  
 SWITCHPROCESS TMP parameter,  
 ALTER TMF command [3-94](#)  
 System names  
     See Node names  
 SYSTEM parameter  
     ALTER AUDITDUMP command [3-45](#)  
     DUMP FILES command [3-129](#)  
     RECOVER FILES command [3-204](#)

## T

Tape volumes  
     adding [3-39/3-42](#)  
     cartridge [4-2](#)  
     changing attributes of [3-92/3-93](#)  
     data block length of [3-44](#), [3-126](#), [4-4](#)  
     default attribute values for [3-226/3-227](#)  
     deleting [3-110](#)  
     density of [3-40](#), [3-226](#), [3-243](#), [3-254](#),  
     [4-1/4-2](#)  
     displaying attributes of [3-179/3-180](#),  
     [3-254/3-255](#)  
     drives for [3-40](#), [3-226](#), [3-243](#), [3-254](#)

Tape volumes (continued)  
     guidelines for using [4-3/4-5](#)  
     identifying [3-39](#), [3-92](#), [3-110](#), [3-179](#)  
     labeling [3-39](#), [3-40](#), [3-226](#), [3-244](#),  
     [3-254](#), [4-2/4-3](#)  
     monitoring [4-3](#)  
     open-reel [4-1](#)  
     resetting attributes of [3-226/3-227](#)  
     selecting model types of [4-4](#)  
     setting attributes of [3-243/3-246](#)  
     status of [3-40](#), [3-67](#), [3-92](#), [3-168](#),  
     [3-179](#), [3-226](#), [3-244](#), [3-254](#)  
     unloading [3-40](#), [3-46](#), [3-129](#), [3-226](#),  
     [3-245](#), [3-254](#)  
     verifying dumps on [3-45](#), [4-4](#)  
 TAPEMEDIA object  
     ADD command [3-39/3-42](#), [4-1/4-2](#), [B-6](#)  
     ALTER command [3-92/3-93](#), [B-14](#)  
     DELETE command [3-110](#), [B-16](#)  
     INFO command [3-179/3-180](#), [B-27](#)  
     RESET command [3-226/3-227](#), [B-32](#)  
     SET command [3-243/3-246](#), [4-1/4-4](#),  
     [B-35](#)  
     SHOW command [3-254/3-255](#), [B-37](#)  
 TAPEMEDIA parameter  
     ADD DUMPS command [3-26](#), [3-30](#),  
     [3-33](#)  
     DUMP FILES command [3-129](#), [3-132](#),  
     [3-134](#)  
     RESET DUMPS command [3-224](#)  
     SET DUMPS command [3-240](#)  
     SHOW DUMPS command [3-252](#)  
 tape-labeling options parameter, ADD  
 TAPEMEDIA command [3-39](#)  
 tape-status-options parameter, ADD  
 TAPEMEDIA command [3-40](#)  
 tape-volume parameter  
     ADD TAPEMEDIA command [3-39](#)  
     ALTER TAPEMEDIA command [3-92](#)  
     DELETE TAPEMEDIA command [3-110](#)  
     INFO TAPEMEDIA command [3-179](#)

## TERM parameter

ALTER PROCESS command [3-78](#)OPEN SERVER command [3-195](#)

## TIME parameter

ADD DUMPS command [3-27](#), [3-30](#), [3-32](#)RECOVER FILES command [3-204](#)TMF configuration, deleting [3-111/3-112](#)TMF library process [3-172](#)TMF management process [3-172](#)

## TMF object

ALTER command [3-94/3-95](#), [B-14](#)DELETE command [3-111/3-112](#), [B-17](#)INFO command [3-182/3-185](#), [B-27](#)START command [3-256/3-258](#), [B-37](#)STATUS command [3-291/3-295](#), [B-41](#)STOP command [3-305/3-308](#), [B-42](#)TMF objects [1-2/1-3](#)TMF shutdown [3-120](#)

## TMF (Transaction Management Facility)

configuration limits and defaults [C-1/C-5](#)displaying attributes of [3-182/3-185](#)displaying status of [3-291/3-295](#)purpose of [1-1](#)starting [3-256/3-258](#)stopping [3-305/3-308](#)

## TMFCOM command interface

interrupting [2-3](#)introduction to [1-4](#)purpose of [1-4](#)running [2-1/2-5](#)starting [2-1](#), [2-2](#)using [2-1/2-12](#)

## TMFCOM commands

See also individual command names

summary of [B-1/B-43](#)TMFCOM messages [A-1/A-233](#)TMFCOM run command [2-1/2-5](#)

## TMFCOM sessions

ending [3-141](#)starting [2-1](#)TMFLIBMEMTHRESH parameter, ALTER BEGINTRANS command [3-62](#)TMFMON process [3-172](#)TMFMON process type [3-82](#)TMFMON2 process type [3-82](#)

## TMFSERVE process

accessing on local node [3-196](#)accessing on remote node [3-197](#)displaying status of [3-289/3-290](#)opening [3-194/3-200](#)TMP process type [3-83](#)TMP process, switching the primary and backup [3-95](#)TMPMEMTHRESH parameter, ALTER BEGINTRANS command [3-62](#)

## tm-flags parameter

ABORT TRANSACTION

command [3-10](#), [3-297](#)

RESOLVE TRANSACTION

command [3-229](#)TOFIRSTPURGE parameter, RECOVER FILES command [3-204](#)TOMATPOSITION parameter, RECOVER FILES command [3-205](#), [3-210](#)TOPHYSVOL parameter, RECOVER FILES command [3-206](#), [3-212](#)

## TRANSACTION object

ABORT command [3-10/3-13](#), [B-2](#)DELETE command [3-113/3-115](#), [B-17](#)RESOLVE command [3-228/3-231](#), [B-33](#)

RESOLVE TRANSACTION

command [3-257](#)

## Transaction processing

See also Transactions

disabling [3-136](#)enabling [3-118](#)

## Transactions

- aborting [3-10/3-13](#), [3-229/3-230](#)
- committed [3-229](#)
- deleting [3-113/3-115](#)
- disabling [3-61/3-62](#)
- displaying status of [3-296/3-304](#)
- enabling [3-61/3-62](#)
- heterogeneous [3-37/3-38](#), [3-61](#)
- hung [3-11](#), [3-114](#)
- identifying [3-10](#), [3-113](#), [3-228](#), [3-296](#)
- prepared [3-229/3-230](#)
- Prolonged ENDING state [3-300/3-301](#)
- resolving distributed [3-228/3-231](#)
- sequence numbers for [3-11](#), [3-114](#), [3-229](#)

TRANSACTIONS object, STATUS command [3-296/3-304](#), [B-42](#)

Transactions, heterogeneous [3-37/3-38](#), [3-61](#)

Transactions, resolving distributed [3-257](#)

transaction-id parameter

- ABORT TRANSACTION command [3-10](#)

- DELETE TRANSACTION command [3-113](#)

- RESOLVE TRANSACTION command [3-228](#)

- STATUS TRANSACTIONS command [3-296](#)

TRANSCOUNTTHRESH parameter, ALTER BEGINTRANS command [3-61/3-62](#)

TRANSPERCPU parameter, ALTER BEGINTRANS command [3-62/3-63](#)

TYPE AUDITDUMP parameter, ADD DUMPS command [3-25](#)

TYPE ONLINEDUMP parameter, ADD DUMPS command [3-28](#)

TYPE parameter

- INFO DUMPS command [3-168](#)
- RESET DUMPS command [3-224](#)
- SET DUMPS command [3-240](#)

TYPE parameter (continued)

- SHOW DUMPS command [3-252](#)

- STATUS OPERATIONS command [3-276/3-277](#)

## U

Unconfigured data volume state [3-273](#)

UNLOAD parameter

- ADD TAPEMEDIA command [3-40](#)

- ALTER AUDITDUMP command [3-46](#)

- DUMP FILES command [3-129](#)

- RECOVER FILES command [3-206](#)

- RESET TAPEMEDIA command [3-226](#)

- SET TAPEMEDIA command [3-245](#)

- SHOW TAPEMEDIA command [3-254](#)

Unloading tape volumes [3-40](#), [3-46](#), [3-129](#), [3-226](#), [3-245](#), [3-254](#)

username parameter, ADD RESOURCEMANAGER command [3-36](#)

user-id parameter, ADD RESOURCEMANAGER command [3-36](#)

Using TMFCOM [2-1/2-12](#)

## V

Verifying online dumps [3-129](#)

VERIFYTAPE parameter

- ALTER AUDITDUMP command [3-45/3-46](#), [4-4](#)

- DUMP FILES command [3-129](#), [4-4](#)

Volatile resource managers [3-37](#), [3-283](#)

VOLUME command [3-101](#), [3-190](#), [3-193](#), [3-309/3-310](#), [B-43](#)

Volume names [3-7](#)

volume parameter

- ADD DATAVOLS command [3-20](#)

- ALTER DATAVOLS command [3-71](#)

- CMDVOL command [3-100](#)

- DELETE DATAVOLS command [3-103](#)

- DISABLE DATAVOLS command [3-119](#)

- ENABLE DATAVOLS command [3-137](#)



volume parameter (continued)

INFO DATAVOLS command [3-161](#)

OBEY command [3-190](#)

OBEYVOL command [3-192](#)

STATUS DATAVOLS command [3-270](#)

VOLUME command [3-309](#)

Volume recovery process [3-172](#)

VOLUMERECOVERY process type [3-83](#)

## W

WHEREPHYSVOLIS parameter,  
RECOVER FILES command [3-206](#), [3-212](#)

Wrongdiskmounted data volume  
state [3-273](#)

## Z

ZTMFAT subvolumes [3-18](#), [3-25](#)

## Special Characters

! command [3-311/3-313](#), [B-43](#)

\$ZTMFAT volumes [3-25](#)

? command [3-314/3-315](#), [B-43](#)

