



Metalogic

# TRIM

## Reference Manual

Copyright © 1979-2004 Metalogic S.à R.L., Luxembourg.  
All rights reserved.

Relative to Supervisor version 500.25  
OPAL version 500.14  
OPALTAPELIB version 500.03  
TAPELIBUPDATER version 500.05

Last revision: July 27th, 2004

METALOGIC believes that the information described in this manual is accurate and reliable, and much care has been taken in its preparation. However, no responsibility, financial or otherwise, is accepted for any consequences arising out of the use of this material.

The information contained herein is subject to change. Revisions may be used to advise of such changes and/or additions.

#### ACKNOWLEDGEMENTS

Many people, over a long period of time, have contributed to this software. To all of our contributors, many thanks.

METALOGIC S.à R.L.  
113, Rue de Luxembourg  
L-7540 Rollingen  
Luxembourg

<http://www.metalogic.eu.com>

---

<b>Chapter 1: Introduction to TRIM.....</b>	<b>1</b>
Tape Library Management .....	1
TRIM Features .....	1
About the TRIM manual .....	4
<b>Chapter 2: TRIM concepts.....</b>	<b>5</b>
The Ideal TRIM Environment .....	7
Software Installation.....	9
Running TRIM Software .....	10
Files and codefiles .....	11
SUPERVISOR.....	11
TAPELIBUPDATER process.....	12
MAGUS library .....	12
METATAPELIB database .....	13
LOGREADER process.....	13
TRIM utility.....	14
TAPEMANAGER library .....	14
OPALTAPELIB library .....	14
TAPELOG file .....	15
SUMLOG file.....	15
VOLUME Library.....	15
User interfaces .....	16
TRIM Recovery.....	17
Multi-host environments .....	18
<b>Chapter 3: Guided Tour: Getting started .....</b>	<b>19</b>
Communicating with SUPERVISOR.....	19
Tape Events.....	22
TAPELABEL event.....	22
LOG VOLUME (Log MajorType 15, Minor Types 3, 5) event..	23
TAPEDB event .....	24
TRIM preliminaries.....	25
Introducing TP commands .....	26
OPAL programs .....	33
OPAL attributes .....	34
Tape label printing.....	35
The Tape Librarian.....	38
Summary .....	39
<b>Chapter 4: TRIM Rules Management .....</b>	<b>40</b>
TRIM rules .....	40

---

Rule definition.....	42
DMSII dumps example.....	42
RULE characteristics .....	43
Unlabelled tapes and TRIM rules .....	44
Tape Creation .....	45
TRIM Utility.....	46
TRIM Utility: Home menu.....	47
Rule usercode.....	48
Rule identity (rule identifier).....	48
Rule index.....	48
Creating rules .....	49
USERCODE .....	50
VOLUME IDENTIFIER (VOLID).....	50
FILEID.....	50
HOSTNAME .....	50
Wildcard characters .....	51
Retention information.....	52
Retention by DAYS or YEARS.....	52
Retention by GENERATION .....	52
Retention by SAVE and NEVER .....	53
LOCATION movement .....	53
Interrogating rules .....	55
Navigating TRIM utility .....	56
Significance of RULEINDEX.....	56
Updating rules.....	58
Using SR and APPLY.....	58
Rules in action .....	59
Generations and locations.....	61
Using SR .....	61
Generation-base rule example: .....	61
Authorisation phase .....	63
Generation family movement .....	65
Linking rules.....	68
Unlabelled tape rules.....	70
Permanent tapes.....	71
Linking delays.....	72
EXCEPTION rules.....	74
Testing and changing rules.....	75
TRIM strategy .....	78
Building a tape library database.....	78

---

Review tape expiration .....	80
New rules and EXCEPTION tapes.....	80
Making an example rule .....	81
Building a rule family .....	82
Reporting and CONFIRM phases .....	83
EXCEPTION rules .....	83
Modifying rules .....	84
Using "linked" rules .....	84
De-implementing old retention systems .....	84
Caring for your TAPELOG .....	84
Regular CONFIRMs .....	85
Software changes and enhancements .....	85
<b>Chapter 5: TP commands.....</b>	<b>86</b>
Railroad Diagrams .....	86
Tape modification commands.....	88
OPAL attributes and METATAPELIB DASDL fields.....	89
Rule reporting and authorisation .....	90
Modifying Tape Information.....	91
TP serial number commands.....	92
BOX modifier .....	94
CDATE modifier .....	95
CLEANED modifier.....	96
DAMAGED modifier.....	97
DEFER modifier .....	98
DELETE modifier .....	99
DESTROYED modifier .....	100
EXPIRY modifier .....	101
FILEID modifier .....	103
JOBID modifier.....	104
JOBNO modifier.....	107
LOCATION modifier .....	108
LOST modifier .....	110
MIXID modifier .....	111
MIXNO modifier .....	112
NOTE modifier.....	113
OFFSITE modifier .....	114
OWNER modifier.....	115
PEND modifier.....	117
PGNEVER modifier .....	118

---

PGOK modifier.....	119
PLOC modifier .....	120
REPORTED modifier.....	121
RETURN modifier.....	122
SEND modifier .....	123
SITE modifier.....	124
SR modifier .....	125
SYSNO modifier.....	127
UNASSIGN modifier.....	128
VOLID modifier .....	129
Reporting and rule processing commands.....	131
APPLY command.....	132
ASSIGN command.....	133
CLOSE command.....	134
CONFIRM command .....	136
CONFIRM errors and warnings.....	138
DUMP command .....	140
FIND command.....	141
Patterns and wildcard strings.....	142
FIND DATE command.....	143
FIND FAMILY command .....	146
FIND LINKED command .....	148
FIND LOCATION command .....	149
FIND NAME command .....	151
FIND PENDING command.....	154
FIND POOL command.....	156
FIND RULES command .....	157
FIND SCRATCH command.....	159
Density modifier.....	160
CD modifier.....	160
EXP qualifier .....	161
POOL modifier .....	161
TBC qualifier .....	161
HELP command .....	163
LC command .....	165
LOG command .....	166
NET command .....	168
OPEN command .....	170

---

SHOW command .....	171
VERSION command .....	172
Run-time Options .....	173
<b>Chapter 6: Getting started with OPAL.....</b>	<b>174</b>
OPAL programs .....	174
OPAL SITUATIONs .....	175
OPAL ODTSEQUENCEs .....	176
OPAL DISPLAYs .....	177
OPAL Contexts.....	178
TAPELABEL vs. TAPEDB.....	178
TAPEDB sub-contexts.....	179
DEFINE command .....	180
DEFINE interrogation.....	181
Sample OPAL Program.....	183
EVALUATE Command.....	186
DO command .....	188
TEST (DO) Command .....	189
Invoke DISPLAY (/) Command .....	190
WHEN Command.....	192
TRIM-specific OPALs.....	193
TPDB_DAILYREPORTS .....	193
TPDB_REPORT.....	194
TPDB_REPSETUP .....	195
TPDB_DEFAULTS.....	196
TPDB_REPHEAD .....	197
TPDB_REPFOOT .....	197
TPDB_PERRECORD .....	198
TRIM-specific OPAL functions and attributes.....	200
GETSCRATCH function .....	200
MOUNT_REQUEST example .....	202
GETSCRATCH: INSILO modifier.....	203
TAPENAMESERIALS attribute.....	204
TAPEFAMILYSERIALS attribute.....	205
TAPEFAMILYNAME attribute .....	205
<b>Chapter 7: TAPEMANAGER Implementation .....</b>	<b>206</b>
Installation and Operations interface.....	207
Initiating TAPEMANAGER.....	207
Interrogating TAPEMANAGER status .....	208
Terminating the TAPEMANAGER Library.....	209
Temporarily disabling of the TAPEMANAGER library .....	210

---

Purge protection.....	211
Tape assignment by serial number .....	212
TAPEMANAGER errors .....	213
Purge Error: Not PGOK.....	213
Purge Error: Dup serial on SN.....	213
Purge Error: Volumeid not matched .....	213
Purge Error: Database not on-line.....	214
Purge Error: MASTER unavailable.....	214
Purge Error: Marked PGNEVER.....	214
Purge Error: Volume not in DB .....	214
Purge Error: No serials given.....	215
Purge Error: Already assigned to NNNN.....	215
Purge Error: FLEXLIB is off-line.....	215
Purge Error: Tape has backup refs.....	215
Purge Error: B&L tape is not empty.....	215
Successful PG and SN operations .....	216
SN considerations.....	216
Automatic purging of expired tapes .....	217
TPDB_AUTOPG Opals.....	218
Tape purging.....	219
<b>Chapter 8: Networking TRIM .....</b>	<b>220</b>
Establishing a multi-host TRIM environment .....	221
Networking in a MASTER TCPIP environment .....	221
Networking a SLAVE TCPIP environment .....	222
Capabilities of a slave TRIM system .....	223
Switching slave and master systems .....	225
Prerequisites.....	225
Changing SLAVE to MASTER.....	225
Changing MASTER to SLAVE.....	228
Restart and recovery on slave systems .....	229
OPAL attributes usage from SLAVEs .....	230
<b>Appendix A: User Interfaces .....</b>	<b>232</b>
COMS windows .....	233
MARC directive.....	237
PRINTLABEL utility.....	239
RECORD[47] .....	241
<b>Appendix B: Tape Robot support .....</b>	<b>242</b>
Robot Tape Silos and TRIM .....	243
Functionality .....	244
TK command.....	246



---

Supported OPAL attributes .....	247
DSI robot support .....	248
USE SILO DSI.....	248
USE SILO DSI_SERIALNO .....	248
USE SILO DSI_NOTM .....	249
TM command interface.....	249
Supported OPAL attributes .....	250
Example OPALs for Robot Automation .....	251
TPDB_SILOW Opals .....	251
TPDB_EJECT Opals.....	253
<b>Appendix C: METATAPELIB DASDL .....</b>	<b>255</b>
Changes to DASDL .....	258

# Chapter 1: Introduction to TRIM

## Tape Library Management

*Knowledge is of two kinds. We know a subject ourselves, or we know where we can find information upon it.*

Samuel Johnson, Boswell Life vol. 2,

Since the early days of computers, magnetic tapes have been a mainstay at computer installations. Even now, with all the advances in disk technology, there is no cheaper way of making and maintaining backup copies of files than using tapes. The low cost of tape media makes them much more cost-effective for storing infrequently used files over long periods, saving expensive disk space. However, as computer installations have increased the amount of information that they store and process, the problems in managing the large numbers of tapes have grown even faster. Metalogic's **TRIM (Tape Rules Information Management)** system is designed to efficiently maintain a tape library on any Unisys A-Series system or in a multi-host environment, but unlike other tape library systems, allows the site to *control* the usage of tapes.

## TRIM Features

TRIM combines the power of Expert Systems with software that is carefully integrated with the MCP and DMSII. This integration, a characteristic feature of all METALOGIC software, allows for powerful features without taking significant system overhead. In particular, all of the features of TRIM constitute a major component of the SUPERVISOR package and SUPERVISOR itself is the engine responsible for much of TRIM functionality.

There are many references to the Metalogic SUPERVISOR program in this document, where appropriate, and you should be aware that the **Metalogic SUPERVISOR Reference Manual** is the definitive source of information on this product. The TRIM Tape Library software is automatically included in the standard SUPERVISOR package but it may be acquired separately with a SUPERVISOR of restricted functionality. This manual will only describe those SUPERVISOR features relevant and available within the TRIM product.

The major features of TRIM include:

- MCP event-based information supplied to SUPERVISOR provides real-time update of the TRIM DMSII database and can enforce positive site-defined actions to prevent operator errors.
- Information about tapes and their history is kept in an optimised, audited DMSII database called METATAPELIB.
- TRIM “rules” allow custom control of tape retention, providing automatic assignment of expiry dates and/or generation controls for all kinds of tape (including unlabelled). Groups of tapes with the same rule are called “families”. Rule linking facilities allow one or more independent tape families to be attached to a master tape family. In such cases, all movement and retention actions on the master tapes will be propagated to the linked families.

For example, this is particularly useful for DMSII dump tapes, each tape having a sub-family of unique AUDIT tapes. When the master dump tapes go off-site or have expired, the relevant, linked AUDIT tapes will automatically follow.

- Integration with the Unisys TAPEMANAGER system software, allows automatic purge verification, protection and authorisation without the need to alter operations procedures for **PG** (Purge) and **SN** (Serial Number) commands and with minimal operator intervention.
- TRIM supports printed tape labels with complete control over the content and layout of each label using a variety of system and tape attributes. This also includes rules-generated tape information such as expiry date and any next pending location.
- Labels can be printed on a remote terminal, or an ODT using the printer port, which needs much less expensive hardware, fewer computer resources and does not require datacom to be in operation.
- TRIM has a comprehensive set of operator commands which allow interrogation and modification of tape information. Commands can be received via a COMS window, SUPERVISOR Remotespo, MARC Directive or ODT. The Metalogic ODT interface has the look and feel of the standard Unisys ODT command facility. Each interface has access to comprehensive on-line HELP facilities.
- TRIM rules provide wild-card matching on usercode, volume and file identifiers plus control of multiple locations. This movement between

site-selected locations can be time or generation based and is always under the full control of the user.

- TRIM users are allowed to maintain rules to control their own tapes, but not those belonging to other users. A site defined Tape Librarian usercode has access to all rules and tape families.
- If the MCP Option CATALOGING is set, many extra features are available for tape tracking. Although the TRIM system will support CATALOGING systems, Metalogic's FLEX packages offers increased functionality.
- Full support of multi-host A-Series BNA environments. TRIM may optionally function in a master-slave configuration i.e. the METATAPELIB database resides on a single system served by other satellite hosts supplying tape information across the network via port files.
- TRIM provides history for each tape up to 5 generations.
- The built-in OPAL programming language provides the site with the capability to refine the TRIM environment, enforce site operational policies and provide flexible, powerful reporting facilities.
- TRIM is robust against system or software failure and will reference SUMLOGs using the LOGREADER utility to recover "missed" tape information after a restart.

In summary, TRIM offers very powerful and automatic tape handling capabilities. Learning how to use and maintain retention rules can take just minutes and rules can be applied and tested directly in real-life SITUATIONS. This guide has several sections that comprehensively inform on how to get started using TRIM and understanding how rules work.

---

## About the TRIM manual

Although Supervisor is a major component of the TRIM system, the capabilities and features of this software are not covered in this manual. You should refer to the **Metalogic Supervisor Reference** manual for more information on its functionality and use though it should be reiterated that, in a TRIM-only environment, many Supervisor features have reduced functionality or are absent.

The following section, **Chapter 2: TRIM Concepts**, briefly describes the installation process and gives a description of how the TRIM package is constructed. This section is essential reading as it establishes the terminology of the rest of the manual. The install process is covered in more detail in the accompanying **Metalogic INSTALL Reference Manual**.

The next section, **Chapter 3: Guided Tour**, introduces the way in which the TRIM system might be used during normal day-to-day running. Since TRIM has been designed to be run and maintained by operations staff, without any need for systems programmer intervention, this section can be used to become familiar with the features and capabilities of the TRIM software as quickly as possible.

**Chapter 4: TRIM rules management** discusses the concepts and maintenance of TRIM rules which automatically control the retention and expiration of tapes.

**Chapter 5: TRIM operator commands** discusses access to and modification of tape library information from various operator interfaces.

Next, **Chapter 6: Getting started with OPAL** gives a brief introduction to some examples of OPAL programming that can assist with the day-to-day running of the TRIM system. The **Metalogic OPAL Language Reference** manual is useful for reference when reading this section.

**Chapter 7: TAPEMANAGER implementation** discusses how the TRIM system integrates with the Unisys MCP TAPEMANAGER mechanism to implement both tape purge protection and automatic purging of expired tapes.

Lastly, **Chapter 8: networking TRIM** discusses how to implement TRIM in a multi-host BNA network using, if desired, a master-slave environment .

## Chapter 2: TRIM concepts

*When old settlers say 'One has to understand the country,' what they mean is, 'You have to get used to our ideas about the native.'*

Doris Lessing, *The Grass is Singing*

As stated in the introduction in Chapter 1, TRIM is just one module of the SUPERVISOR package. The SUPERVISOR software provides all of the tape tracking and event handling capabilities, much of the operator interface and has a programmatic interface direct to tape library information through the use of the integrated OPAL compiler. OPAL supplements the TRIM system with significant power and flexibility for reports, exception handling, printing tape labels and many more real-time activities. The functionality of OPAL is discussed in more detail in **Chapter 7: Getting started with OPAL**.

First, some simple definitions.

The core of TRIM is its rule system; a "**rule**" may be defined as those controls needed to uniquely identify a tape by its volume identifier, file identifier and owner and apply pre-assigned retention controls to enforce automatic tape expiration by time, by generation or both. The **volume** and **file identifier** fields are the names used to describe a multi-level tape name, easily seen for a mounted tape using the A-Series ODT command 'PER MT'. However, the "**owner**", in this case, is the usercode of the task or job that originally created the tape. Flexibility, in the form of wildcard patterns, allow one rule to match many tapes which may have been created by the same application but whose names change from day to day e.g. because the date is built into the tape name.

Rules can be wholly maintained by the TRIM software and may be easily modified, tested and re-applied, if modified, to existing members of the rule's family of tapes. A "family" is defined as the collection of tapes whose tape name and owner attributes match a specific rule identifier.

Each rule is given a unique identifier that may optionally be protected by a usercode. The combination of the tape name attributes uniquely identifies a rule and the user can select expiry criteria to match the rule. The expiry criterion is usually by time retention (keep this tape for 30 days) or by generation (keep this tape for 5 generations; when it becomes generation 6 then release it).

When a tape is created, TRIM searches the rule set looking for a unique match. The rule attributes can include wildcard characters if desired, making the rules much more flexible. When a rule is matched, the expiration and movement criteria are applied. This new information is immediately available after the creation of the tape.

When tapes are released because of the rules controls, they will become marked as **PGOK** (purge authorised). This means that the tape has been returned to the site's scratch pool for re-use. The TRIM command, **TP FIND SCRATCH**, will return information about all purge authorised and SCRATCH tapes that currently exist in the METATAPELIB database.

The following chapters deal with how to use rules in great detail. To get the optimal benefits from using TRIM, it is strongly recommended that you start learning about rules, and how to maintain and use them, as soon as possible.

## The Ideal TRIM Environment

*GERRY: We can't get married at all....I'm a man.*

*OSGOOD: Well, nobody's perfect.*

I. A. L. Diamond and Billy Wilder, 'Some Like It Hot' (1959 film; closing words)

TRIM can be implemented, without exception, on any Unisys A-Series system. There are, however, certain system criteria that will suit a TRIM environment better than any other. These criteria, discussed below, hold very well as recommendations for the running and maintenance of any tape library system:

- Use unique tape serial numbers. Duplicate serial numbers can cause problems with reporting and automatic purging.
- Use the facilities provided by the Unisys TAPEMANAGER implementation. This can provide secure protection against accidental purging and also can be used to automatically purge authorised, expired tapes.
- METALOGIC provides a Unisys-compatible TAPEMANAGER library which fully provides this tape security. In the standard TRIM implementation, the feature is available for both single and multi-host systems.
- Use the “scratch” pool concept. TRIM marks an individual tape as expired, subject to the retention criteria specified by its rule, by setting a special Boolean attribute called PGOK to TRUE. TRIM has commands which allow the inspection of all tapes currently held in the “scratch” pool to be visible which include those marked as PGOK (purge authorised) and those tapes which really have been purged or SN-ed manually.
- If using the TAPEMANAGER implementation, TRIM will only allow labelled tapes to be purged if the PGOK flag is active. Even unlabelled tapes can be protected from accidentally SN-ing with valid, non-expired tape serial numbers.
- If your site has third-party, non-Unisys tape drives such as DAT, use different serial number ranges to uniquely identify tapes used in these drives. Very often, the MCP will not give the correct density for such drives making it very difficult, in the tape library, to identify whether a tape volume is open-reel (BPI 6250) or a third-party peripheral.



- 
- Backup your Tape Library system regularly including audit files, ideally at least once a day. If using the latest releases of DMSII, on-line dumps can be done without even closing the database.
  - For tapes that are moving regularly to on-site and off-site locations, ensure that tape movement reports and purge authorisation activities are actioned on a daily basis. In the case of the TRIM system, two simple commands are need to achieve this and failure to do this regularly, even if none of your tapes go off-site, can lead quickly to an acute shortage of eligible scratch tapes.

## Software Installation

As with all Metalogic products, the installation process is achieved by running the **OBJECT/META/INSTALL** utility. This section does not describes the general install process but we refer you instead to the **Metalogic INSTALL Reference Manual** for more detailed information, with special attention to the use of configuration variables.

Many of these configuration variables set during the installation process can be easily changed at any time using the INSTALL utility without resorting to a full re-install. Again, this facility is described in greater detail in the INSTALL reference manual.

INSTALL asks many questions about the set up of the METATAPELIB tape Library database in particular. File locations, usercode and various DASDL options may all be queried during this process. INSTALL will, when requested, generate the DMSII software automatically as well as a special library codefile, called METALOGIC/OPALTAPELIB4, which handles all database access. This codefile must be re-generated every time a new METATAPELIB DESCRIPTION file is compiled whether the system DMSII software is being upgraded or the DASDL file has been manually changed.

INSTALL will automatically load all relevant codefiles, including the newly compiled OPALTAPELIB library, to their final destinations and apply any necessary privileges that need to be conferred (using the **MP** ODT command). INSTALL will SL the MAGUS and OPALTAPELIB libraries followed by a QUIT and restart of SUPERVISOR. The restart of Supervisor will automatically re-initiate the Tape Library software.

Prior to installing a new version of TRIM, it is strongly recommended that a full DMSII dump be always performed for the METATAPELIB database before loading and new files. This should include the tailored DMSII software (e.g. the CONTROL file and DMSUPPORT library) plus the current OPALTAPELIB4 library codefile. In most cases, installation of a new Metalogic software release can be done in a few minutes.

In earlier releases, Metalogic-enforced changes to the DASDL symbolic necessitated DMS reorganisations performed by the INSTALL utility. This procedure has now changed and the reorganisation path has been superseded by a programmatic LOAD-DUMP mechanism driven by INSTALL. Any future changes to the DASDL symbolic will use this new procedure.

## Running TRIM Software

Starting up the TRIM system software is critical on the successful installation and invocation of the Supervisor program. After completion of the installation process using the INSTALL utility, the new Supervisor codefile should be invoked by Quitting the active SUPERVISOR (if you have already been using the package) or, if you are a first-time user, by entering from the ODT:

```
??RUN *METALOGIC/SUPERVISOR
```

Supervisor performs many actions during initialisation. If this the first invocation of Supervisor, several important actions will be performed:

Supervisor will create it's own MCS name (\*METALOGIC/SUPERVISOR) in the system's DATACOMINFO file

Add itself into the list of Automatic Initiators (AI)

Create a system queue for its own usage (usually Q255)

Create a recovery file called \*SCHEDULE which holds information about the TRIM and Supervisor environments

If the Supervisor **TAPELIB** option is subsequently set (using the **TT SO** command) and the TRIM database software has been installed, SUPERVISOR will start recording tape events into the METATAPELIB database immediately.

In future restarts of Supervisor, if the TAPELIB option is set, Supervisor will automatically bring up the TRIM system, opening the METATAPELIB database and will scan the SUMLOG file for any missed tape activity that may have occurred whilst Supervisor was temporarily unavailable.

## Files and codefiles

*'What are the bugles blowin' for ?' said Files-on-Parade.*

*'To turn you out, to turn you out,' the Colour-Sergeant said.*

Rudyard Kipling, Barrack-Room Ballads (1892) 'Danny Deever'

The TRIM Tape Library system uses a DMSII database, called METATAPELIB4, for its primary data storage. The METATAPELIB database has various set structures allowing fast access to tape library data for both command and programmatic interfaces. This operator command subset is extensive and interfaces exist for the ODT, COMS windows and MARC.

The INSTALL process loads various files, many of which are very critical to the TRIM system. The files discussed here are presented with their default names though it is possible to change the names of some codefiles if desired. Usually the restriction remains that 'METALOGIC' must appear somewhere inside the chosen codefile name.

## SUPERVISOR

This is a DCALGOL codefile named **METALOGIC/SUPERVISOR** that, in the TRIM world, gathers data about tapes, directly accepts ODT input and runs OPAL programs. It is the primary supplier of tape-related information to the TRIM system and provides the transport mechanism used to handle specific TRIM enquiry or update operator commands.

SUPERVISOR is the core of the system and helps to provide powerful functionality to make TRIM unequalled by any other A-Series Tape Management system. Tape unit related information is dynamically received from the MCP as tapes are scratched, used or created.

This information is tracked in real-time by Supervisor and processed, passing tape events or "notices" directly into the TRIM system software. Further, Supervisor can run synchronisation programs that analyse tape information from the system SUMLOG in the event of a halt-load or restart, to ensure that no important tape events have been missed.

Only if the Supervisor TAPELIB option has been set (see the **Metalogic Supervisor Reference** manual for more details) will tape event information be automatically written to the Tape Library database (see METATAPELIB).

Also, Supervisor can feed this tape event information directly into one or many active OPAL programs. OPAL is a very powerful scripting language specifically designed for the problems of programming operational software and Metalogic supplies many example programs that can be customised or re-written by the user. Typically, these programs can be used to produce printed tape labels or filter out specific tapes or serial numbers from being written to the METATAPELIB database.

### TAPELIBUPDATER process

This process runs under the control of the Supervisor program and is called METALOGIC/SUPERVISOR/TAPELIBUPDATER. It receives tape event information from Supervisor, updating the METATAPELIB database using library entryptoints in the OPALTAPELIB library, and also processes operator inquiry/update commands via Supervisor's **TP** command interface. Note that **TP** is a prefix used to route commands from Supervisor into the TRIM software.

Usually, TAPELIBUPDATER will wait indefinitely for SUPERVISOR to insert **TP** commands or tape notices into its input queue.

If the input message is a tape event, the information provided by Supervisor is very comprehensive and is stored immediately on the database. If there are any tape expiration or generation "rules" available in the database then the first TRIM rule, matching owner, tape volume and file identifiers, will be applied automatically assigning any expiry or generation controls.

### MAGUS library

The SL-ed MAGUS library, usually called \*METALOGIC/MAGUS, is a sophisticated NEWP program which provides extra functionality to the TRIM package, not usually available in the DCALGOL language. MAGUS also controls Metalogic's software key licensing and none of the Metalogic software will function correctly if this library is absent or not SL-ed.

## METATAPELIB database

METATAPELIB is an audited DMSII database for storing the tape information gathered by SUPERVISOR. It has numerous structures allowing fast access to a wide variety of tape-specific data. METATAPELIB maintains all relevant database information from the tape event plus appropriate timestamp information that is passed by Supervisor. Supervisor's **TP** interface allows many database items to be manually updated directly usually by terminal input or via the OPAL programmatic interface.

Ad hoc inquiry, update and reporting of tape information is achieved primarily by the **TP** command interface and will soon be available using the TRIM utility. Updates and inquiries can also be done with any other DMSII reporting tool including ERGO or DMINQUIRY.

METATAPELIB keeps the history of earlier generations of a tape volume; that is, if a volume changes, the previous information is kept and the oldest label data for that tape are deleted if the limit on generations has been reached (currently this is 5).

If the Unisys CATALOGING implementation is active (MCP option CATALOGING must have been SET) and the Metalogic FLEX package is available, then the METATAPELIB database will hold additional information from the system's Volume Library and provide interfaces to the FLEX software.

## LOGREADER process

The LOGREADER utility is an external codefile, named **METALOGIC/SUPERVISOR/LOGREADER**, which is invoked automatically by Supervisor after restarts or a halt-load. LOGREADER is responsible for scanning the system SUMLOGs, searching for any tape information that may have missed whilst Supervisor was not running. Usually, this can occur after a halt-load or a scheduled SUPERVISOR outage. It is **always** automatically invoked during each Supervisor restart.

LOGREADER can also be invoked stand-alone to read any selected SUMLOG. In such cases, ALL tape event information extracted from that log will be processed and, where applicable, the METATAPELIB will be updated.

## TRIM utility

The TRIM utility provides the ability to create tape retention “rules” allowing the application of expiration, generation and movement controls on all or a subset of tapes in the Tape Library. These rules are held in the METATAPELIB database, are simple to maintain and use combinations of owner, volume and file identifiers (with wildcard options) to uniquely identify tapes that qualify for the retention component of the rule. There are a number of small standard OPAL routines that are supplied to assist in the reporting of tape movements. A small subset of **TP** commands (e.g. **CONFIRM**) are used to report and authorise tape movement and purge eligibility.

Currently, the TRIM utility only controls the rule-handling components. Scheduled enhancements are planned which will provide an alternative interface to METATAPELIB tape information through many tailored reports and an enhanced inquiry/update interface.

## TAPEMANAGER library

TRIM provides functionality within the standard Unisys TAPEMANAGER implementation that allows all operator- or program-initiated **PURGE** and **SN** commands to be validated first by a tape verification procedure. In such cases, the interface provided between TRIM and TAPEMANAGER can verify that a tape really has expired before permitting it to be purged. TAPEMANAGER will check the serial number entry in the METATAPELIB database to ensure that the tape is not only PGOK but that tape information is correct (e.g. checking that the tape name of the physical tape matches the database entry). This interface greatly reduces the probability of accidental tape purges through operator error. Please **Chapter 7: TAPEMANAGER implementation** for more details.

## OPALTAPELIB library

OPALTAPELIB is also an SL-ed library and is responsible for the control and synchronised access to tape information held in the METATAPELIB database. All DMSII functionality is held within this library which allows interpretative access from Supervisor’s OPAL programs, responds to **TP** commands, interfaces to the TAPEMANAGER library and, if available, the FLEX suite of programs. This library codefile, usually named \*METALOGIC/OPALTAPELIB, is automatically recompiled by the INSTALL utility whenever the TAPECONTROL or TRIM module is installed.

### TAPELOG file

All database update and inquiry actions are logged to a data file maintained by the OPALTAPELIB library. This file, called TAPELOG, records all operator **TP** commands, tape activity and events. It is readily accessible via the **TT TP LOG** command from COMS, MARC or the ODT and has extensive filtering capabilities if one kind of message needs to be tracked.

### SUMLOG file

The system SUMLOG file is only accessed by TRIM software in the event of a H/L or restart of the SUPERVISOR program. As stated earlier, the LOGREADER process is responsible for scanning the SUMLOG, searching for missed tape event information.

### VOLUME Library

If the MCP option CATALOGING is set and the Metalogic FLEX package is available, then various TRIM commands may be used to update the system Volume Library with additional information about the status of an individual tape. For example, using the **TP <serial> OFFSITE** command will also mark the tape as OFFSITE in the Volume Library.



## User interfaces

There are a number of methods of passing information in and out of the TRIM system all of which use SUPERVISOR as the primary transport mechanism.

From the ODT, SUPERVISOR will accept input via “TT” ODT commands and return the responses to the originator. This interface has been carefully designed to maintain the conventions of normal system ODT commands by, for example, respecting the specifications of the unit's TERM specifications such as FIRSTLINE and RESPONSE.

Alternatively, as with other MCSes, Supervisor will also respond to commands issued using the <mix number>SM ODT command.

From datacom stations (i.e. COMS windows), this same operator interface is available again using the TT prefix. Also, virtually all other normal ODT commands are available making the standard Supervisor command set into a superset of the standard Unisys REMOTESPO. See the following sections **Chapter 3: Getting started with TRIM**, **Appendix A: Setting up user interfaces** and the **Metalogic SUPERVISOR Reference Manual** for more details.

Alternatively, a MARC Directive capable code file is available on the METALOGIC release tape. This allows the use of the TP command directly from a MARC Action line. The DIRECTIVE can be established with security controls to restrict the usage of TP commands to, for example, users with both system and privileged capabilities.

The MARC Operations Guide give a more complete description of this command and the DIRECTIVE implementation is also discussed in more detail in **Appendix A: Setting up user interfaces**.

## TRIM Recovery

Due to the excellent recovery handling system built into DMSII, the loss of data due to program fault, halt-load, etc. is greatly minimised. As long as the METATAPELIB4 database is backed up once a day or more, the risk of losing tape information is low.

In the event of the SUPERVISOR program being unavailable for some reason or it has been restarted after a halt-load or a QUIT, the LOGREADER utility will be automatically run to search system SUMLOGs for missed tape activity, which is then used to update METATAPELIB.

SUPERVISOR maintains an internal timestamp indicating the last time that the database was updated so that LOGREADER can begin reading the SUMLOG from beyond that date and time. This optimisation greatly decreases SUPERVISOR recovery time and avoids the re-processing of data already successfully captured in the database.

Note that the TRIM does **not** otherwise access the SUMLOG during normal operations. All tape-related event information is received directly from MCP events.

In the event of a fault with the TAPELIBUPDATER task, Supervisor will automatically restart it if tape information or TP commands are currently waiting.

## Multi-host environments

By default, TRIM functions in a single-host environment, one database per machine. This mode of operation is controlled through two configuration variables that are maintained using the INSTALL utility.

These variables, **TL\_MASTER** and **TL\_DBHOST**, can also be used to control TRIM "slave" A-Series hosts which are connected to a "master" system via BNA or TCPIP. TRIM allows simple configuration of the network using the TP NET command and switching between master-slave on either TCPIP or BNA networks requires just two commands.

Whenever TRIM is installed for the first time on a new system, INSTALL will ask whether the system is to be a slave or master. These slave systems do not have a local copy of the METATAPELIB database but instead pass information directly to a master. Only slave and master systems can communicate; no slave-slave or master-master communications are permitted.

As long as BNA or TCPIP connectivity exists and the above mentioned configuration variables have been established, no other actions are required to achieve a master-slave environment. Note that copies of a TRIM capable Supervisor **must** be running on both master and slave systems.

As SUPERVISOR handles tape activity on slave systems, the TRIM system will send tape information via port files to the master system. The tapes will be marked in the database with a HOSTNAME attribute to differentiate them from locally produced tapes. Certain **TP** interrogation commands will return responses across the network.

If you have not already done so and wish to run TRIM in a 'master-slave' environment, you should refer to **Chapter 8: TRIM and BNA networks**.

## Chapter 3: Guided Tour: Getting started

*Are you sitting comfortably? Then we'll begin.*

Julia Lang, Listen with Mother, BBC Radio programme, 1950–1982

This chapter is intended to guide new users into an introduction into the usage capabilities of the TRIM Tape Library system. It is assumed that the Metalogic software has been successfully installed, SUPERVISOR is now active, and the METATAPELIB database has been recording new tape creations, scratches and accesses.

To determine whether this last statement is true, access to information held in the Tape Library is necessary. Various operator interfaces from the ODT, Supervisor RemoteSpots, MARC and COMS windows are available for the retrieval of such data, direct from the METATAPELIB database.

## Communicating with SUPERVISOR

The tour of TRIM starts with the facilities available to the operator in the machine room. You may recall that the SUPERVISOR user interfaces have transports into the TRIM Tape Library system and communication from the ODT is just one of these interfaces. SUPERVISOR can handle input in two forms from the ODT: the easiest way is to prefix all Tape Library commands (denoted by the identifier '**TP**') with the text '**TT**'.

In all Unisys systems, the '**TT**' command always gives the response 'TEST MESSAGE' however SUPERVISOR can also detect its usage and regards any text following '**TT**' as command input for itself. SUPERVISOR responses will usually be displayed a short time after the '**TEST FUNCTION**' message is displayed on the ODT's FIRSTLINE specification.

On occasion it might be possible for Supervisor to respond to a **TT** command so quickly that the response is overwritten by the **TEST FUNCTION** message. Typically, this problem does occur with single line informational responses from Supervisor (e.g. syntax errors) and so, to avoid this problem, one-line messages are automatically displayed at **TERM** setting for **FIRSTLINE+1**.

For example, response to a **TT HO** command might appear as:

```
TT HO XXXXXX
TEST FUNCTION
XXXXXX BASE UNIT NOT FOUND
```

Try the following simple command, '**TT WS**', which returns useful version information about Supervisor and should generate a response similar to that shown below:

```
TT WS

----- METALOGIC SUPERVISOR Mix 31250 -----
SUPERVISOR Version 48.480.65 Compiled at 07:54:10 on 04/03/03
Attributes Version 48.480.12 Modified at 07:46:40 on 04/03/03
OPAL      Version 48.480.22 Compiled at 07:48:51 on 04/03/03
MAGUS     Version 48.480.23
MCPs from Version 44.0 to 48.999 are compatible with this codefile

---- LICENSE DETAILS ----
Installed modules : SUPERVISOR   Expiry date : 02/06/2003
                  LOG CONTEXTS  Expiry date : 02/06/2003
                  TRIM          Expiry date : 02/06/2003
                  ODT CONTROL   Expiry date : 02/06/2003

---- RUN-TIME DETAILS ----
Maximum WHEN slots : 40
Task PRIORITY      : 50
Monitor session    : Inactive
Compile-time option DEBUG is SET
```

Alternatively, in the event that '**TT**' is unavailable for some reason from the ODT, any message input preceded by a <mixno> SM, where <mixno> is the mix number of the METALOGIC/SUPERVISOR program, will be passed directly to SUPERVISOR. Any generated response will always be returned to the 'master spo', a selected ODT number specified through the '**TT USE ODT**' command (see the **Metalogic Supervisor Reference** manual for more details).

Outside the machine room a SUPERVISOR REMOTESPO or COMS window must be used. These interfaces permit direct access to SUPERVISOR from remote datacom stations.

Today, use of SUPERVISOR COMS windows is the preferred option because there are many security controls that can be imposed on usercodes and stations, though COMS utility, preventing unauthorised access to SUPERVISOR or TRIM. This aspect is important because the window interface provides very powerful ODT-like capabilities with access to a large subset of ODT commands.

The instructions to set up a COMS window interface for SUPERVISOR are discussed in **Appendix A: Setting up user interfaces**. If you are intending to access SUPERVISOR in such a way, please take time now to read this appendix, set up the window in COMS utility and return here when you have done so. Assuming you have access to COMS utility, it should only take 5 minutes to set up the window.

Once the COMS utility changes have been introduced, you can easily access the SUPERVISOR window:

#### **?ON SUPERVISOR**

If this is the first time you have been on the window, the user will be greeted by the '**TT WS**' response seen above. You are now ready to use TRIM and SUPERVISOR commands. .

Alternatively, a limited **TT REMOTESPO** command is available to provide a similar capability to that provided by the Unisys REMOTESPO command. The Metalogic REMOTESPO commands also takes a station name, for example:

#### **TT REMOTESPO ET25**

This command provides RemoteSpo capability to the COMS station, ET25. This will generate a status line message on the terminal

#### **Message received on MARC/1**

You can access the REMOTESPO dialog by using the COMS command:

#### **?ON REM0001**

Where you must input some text to truly open a dialog with SUPERVISOR. Try transmitting a space and you should get the '**TT WS**' response as before.

Note that the station name provided can be a fully specified COMS dialog name if needed, e.g. ET25/CANDE/1 is a CANDE session. The REMOTESPO command can be actioned at any ODT, COMS window or via the DCALGOL DCKEYIN intrinsic.

In current software, only one current REMOTESPO session is permitted at any time. However, up to a maximum of 15 SUPERVISOR-COMS window sessions can be run concurrently and this interface is **strongly recommended** for accessing the TRIM Tape Library.

## Tape Events

*An event has happened, upon which it is difficult to speak, and impossible to be silent.*

Edmund Burke, politician and man of letters, speech 1789

SUPERVISOR is an event-driven program, triggered by many events provided by the MCP as well as various internal processes. Once the SO+TAPELIB option is activated, SUPERVISOR will automatically record all tape activity, including usage, into the METATAPELIB4 database.

TRIM also supports pseudo-tape event information from volumes created by FLEX and COPYWRITE i.e. the COPYWRITE software send tape label events to SUPERVISOR for Disk Farm, CD and CD image volumes.

The primary MCP events assist TRIM in recording tape activity are discussed below:

### TAPELABEL event

The MCP generates a **TAPELABEL** notice whenever a new tape is created on the system. This information is analysed and supplemented by system and program related data extracted by SUPERVISOR at run-time and is passed to the TRIM system for database storage.

The following extract from the TRIM TAPELOG (using the command **TP LOG**) shows entries for a new tape plus FLEX CD Image and Disk Farm volumes.

TP LOG

----- 13/07/04 LOG -----

21:33:19 Evt: DISK FARM [DF0005] (FLEX)DISK\_A (to IAN)  
21:11:52 Evt: CD IMAGE [CI0023] (FLEX)DEV/CI0023F2004195A  
20:30:39 Evt: CREATED [000005] (FLEX)FLEXCOPY2004195A

SUPERVISOR detects all TAPELABEL events by default and tape information is automatically written to METATAPELIB4. However, OPAL scripts can intercept any **TAPELABEL** event allowing additional site controls e.g. to exclude tapes that should not be logged on the database.

### LOG VOLUME (Log MajorType 15, Minor Types 3, 5) event

With the arrival of large capacity tape drives which can hold very large databases on a single reel, it is now very likely for database audit dumps, produced whilst an on-line dump is still running, to become linked to the PREVIOUS completed dump.

This is because the TRIM system only updates the METATAPELIB4 database once the on-line dump is complete. Although this problem relies on the customer using "linked" rules, this is a very common of handling audits and Metalogic have decided to address this important issue in the following way.

Instead of just tracking tape creation at label time, SUPERVISOR now detects specific LOG VOLUME events for new tape volumes that have just been opened for output. Tracking LOG VOLUME events also allows the capture of tape purges (Minor Type 3) which are also logged into the database.

A "raw" entry is written into METATAPELIB4 that is assigned a creation date and time for the tape reflecting the open time. The entry is almost complete with TRIM expiration rules automatically applied but both job and task name information will be marked as "**TAPE\_IN\_USE**".

When the tape creation notice is eventually received from the MCP (at the time the tape is physically closed), the job and task information are now updated with correct names but creation date and times are NOT altered. TRIM rules are not re-applied if a valid rule has already been assigned.

The event entries in the TAPELOG appear as

```
01:39:15 Evt: CREATED [000026] DEV99021A by FLEX
01:30:18 Evt: NEW VOL [000026] DEV99021A by FLEX
```

In the above example, the response for a 'TP 26' command would show the creation time to be 01:30:18 whereas on early TRIM versions, 01:39:15 was displayed.



This is an important change to the way that the TRIM system tracks tapes as they are created on the system. Where the application creating the tape is aborted before completion, the "raw" entry will remain in the database unless the caller does a programmatic CLOSE with PURGE.

An example of this is a normal COPY JOB, which was terminated before any files were copied to the tape; in this case, the MCP automatically purges the tape as part of its clean up.

### TAPEDB event

In the next phase, the TRIM system searches the rules database attempting to match tape details with the patterns for each rule. When a match is found, the rule's expiry and retention criteria are applied and the now revised tape information is written to the METATAPELIB database.

Following this, a second event, the **TAPEDB** notice, is generated by the TRIM software and duly despatched to SUPERVISOR. This is essentially an echo of the original **TAPELABEL** notice but has much more detailed information, including attributes associated with the newly assigned rule such as EXPIRYDATE, PENDINGLOCATION etc.

These additional date fields in the **TAPEDB** notice make it ideal for reporting purposes e.g. generating printed labels or for applying user-specific changes to a tape entry on the database, after it has been created.

Both of these events can be tracked by SUPERVISOR using OPAL programs offering extra filtering and reporting facilities. Examples of OPAL programs driven by these specific events are discussed in more detail in **Chapter 6: Getting started with OPAL**.

## TRIM preliminaries

If you are using the TRIM system for the first time, it is advisable to check several aspects of the installation. Since not all SUPERVISOR users use the TRIM module, it is advisable to check the SUPERVISOR option **TAPELIB**. This option should be automatically set during a first installation and is needed to permit access to the METATAPELIB database. The option must also be set on any “slave” SUPERVISOR hosts which are sending tape notifications to their “master” system

**TT SO + TAPELIB**

If you normally use the US form of dates (MM/DD/YY) then enter:

**TT SO + USDATES**

Dates will be displayed in US format on all tape interrogations and reports and will be expected when modifying dates through the **TP** interface.

In the event that the TAPELIB option is reset, tape creations and purges will not currently be recorded into the METATAPELIB database. By setting the TAPELIB option and immediately restarting SUPERVISOR using the QUIT command, the LOGREADER utility will be automatically processed during the restart and will search for all tape events in the current SUMLOG. From the ODT or SUPERVISOR window:

**TT SO + TAPELIB**  
**TT QUIT SUPERVISOR**

The LOGREADER utility will issue a display message indicating how many tapes had been processed.

Alternatively, you might like to create several tapes of your own so that we can continue with the tour. Please take some time to do this now if possible.

## Introducing TP commands

Access to tape library information is very simple using the **TT TP** command. The **TT** prefix is mandatory from ODTs and optional from COMS windows or REMOTESPOs. The **TP** prefix tells SUPERVISOR that this is a Tape Library only command and is passed to the TAPELIBUPDATER task for processing. TAPELIBUPDATER will perform the action and route any response back to SUPERVISOR who will display it back to the originator. For example:

**TT TP VER**

The **VER** or **VERSION** command will usually return useful information about some of the TRIM-specific programs and the general TRIM environment. Typically the following response is returned:

```
TP VER

Metalogic OPALTAPELIB library 50.500.003
  Compiled 12:30:12 on 09/07/2004
Metalogic TAPELIBUPDATER module 50.500.005
  Compiled 13:21:43 on 06/07/2004

TRIM Networking Configuration:
  MASTER system
  Using native TCPIP as network provider (port #44444)
  Connected slave hosts:
    #1 DELL8500MCP (10.0.0.16, SEG=1456)
    #2 METAMCPA (10.0.0.20, SEG=1456)
  METATAPELIB database is ON-LINE

Runtime options:
  USDATES      : RESET
  CATALOGING  : SET
  LIBRARIAN    : TAPELIB
  Visibility   : ALL      Access : MODIFY
TAPELOG title is '(TAPELIB)METATAPELIB/TAPELOG ON DEV'
```

This system is running as a “master” TRIM host, currently supporting two “slave” hosts, DELL8500MCP and METAMCPA, in a TCPIP environment. The TCPIP connections are using port 44444 (which is the default).

The Tape Librarian usercode is TAPELIB and the USDATES option is RESET.

Information is also provided about the TAPELOG file, the METALOGIC/OPALTAPELIB library codefile and the current version of the DASDL used to compile the METATAPELIB database. This information is often very useful when discussing any software problems with Metalogic support personnel.

Let's look at some real tape information. For example, to look at the tapes that have been made so far today, enter the command:

**TT TP TODAY**

The TRIM system will return a list of tapes, in reverse chronological order, of all tapes made in the last 24 hours. An example response is shown below

```
TT TP TODAY

----- FIND: DATE after 11:55:31 on 25/05/2002 ----- Location ----- Created ---
000001: (FLEX) TESTTAPE/FILE000                                11:38 25/05/2002
000008: (FLEX) DBSPACK95144A/FILE000                          11:35 25/05/2002
R70846: (FLEX) META4840108/FILE000                          11:35 25/05/2002
R70845: (FLEX) META4840108/FILE000                          11:26 25/05/2002
R70844: (FLEX) META4840108/FILE000                          11:13 25/05/2002
R70843: (FLEX) META4840108/FILE000                                OFFSITE 23:07 24/05/2002
R70842: (FLEX) META4840108/FILE000                                OFFSITE 22:55 24/05/2002
R70703: (FLEX) META4840108/FILE000                                OFFSITE 21:25 24/05/2002
BOB001: (TAPELIB) METATAPELIB4                                1:1 TAPE RACK 17:32 24/05/2002
000093: (FLEX) DBSPACK02143A/FILE000                                TAPE RACK 17:16 24/05/2002
000061: (FLEX) DBSPACK02143AN/FILE000                          TAPE RACK 16:13 24/05/2002
000056: (FLEX) DBSPACK02143AM/FILE000                          TAPE RACK 15:36 24/05/2002
000038: (FLEX) DBSPACK02142A/FILE000                          TAPE RACK 12:40 24/05/2002
```

Each line of tape information displayed consists of tape serial number, full name of the volume (including the owner i.e. the usercode of the task that created this tape), date and time of creation and the physical location of the tape. In the example above, this field is either blank or displaying locations OFFSITE and TAPE RACK. All the tapes that show this location were created the day before. Note the '1:1' string prior to the location for tape serial number BOB001; this is a database dump tape and displays the cycle and version of the dumped files.

Unless you have already implemented some tape "rules" to control movement and retention and you are authorising pending movements (using the **TP CONFIRM** command), it is extremely unlikely that the LOCATION field will be anything other than blank. However, it is very likely that a "pending location" of EXCEPTION will be set up for this tape since this indicates the first physical location to which it should be moved. By default, if no rules exist or the tape does not match any existing rules, new tapes are given a default rule identifier of EXCEPTION. Note that location movements should generally be controlled by rules though they can be overridden manually.

Individual tape information can be interrogated by using a serial number following the **TP** command. For example, say we wanted to examine tape "00001" in more detail, the following would be entered:

**TT TP 1**

Note that where there are leading zeros on a serial number, they are NOT required in the command, hence the use of "1" instead of "000001". Our response shows:

```
TT TP 1

----- METATAPELIB: MT [000001] -----
Volume :TESTTAPE/FILE000
Cycle      : 1                      Reel      : 1
Version    : 0                      Density    : BPI1250
Owner      : FLEX
Location   :
Creation info: 11:38:03 on 25/05/2002
Savefactor : 30
Host info  : STIRLING1 #6343
Expiry date is 24/06/2002
Pending location : EXCEPTION          Pending date : 24/05/2002
Volume last accessed : 23:16:03 on 23/05/2002
Created by task 6446 : *LIBRARY/MAINTENANCE
Owned by job no 6445 : DUMP/DBSPACK/95143A
Usage count  : 6 times since first entered

Rule info  : EXCEPTION
```

Note that the above display reflects the current information held in the database about this tape. Most of the fields you can see are self-explanatory but some require further clarification.

Without any rules in the TRIM database, this will be a typical response to a database enquiry on a new tape. Even with no rules, as stated earlier, TRIM provides a default rule, EXCEPTION, which has one default location called EXCEPTION and will use the tape SAVEFACTOR specification to assign an expiry date. Note that the location EXCEPTION is not immediately assigned; it is first marked as a pending location and is assigned a "pending date" set to the current date i.e. the creation date of the tape.

Most of the other attributes are accessible at all times. The VOLUME LAST ACCESSED information refers to the last time that this tape was mounted on a unit or physically accessed by a program. In the event of the latter, the USAGE count field will be incremented by one.

The NOTES field is not usually updated by the TRIM software, with one exception (using the **TP <serial> SR** commands), but is available for site usage to provide extra information about individual tape volumes. The text may be changed through the **TP <serial> NOTES** command.

If the provided serial number was invalid, for example trying to use the command 'TP 0009X' the following response will be returned:

**Error:Volume 00006X not found in METATAPELIB**

It would be useful now to inspect a tape that has a valid rule already assigned. Although this may not be possible at the moment, it is worth taking a quick detour to look at how some of the tape attributes have changed. For example, consider the following tape with the serial number "000093":

```

TT TP 93

----- METATAPELIB: MT [000093] -----
Volume :DBSPACK95143A/FILE000
Cycle       : 1                      Reel       : 1
Version    : 0                      Density    : BPI1250
Owner      : FLEX
Location   : EDINBURGH              Sent date  : 24/05/2002 at 23:30:01
Creation info: 17:16:03 on 24/05/2002
Savefactor : 30
Host info  : STIRLING1 #6343
Pending location : SCRATCH POOL      Pending date : 31/05/2002
Expiry date is 31/05/2002
Volume last accessed : 23:16:03 on 23/05/2002
Created by task 6446 : *LIBRARY/MAINTENANCE
Owned by job no 6445 : DUMP/DBSPACK/95143A
Usage count  : 6 times since first entered

Rule info  : (FLEX)DBSPACKDUMP
Notes     : DBSPACK DAILY DUMP

```

A TRIM rule has been applied to this tape, called DBSPACKDUMP, which is owned by the FLEX usercode. As you can see from the display, there is an expiry date assigned, the DBSPACKDUMP rule has specified a seven days retention period from the date of creation. It is currently in an off-site location, EDINBURGH, but it has a PENDING LOCATION of SCRATCH POOL. The PENDING LOCATION marks the next destination for this tape and, in this case, it will be marked as PGOK and assigned a final location of SCRATCH POOL on the date, 31/05/2002.

Note that EXPIRY date and PENDING date are the same and this is generally always the case prior to expiry. The SENT date indicates the date and time that the last location change was authorised i.e. when this tape's movement to EDINBURGH was confirmed.

If the tape is subsequently re-used, some of the information about the original volume can still be seen by using the @ variant of the **TP** command. For example:

```
TT TP 93 @1

----- METATAPELIB entry for MT [000093] @ 1 -----
Volume   : DBSPACK02083A/FILE000
Owner    : FLEX
Creation info: UNKNOWN
Host info : #6343
Created by task 4811
This label was overwritten 18:21:49 on 21/04/2002

Marked as : PGOK, VOLUMED
```

The **@n** syntax requests information about the *n*th previous usage of this tape. Thus **@1** specifies the one before the most current generation. If no "@" modifier is given, the most recent generation is always assumed.

Note that if the tape is PGOK or had been scratched by the TRIM system prior to re-use this would be as expected if the TRIM system is functioning correctly. The number following the @ symbol can be considered as the **generation** number.

Another very useful asset is the **TP LOG** command. This allows access to the TRIM TAPELOG file to examine tape events, interrogation and modifications to tape information and general error messages. Information is returned in reverse chronological order and a typical response is shown below.

```
TT TP LOG

----- 25/05/02 LOG -----
10:58:18 ODT: SCR 1250
10:52:22 ODT: FIND RULE (FLEX)TEST
10:50:58 ODT: FIND RULES
10:30:12 Evt: CREATED [R70847] META4040108 by FLEX
10:29:11 Evt: SCRATCH-ed [R70847] not PGOK
10:28:59 Msg: ALLOW SN OF MT 49 TO R70847
10:19:30 Evt: CREATED [R80202] META4040108 by FLEX
10:09:38 Evt: SCRATCH-ed [R80202] not PGOK
10:09:23 Msg: ALLOW SN OF MT 49 TO R80202
08:46:09 Evt: CREATED [BOB001] METATAPELIB by TAPELIB(not SCR)
08:45:51 Msg: Usage [BOB001] METATAPELIB
08:45:05 Msg: Usage [BOB001] METATAPELIB
08:44:30 Err: DENY PG OF BOB001 (Not PGOK)
08:25:19 ODT: R80201
08:23:36 Evt: CREATED [R80201] METATRI4040109 by META
----- 24/05/95 LOG -----
23:31:28 Msg: Volumes confirmed 3 (warnings 0),Links 0,Errors 0
23:31:15 ODT: CONFIRM
```

Each entry in the TAPELOG has it's own timestamp and message category. For example, tape event information is categorised by the key word "**Evt**", general messages by "**Msg**" and operator commands by "**ODT**". The **TP LOG** command can filter using these key words so try, for example:

```
TT TP LOG EVT
```

```

----- 25/05/95 LOG -----
10:30:12 Evt: CREATED [R70847] META4040108 by FLEX
10:29:11 Evt: SCRATCH-ed [R70847] not PGOK
10:19:30 Evt: CREATED [R80202] META4040108 by FLEX
10:09:38 Evt: SCRATCH-ed [R80202] not PGOK
08:46:09 Evt: CREATED [BOB001] METATAPELIB by TAPELIB(not SCR)
08:23:36 Evt: CREATED [R80201] METATRIP4040109 by META
----- 24/05/95 LOG ----
16:19:22 Evt: CREATED [R80200] META4040107 by FLEX

```

The **TP LOG** command currently will retrieve information up to 72 hours old in the TAPELOG file. Further extensions are planned to allow more flexible date searching and to access off-line TAPELOGs.

If any SCRATCH tapes have been created whilst on this guided tour, they can easily be seen by using the **TP SCRATCH** or **TP FINDSCRATCH** commands. If you have purged any of the tapes that you created earlier, they will appear in this list, categorised by tape density. A typical response to this command is shown:

```
TT TP SCRATCH
```

```

----- FIND: PGOK (1250) ----- Location ----- Created ----- Expired
000011:DBSPACK95135A/FILE000      SCRATCH POOL 00:00 15/05/2002 23/05/2002
000017:DBSPACK95137BN/FILE000     SCRATCH POOL 15:18 17/05/2002 23/05/2002
000019:DBSPACK94320A/FILE000      SCRATCH POOL 10:49 16/11/1994 23/05/2002
000031:DBSPACK95083A/FILE000      SCRATCH POOL 21:46 24/03/2002 23/05/2002
000042:DBSPACK95060A/FILE000      SCRATCH POOL 21:50 01/03/2002 23/05/2002
000044:DBSPACK94315AM/FILE000     SCRATCH POOL 16:23 11/11/1994 23/05/2002
000051:DBSPACK95082A/FILE000      SCRATCH POOL 21:48 23/03/2002 23/05/2002
000055:DBSPACK95090A/FILE000      SCRATCH POOL 08:15 03/04/2002 17/05/2002
000063:DBSPACK95061A/FILE000      SCRATCH POOL 08:48 03/03/2002 23/05/2002
000077:DBSPACK95115A/FILE000      SCRATCH POOL 21:37 25/04/2002 23/05/2002
000084:DBSPACK95114A/FILE000      SCRATCH POOL 21:39 24/04/2002 23/05/2002
000096:DBSPACK95109A/FILE000      SCRATCH POOL 21:43 19/04/2002 17/05/2002
000099:DBSPACK95110A/FILE000      SCRATCH POOL 21:38 20/04/2002 17/05/2002
000100:DBSPACK95130A/FILE000      SCRATCH POOL 21:35 10/05/2002 23/05/2002

----- FIND: PGOK (1600) ----- Location ----- Created ----- Expired
BLLOG7:CMIGTAPE/FILE000          SCRATCH POOL 00:00 20/09/1993 19/12/1993
BOB055:META91303B/FILE000       SCRATCH POOL 00:00 30/10/1991 28/01/1992
BOB064:TESTPACK/FILE001         SCRATCH POOL 00:00 27/01/1993 27/04/1993
BRISTB:METAOECD/FILE000         SCRATCH POOL 00:00 09/02/1985 10/05/1985
MDUMP :MEMDUMPA12/FILE000       SCRATCH POOL 00:00 25/09/1990 24/12/1990

```



TRIM will return up to 200 tapes in this list. The subset of scratch tapes associated with each density can be displayed simply by adding the relevant value after the command:

**TT TP SCRATCH 1250**

If we want to send this report to the printer the command should be prefixed with the modifier "PRINT", e.g.

**TT PRINT TP SCRATCH**

In this case, all SCRATCH tapes will be output to a single backupfile by a PRINTER sub-task invoked by SUPERVISOR. You can see this task by examining the COMPLETED entries list or by looking at the response to a **PS SHOW USER SUPERVISOR** command (assuming that SUPERVISOR was the selected usercode for the SUPERVISOR codefile to run under.

There are many other **TP** commands available, many of which allow wildcard patterns to facilitate searches. The searching capability is optimised to be as efficient as possible, but the more information you provide in the search parameter the faster the search.

For example, to find all tapes which were made with the a name starting "TEST", use the **TP FIND NAME** command:

**TT TP FIND NAME TEST=/**

Where '=' means any length string between 'TEST' and '/'. The **NAME** command can take a usercode if needed otherwise it is ignored when checking. Most tapes are library maintenance format i.e. they have a file identifier of 'FILE000' assigned by the system. For such tapes, a trailing '=' will be required for the '**FIND NAME**' command to detect them. In the above **FIND** command, all multi-level tape names will be easily found.

The **TP** set of commands is flexible and very extensive, allowing the production of concise reports and the ability to change individual tape information very easily. This introduction gives a flavour of the kind of access that is provided into the METATAPELIB database.

## OPAL programs

Much of the flexibility of TRIM is that access to tape information is also permitted through OPAL programs and it is easy to quickly define your own tape search criteria and reporting. OPAL can be regarded as a powerful macro building facility in the same category as Microsoft's Visual Basic™.

OPAL programs are not only used to extract from or modify information in the database, they can also be used to filter and monitor tapes automatically as they are used and created. We shall be covering OPAL in more detail in Chapter 6 but some of its power is shown in this section.

In conjunction with the TRIM rules implementation, several OPAL programs are currently used to help generate the daily tape movement reports and also to provide flexible label printing. Some sites regard the printing of tape labels important, and it is only this feature that we will cover here in the Guided Tour.

OPAL programs come in various flavours. There are three program types:

**SITUATION**  
**ODTSEQUENCE**  
**DISPLAY**

The SITUATION is used to detect system events passed to Supervisor by the Operating System or MCP. The ODTSEQUENCE is usually a sequence of actions, which may be invoked when a SITUATION has been triggered. Alternatively, the DISPLAY program can be used to report information back to a screen device when the SITUATION is triggered.

In the TRIM environment, each OPAL program type has its own “**context**” in which it can respond to events and handle information. These contexts are as follows:

**PER**  
**TAPELABEL**  
**TAPEDB**

The PER context allows events and information to be handled for all or individual tape devices. The TAPELABEL context permits access to information about MCP-provided tape creations or purges.

Lastly, the TAPEDB context permits access to events and information about tapes held in the METATAPELIB database; it is usually TAPEDB programs which should be used for printing labels because the available data associated with the new tape is much more comprehensive than TAPELABEL.

Metalogic provide a large selection of example OPAL macros for site use. These macros can be entered into the TRIM system by entering the Supervisor command:

```
TT ENTER FROM OPALS/TPDB
```

This file should have already been loaded during the INSTALL process. The **ENTER** command (see the **Metalogic Supervisor Reference manual** for more details), is used to load one or more OPAL macros from an external file.

After some moments, SUPERVISOR should return some messages indicating how many OPAL programs were successfully compiled and that the **ENTER** has completed. With these programs loaded, it is now easy to start printing labels for newly created tapes.

## OPAL attributes

The OPAL language can be used to construct macros or programs, which can access information within each of the TAPEDB, TAPELABEL and PER contexts. Individual items of information, unique to each context, are known as **attributes**, and describe characteristics such as OWNER (i.e. creating usercode) of a tape, TITLE, (its full name), SERIALNO and DENSITY etc. Each of these items of information maps onto an OPAL attribute and can be interrogated or reported.

For users who are very familiar with A-Series Task or File Attributes, the concept of OPAL attributes will not be unfamiliar and, in fact, many OPAL attributes have direct equivalents to many Unisys attributes.

The Supervisor command **TT HELP ATTR** can be used to provide extensive information about all the attributes available.

The provision of a tape label printing service using Metalogic-provided OPAL programs might be a good way to show how OPAL is used. Many of the commands discussed in the next section are covered in great detail in later chapters.

## Tape label printing

Assuming the availability of a simple, cheap dot-matrix printer, printing tape labels can be done in a number of ways. The printer can be directly attached to a site ODT via the printer port or attached to a remote datacom station, using the printer-pass-through facility available from most Unisys-terminal emulations. The latter configuration requires COMS Utility support and its set-up is described at length in **Appendix A: Setting up user interfaces**.

To use the ODT-supported printer, from that ODT enter the following command:

```
TT WHEN TPDB_LABEL DISP TPDB_LABEL
```

This procedure will now result in a standard tape label whenever a tape is created on the system, very similar to the ADM EVENT PRINTLABEL mechanism available in the standard Unisys MCP. A special OPAL program type, called DISPLAY, defines the layout of this label. OPAL DISPLAY programs are used purely for formatting output directly to a terminal or print file and are very easy to build and modify.

SUPERVISOR can handle several types of system event, one of which is the TAPEDB notice and it uses a special program type, called a SITUATION, which is triggered every time a tape is created. The WHEN command used above tells SUPERVISOR to link the SITUATION and DISPLAY program together: **WHEN** a tape label notice is received, **DISPLAY** a label to the screen.

These OPAL programs can be modified to allow the label format to be changed (the DISPLAY program) or to provide some filtering (the SITUATION) to perhaps inhibit labels for certain serial numbers or tape names.

Let's change the definition of the DISPLAY program **TPDB\_LABEL**, which can only be modified when it is not running. So, to stop the WHEN permanently:

```
TT WHEN TPDB_LABEL DISP
```

To examine the layout for the **TPDB\_LABEL** DISPLAY program, enter:

```
TT DEFINE ? DISPLAY TPDB_LABEL
```

The following response (or something similar) should be returned:

```

DEFINE + DISPLAY TPDB_LABEL(TAPEDB) :
#[ESC], "$ ", /,
" METALOGIC TRIM" , /,
"TITLE  :", TITLE, /,
"JOB    :", JOBID 28, /,
"SERIAL :", REPEAT(" ", LENGTH(HEAD(SERIALNO, "0"))) , TAIL(SERIALNO, "0") ,
REPEAT(" ", 1) ,
IF(CATALOGLEVEL > 0 AND VLNAME(TAPEPE, SERIALNO) NEQ "") THEN
"REEL  : " & STRING( INTEGER(ABS(VLMEMBERS(TAPEPE, SERIALNO))) , *)
ELSE
#("REEL  : ", REEL) , " OF " , /,
"UNIT   :", UNITNO 3, REPEAT(" ", 4) , "OWNER: " , OWNER 10, /,
"DATE   :", CREATIONDATE, , "TIME: " , TIME(CREATIONTIME) ,
///, " #[ESC]"

```

This format is quite similar to the Unisys ADM EVENT PRINTLABEL facility, but OPAL has many extensions, in particular full ALGOL/WFL string and arithmetic expressions, variable handling etc. The text **TAPEDB**, in parentheses, refers to the context of the OPAL program, which controls the use of many tape-specific attributes for reporting our label (see **the Metalogic Supervisor Reference** manual for more detailed information on contexts and OPAL programming).

Alter the label's heading, replace the text "METALOGIC TRIM" with your company name, just typing text directly over the top of the string, making sure that the pair of quotes is retained around the name. Just transmit the whole of the screen just beyond the very last line. The entire OPAL macro will be re-displayed with the following text appended:

**DISPLAY DEFINITION ENTERED**

Re-run your new OPAL programs from the ODT, as before, using the WHEN command. Any new tape being created will now generate a printed label with your company name as it's heading.

You may need to do other amendments to remove extra line-feeds etc. , some experimentation may be necessary until you get the right output. Note that, in the DISPLAY program, the character '/' means start a new line (i.e. CR/LF) and the string '[ESC]' is a special ESCAPE sequence that tells the terminal or ODT to send this page of output to the printer.

If revision of the **TPDB\_LABEL** DISPLAY program is necessary to control the format of the output, then the program can be tested stand-alone without the requirement for a new tape to be created.

Assuming that tape serial number B10060 is a valid entry in METATAPELIB, then labels can be manually printed by the following command:

```
TT / TPDB_LABEL B10060
```

The "/" command requests Supervisor to execute the named DISPLAY program and, because the context of TPDB\_LABEL is TAPEDB, a valid tape serial number is also required. This can be executed, as many times as required after changes to the OPAL, until the label format is correct.

As tapes are created on the system, printed labels will be output to any dot matrix printer and will look very similar to the following layout:

```
                METALOGIC TRIM  
TITLE   : (FLEX) DBSPACK95168A/FILE000  
JOB     : DUMP/DBSPACK/95168A  
SERIAL  : B10060 REEL : 1 OF  
UNIT    : 150      OWNER: FLEX  
DATE    : 17/05/95 TIME: 21:12:51
```

Note that OPAL DISPLAY programs if linked via the WHEN command and run from an ODT, will be automatically restarted in the event of a halt-load, SUPERVISOR restart or datacom re-initialisation. Any WHENs that were run from remote stations do NOT get restarted in such circumstances. However, this problem can be easily avoided by using a COMS window to drive the labels, as discussed in **Appendix A: Setting up user interfaces**.

## The Tape Librarian

The Tape Librarian usercode is very important to the day-to-day running of the TRIM system for two reasons. First, only the Tape Librarian usercode is allowed to delete tapes from the METATAPELIB database and secondly, it has overall control of the creation, deletion and maintenance of TRIM rules.

The Tape Librarian usercode is often a privileged usercode with password and limited to selected individuals though there is no requirement for it to have PU status. However, it is important that this usercode is carefully controlled allowing access to a restricted number of individuals. The usercode is assigned by the INSTALL utility during the first installation but may be easily changed using the Supervisor **TT USE** command.

```
TT USE

      ---- USER and FAMILY settings ----
USER          : SUPERVISOR      USER for TAPELIB      : TAPELIB
USER for ODTSECURITY : META      USER for EXTERNAL     : META
USER for MAIL       : META      FAMILY for NAPLOGS      : DEV
FAMILY for EXTERNAL  : DEV       FAMILY for SAVES       : DEV
FAMILY for LOGS      : DEV       FAMILY for SYSTEM      : DISK
FAMILY for SCHEDULE  : VMSA

      ---- TASK and FILE settings ----
FILE for REBUILD     : (OPS)OPALS/SUPERVISOR/GETTINGSTARTED ON PACK
JOB after TL         : Not specified
TASK for RECORDER    : *METALOGIC/SUPERVISOR/RECORDER ON DEV
TASK for NAPSTATSLIB : (IPP)NAPSYSTEM/NSL ON DEV
FILE for NAPRUNLIGHT : *METANAP/RUNNING/LIGHT ON DEV

      ---- Miscellaneous settings ----
QUEUE              : 255          BACKTRACK mode         : NONE
ODT                 : 1           LANGUAGE                : ENGLISH WILL BE SPOKEN
WINDOWSECURITY      : MINIMUM
DESTINATION          : MAIL:BOBNIAN
```

Here, the entry associated with "User for TAPELIB" is set to "**TAPEUSER**" and indicates that the usercode TAPEUSER has been designated as the Tape Librarian. Usually, all the database files are held under this usercode including the DMSII structures and support files plus the TRIM TAPELOG log file. The Tape Librarian usercode has additional privileges when used with the Trim utility and may create or modify rules for all tapes whereas normal users may only make rules for tapes created by their own usercode.

The Tape Librarian usercode is also essential for ad-hoc deletion of tapes from the database and, in fact, this is the **only** way that tapes can be deleted using a TRIM interface. From a COMS window, for example, the usercode may be passed using the Supervisor **TT FOR** modifier or from a window that has already been established with a usercode using the TERM command.

```
TT FOR TAPEUSER/PASSWD TP B10060 DELETE
```

```
TERM USER TAPEUSER/PASSWD  
TT TP B10060 DELETE
```

Many of the other fields shown in this display are beyond the scope of this manual and do not refer directly to the maintenance of the TRIM system. Please refer to the **Metalogic SUPERVISOR Reference Manual** for more information.

## Summary

This concludes the first part of the TRIM Guided Tour. We have covered some of the interfaces available in TRIM, the creation of tapes, interrogation of tape information and the production of printed labels.

As long as access to an ODT or SUPERVISOR-COMS window is possible, accessing tape information directly from the database is both easy and flexible.

As has been stressed in earlier sections, the most important component of TRIM is rules management and this is discussed in the next chapter: **TRIM Rules Management**.



## Chapter 4: TRIM Rules Management

*L'existence précède et commande l'essence.*

L'Être et le néant (being and Nothingness), Jean-Paul Sartre

Metalogic's Tape Library system, known as TRIM (Tape Rules Information Management), allows sites to control the retention and movement of tapes for one or more systems without the need for manual intervention. These automatic controls are provided by the setting up of 'rules' that provide information about tape expiration by generation and/or expiry and the various physical off-site locations that these tapes are moved to during their lifetime.

The establishment of these rules is the most important component of TRIM; they offer flexibility and durability and can be readily tested and modified, even if a rule has been in use for a considerable length of time.

This chapter is meant to assist both new and existing users of the Metalogic Tape Library system and is intended to provide information on how to test and implement TRIM rules to control the movement and expiration of tapes.

### TRIM rules

TRIM "rules" logically controls the automatic expiration of some or all tapes in the library by the application of time and/or generation retention criteria. The rules also enforce the movement of tapes between as many as 8 on-site and off-site locations. Individual rules can be accessed and changed through the TRIM utility, \*OBJECT/TRIM, which store rules information into the RULES dataset structure of the METATAPELIB database. This utility is automatically loaded during the installation process and can be accessed by all users.

Each rule is uniquely identified by a 17-character name, optionally protected by a **Usercode** rule owner and/or RULEINDEX. The use of a USERCODE permits security access to the rule definition to only that usercode or the Tape Librarian usercode (as defined through the Supervisor **TT USE** command).

RULEINDEX can take a value between 1 and 255 and allows the use of the same rule identifier to be logically grouped for the same application which produces different tapes for daily, monthly or quarterly runs but each subset of tapes requires different retention characteristics.

RULEINDEX is usually seen following the rule identifier after the character “/”; the full RULETITLE should be considered as a file title.

It is advisable to make rule names meaningful, giving some clue as to the nature of the application or applications responsible for creating that family of tapes.

Some typical rule names are shown below:

```
EXCEPTION  
(META) TESTRULE  
(TAPELIB) DBDUMP/1  
(TAPELIB) DBDUMP/2
```

## Rule definition

Rules should only be defined using the TRIM utility: you should refer to this manual and, if necessary, the Metalogic METANOTES files, supplied on the release tape, for late-breaking information. This chapter will, in particular, provide an introduction to the creation, modification and testing of new rules.

Most tape libraries maintain their tapes by **TIME** or **GENERATION**. For time-based retention, selected tapes will expire after a specific number of days, when they are expected to return to a scratch pool and be available for re-use.

For generation-based retention, older tapes belonging to the same **RULE FAMILY** will return to the scratch pool only when they exceed the number of required generations.

### DMSII dumps example

For example, consider that the DMSII dumps for a specific database are kept unconditionally for 6 generations; creating a new generation for that database will force the newly assigned generation 7 to be marked as PGOK (purge authorised) and subsequently return to the scratch pool for re-use. In the TRIM world, '**Family**' refers to the set of tapes, which have been associated with the same rule name.

In both the above cases, it is likely that tapes will be moved to off-site locations or 'safe' storage sites. Perhaps these tapes will stay in those locations for a certain number of days before moving to their next location or, eventually, back to the scratch pool for re-use. Possibly, they may move depending on their generation number i.e. generation 1 moves to LOCATION1, generation 2 moves to LOCATION2 etc.

It is the above controlling processes that TRIM rules can reproduce and automate. TRIM rules will allow the mixing of time and generation based controls, if desired, for both retention and movement.

## RULE characteristics

All TRIM rules consist of the following characteristics:

RULE USER	Owner of the rule, blank means Tape Librarian
RULE IDENTIFIER	17-character identifier
RULE INDEX	Supplementary information for rule id
USERCODE PATTERN	17-character wildcard identifier (owner)
USERCODE EXAMPLE	Used to confirm pattern
VOLUMEID PATTERN	17-character wildcard identifier (volume name)
VOLUMEID EXAMPLE	Used to confirm pattern
FILEID PATTERN	17-character wildcard identifier (fileid name)
FILEID EXAMPLE	Used to confirm pattern

Each of the above fields map to a DMSII database field declared in the METATAPELIB4 DASDL.

Any individual user can make or change its own rules; these rules will only apply to tapes created under that usercode. In such cases, the TRIM input screens are automatically changed to make the usercode fields protected.

## Unlabelled tapes and TRIM rules

TRIM allows rules to be set up to handle **unlabelled** tapes. If a tape rule identity is created with the prefix '**UL\_**', the TRIM utility will consider that the rule will only be matched for unlabelled tapes. In this case, instead of using a Volume Id and File Id for matching, TRIM requires patterns for JOB and TASK names to be established.

Some care is needed when deciding on these patterns since the TASK name, in particular, can include 'ON <packname>' components. TRIM will only verify a new unlabelled rule against other existing **UL\_** rules.

At tape creation time, the creating JOB/TASK names will be matched only against **UL\_** rules. If no match is found, an EXCEPTION rule will be assigned as usual. Both the **SR** and **APPLY** commands will work as expected, and in fact **SR** is an excellent way of testing out new rules.

The TRIM utility will automatically output the headings 'Job Id' and 'Task Id' on the appropriate screens when handling unlabelled tape rules. Note that within each tape entry in the METATAPELIB database, only the first 28 characters are held for both task and job names. This limitation reduces the disk overhead associated with each tape record but its restriction should be remembered when handling **UL\_** rules.

Where a job or task name is greater than 28 characters, the TRIM system will attempt to make sense of the truncated name so it is strongly recommended that you check the names seen in a **TP <serial>** response.

**Examples of unlabelled rule names:**

```
UL_TEST  
(META) UL_EXAMPLE  
(FLEX) UL_TEST/3
```

## Tape Creation

When a program has successfully created an output tape, SUPERVISOR receives a tape event notice (internally called TAPELABEL) from the MCP, which is then passed to the TAPELIBUPDATER process. All relevant METATAPELIB rules are then searched, attempting to match each rule's usercode volume and file id patterns against the corresponding new tape attributes. If a matching rule is found, the tape information written to the database is modified to include an EXPIRY date (if applicable), a PENDING LOCATION, a PENDING DATE for that movement (always set to today's date) and all tape rule attributes are updated.

If no matching rule is found, then the TRIM system will check a special rule subset with a rule identifier of EXCEPTION. Initially, an EXCEPTION rule under the owner of the tape is checked; if this is not present, the non-usercoded EXCEPTION rule will be searched. If no EXCEPTION rules exist at all, then the TRIM system will arbitrarily assign a default rule identity of EXCEPTION. As you may have seen in the **Chapter 3: Guided Tour: Getting started** section, the SAVEFACTOR is used to assign an expiry date and only a single location of EXCEPTION.

If the tape is unlabelled, only rule identifiers beginning with the prefix UL\_ will be searched. For these rules, the job and task names are used in the rule matching process because, for obvious reasons, volume and file ids are not active for an unlabelled tape. The TRIM utility will automatically change the input and inquiry screens to include the headings 'Job id' and 'Task Id' when handling a UL\_ rule. The METATAPELIB database only maintains 28 characters for a job or task name; where necessary, if 'ON <PACKNAME>' is part of the name and the name length is longer than 28 characters, the ON part may be dropped in favour of '/PACKNAME'. You should be aware of these effects when building job and task name patterns. Note that UL\_ rules are never used by the TRIM system for matching normal labelled tapes.

Newly created tapes can be immediately viewed using the **TP <serial>** or **TP TODAY** commands. The TAPELOG will also show information about tape events when the **TP LOG EVT** command is used.

After the TRIM software has handled the TAPELABEL notice and a rule has been found and applied, the TRIM system generates the TAPEDB notice that is then sent back to Supervisor. Supervisor will trigger any OPAL programs that are waiting on TAPEDB events when the notice arrives.

## TRIM Utility

The TRIM utility is responsible for the generation and maintenance of rules. Logical groups (or **Families**) of tapes can be assigned similar retention characteristics allowing all the tapes output from one or more applications to be grouped under one rule name. Rules are set up according to the composite name of the tape and/or the usercode of the creating task.

New or modified rules can be easily tested and applied to individual tapes or a whole family. This allows the user to quickly verify that the changes made for a rule will work for the tapes that he or she wishes to control.

Let's run the TRIM utility now from either a datacom station under CANDE or MARC, or the ODT running under the MARC ODT handler. If you are running under the **Tape Librarian** usercode (as seen in the response to a **TT USE** command) then you will be allowed access and modification rights to all rules in the database. Any other usercode, even privileged, will only have access to its own subset of rules.

The TRIM utility codefile should be available on your normal system disk. Log on under your chosen usercode and enter:

**RUN \*OBJECT/TRIM**

Please note that all military-style dates, where applicable in this chapter, are depicted in non-US formats i.e. DD/MM/YY. This behaviour can be controlled using the Supervisor **TT SO USDATES** command.

## TRIM Utility: Home menu

After TRIM has initialised, you are presented with the Home Menu screen:

```

METALOGIC S.a R.L
Tape Rules Information Manager
Version

```

---

```

Welcome to the TRIM Utility.

Here, you can create or modify TRIM tape retention rules
or manually enter new tape information.

Please choose one of:

1. Rules Maintenance ... RULES
2. Data Entry ..... DATA
3. Quit TRIM ..... QUIT

```

---

```

Action: > < ( Enter RULES, DATA or QUIT)

```

The 'Action' field requires a valid command; for example inserting a "Q" or "QUIT" in this box will terminate the program.

Entering 'RULES' will navigate the utility to a screen where TRIM expiration and retention rules can be created, modified or deleted.

```

METALOGIC S.a R.L
METATAPELIB Rules Maintenance
Version 49.490.01

```

---

```

Please select the required service:

Action
( Add, DElete, INquire, MODify, Quit )

RULE DETAILS

```

---

```

You must provide some information on the RULE you
if this is a new entry:

Rule usercode ..... FLEX
Rule identity ..... > <
Rule index ..... > <

```

---



Entering 'DATA' will link to a Data Entry form where new tapes can be manually added into the METATAPELIB database.

In the previous case, the example screen shown is that which would be presented for the usercode FLEX, where FLEX is NOT the Tape Librarian usercode. As you can see, the Rule usercode field has been pre-filled and cannot be changed. INQ or INQUIRY is used to inquire on an existing Rule. If the 'User Id' and 'Index' fields are left blank information about the first FLEX rule that exists in the database will be returned.

The fields are explained as follows:

### Rule usercode

This is the “owning” usercode of the rule, which will be applied to all tapes belonging to that owner that satisfies this rule. The OPAL attribute RULEOWNER will return this value when interrogated for a particular tape. The exception is the Tape Librarian usercode who can create rules, which apply to multiple usercodes.

### Rule identity (rule identifier)

This is the user-selected unique name for the rule that controls tape retention and locations for specified set of tapes. The OPAL attribute RULEID will return this information for an individual tape, as well as being seen in the response to a **TP <serial>** command.

### Rule index

This optional field allows more than one tape rule to be grouped under one rule identity. For each defined rule belonging to this identity, the index field must be incremented by 1. It is the combination of all three of these fields that comprise the term **rule**. Most rules are usually composed of a rule owner and a rule identifier; the rule index is not usually required though it may be used to help group closely related rules and their tape families.

## Creating rules

Next, type **ADD** in the action field and **TEST** in the Rule identity field and transmit. The following screen will be displayed.

METALOGIC S.a R.L									
Metatapelib Add Rules									
(FLEX) TEST									
User/Owner			Volume ID			File ID			
User/Owner	Pattern	FLEX		and Example	FLEX				
Volume Id	Pattern	<	>	and Example	<			>	
File Id	Pattern	<		and Example	<				>
Hostname	Pattern	<	>	and Example	<				>
<u>RETENTION DETAILS</u>									
Expires after :		<	>	Days, Years:	<	>	Or Generations :	<	>
				Save, Never					
Location	<	>	Duration in	<	>	Duration units	<	>	
itinerary :	<	>	location:	<	>	(days or gens)			
	<	>		<	>				
	<	>		<	>				
	<	>		<	>				
	<	>		<	>				
	<	>		<	>				
<u>LINKAGE INFO</u>									
User	<		>	Identity	<		>	Index	< >
<u>Action :</u>									
< >									
( Return Next Quit )									

This is the main TRIM utility screen responsible for the creation of rules.

Since FLEX is not the Tape Librarian usercode, certain fields are automatically set up and may not be changed. This is the main screen for establishing the subset of tapes that you wish to capture with this particular rule.

The TRIM utility expects a series of “patterns” to be provided which indicates the name of the tape (or tapes) you wish to capture. These patterns have extremely powerful **wildcard** capabilities allowing a wide selection of tapes to have the same rule applied, regardless of the full volume name. For example, many daily or weekly backup jobs construct a tape name with a built-in Julian or military date, which will change from one run to the next.

It is not possible to create two TRIM rules to match an individual tape; TRIM automatically checks all rules patterns against all other existing rules.

---

Pattern matching is possible with the following tape fields:

## USERCODE

The USERCODE field refers to the usercode of the task that created the tape volume; this information is retrieved by SUPERVISOR or the LOGREADER task at the time the of tape creation. USERCODE is optional but cannot exceed 17 characters, including wild cards. If a usercode pattern is missing, then the TRIM rule matching will only match tapes created by jobs running without any usercode.

## VOLUME IDENTIFIER (VOLID)

The VOLID field is the primary pattern used by TRIM and its presence is mandatory in all rules definitions (except the implicit EXCEPTION rule). VOLID cannot exceed 17 characters, including wild cards.

## FILEID

For tapes with second-level file names (e.g. LIBRARY/MAINTENANCE tapes always have FILEnnn), the 'File ID' field should always be filled in since this is also used in the name pattern matching. For LIBRARY/MAINTENANCE tapes, it is important the FILEID pattern should be set to 'FILE###=' or 'FILE=' to allow tape volumes that extend to multiple reels.

There are, of course, exceptions to this; DMUTILITY-created database dumps do not have a second-level name nor do most program-created data tapes. In these cases, both the FILEID pattern and example fields should be left empty.

## HOSTNAME

The HOSTNAME field refers to the originating BNA or TCPIP host on which the tape volume was created. In a master-slave environment, HOSTNAME permits different tape expiration criteria to be applied, by system, for tapes created with identical usercode, volume id and file id. HOSTNAME is optional but cannot exceed 17 characters.

## Wildcard characters

Wildcard characters may only be used in pattern fields (i.e. Usercode, Volume Id and File Id). The combination of these patterns uniquely identifies the rule to which a set of tape volumes can belong. Each pattern field can contain a mixture of normal alphanumeric characters or wildcard characters. The available wildcard characters are as follows:

=	Matches any string of characters.
~	Matches any string of characters except '/'
#	Matches a single numeric character.
@	Matches a single alpha character.
&	Matches a single alpha-numeric character.
?	Matches any single character.

Note the '#' symbol often uses the alternative '£' character on many keyboards. You should NOT use wildcard characters in the example fields.

For example,

**=DUMP###**

would match 'MONDAYDUMP001' or 'SATURDAYDUMP999' but not 'MONDAYDUMPABC'. Also,

**INVDB#####@**

would match 'INVDB93123A' or 'INVDB94001Z' but not 'INVDB010194'.

The TRIM utility ensures that the patterns you provide are really intended to detect the tapes you want; that is why for each specified User/Owner (when available), Volume ID and File ID patterns you must provide valid examples. The TRIM utility will not allow the creation or update of any rules unless the example matches the provided patterns.

Note that wildcards are not obligatory, in fact the more information you can provide to uniquely identify a rule the less likely you are to get collisions.

Further, it is impossible to create two TRIM rules, which can satisfy the attributes of a single tape name and owner. As well verifying the example and pattern fields, the TRIM utility applies the new rule to the example fields of all other existing rules in the database. If a matching rule is found, the rule create will be aborted and an error message is returned.

## Retention information

The rest of the fields on this screen are used for establishing retention criteria. Usually, tapes are preserved up to a pre-assigned expiration date or for a specified number of generations. After that time, the tape will be returned to on-site, if had ever been moved off-site, and directly into the scratch pool for re-use. The manual tracking and control of tape movement within a large tape library, especially when handling retention by generation, can be a very time-consuming process. TRIM facilitates this process by allowing the user to maintain all this information automatically.

Tape retention is controlled in units of **days**, **years** or **Generations**. The **days/years** figure is used to assign an EXPIRY date to the tape at creation time. Since SUPERVISOR receives a tape label notice after the TRIM rule has been applied, this expiry information can be displayed on printed tape labels, if required.

## Retention by DAYS or YEARS

The easiest method of applying expiration is by time; TRIM will accept the keywords DAYS or YEARS with a non-zero retention time. These values may be used in for the normal expiration (i.e. setting of PGOK and movement to SCRATCH POOL) and location movement durations.

If an expiry value has 999999 DAYS, then this is equivalent to NEVER, setting the PGNEVER attribute .

## Retention by GENERATION

A GENERATION is defined as a member of a “family” of tapes, which are controlled by the same rule, but was created by a different job run at a different date or time. This definition means that any duplicates that were created from the run of the SAME job will NOT be regarded as a separate generation.

Whenever a new tape is created with a rule using generation expiration, that tape will become generation 1 in the family, the previous generation 1 will become 2 etc. Those generations that now exceed the limit imposed by the GENERATION field will become expired.

## Retention by SAVE and NEVER

It is now possible to assign the keys words SAVE and NEVER in the Retention Units field of a rule definition. SAVE means, by default, that the SAVEFACTOR of the tape will be used to calculate expiry date. No retention value will be allowed but normal generation and location controls can be set.

The NEVER modifier allows a tape matching this rule to be assigned PGNEVER status; again no retention value is permitted.

## LOCATION movement

The user may also specify a LOCATION movement list, allowing control over the movement of tapes to one or more off-site locations. Currently, there is no control over the location names chosen by the user, but it is anticipated that future releases will have much tighter controls. So, you should choose meaningful location names since they will be visible in **TP** command responses and appear in the daily reports, which inform Tape Library personnel of tape movement itineraries.

You may specify up to 8 separate locations and the duration that each tape remains in each location except the last. Where the last location duration field is left empty, the TRIM utility will automatically make the best judgement, taking into consideration the overall retention criteria. TRIM daily reporting mechanisms, which we have not yet discussed, will automatically ensure that tapes scheduled to move to new locations on or before today's date will be reported.

Let us consider an example. Complete the screen as follows:

METALOGIC S.a R.L					
Metatapelib		Add	Rules		
(FLEX) TEST					
User/Owner	Pattern	FLEX	Volume ID	File ID	
User/Owner	Pattern	FLEX	and Example	FLEX	
Volume Id	Pattern	<TEST###	>	and Example	<TEST123 >
File Id	Pattern	<FILE=	>	and Example	<FILE000 >
Hostname	Pattern	<	>	and Example	< >
<u>RETENTION DETAILS</u>					
Expires after :		<10 >	Days, Years: <DAYS > Or Generations : < >		
			Save, Never		
Location	<TAPE RACK >		Duration in	<1 >	Duration units <DAYS>
itinerary :	<VAULT >		location:	<2 >	(days or gens)
	<OFF SITE >			< >	
	< >			< >	
	< >			< >	
	< >			< >	
	< >			< >	
<u>LINKAGE INFO</u>					
User	<		> Identity	<	> Index < >
Action :		<	>		
( Return Next Quit )					

Transmitting this should return 'DB UPDATED' in the status line.

This rule has a time expiration of 10 days and three locations TAPE RACK, VAULT and OFF SITE. The first two locations have durations of 1 and 2 days respectively but the duration for OFF SITE may be left blank, if it is the last designated location. In this case, TRIM will effectively assign a default duration of 7 days, making the total location times the same as the expiry period.

After 10 days, tapes currently residing in the OFF SITE location will return back into the SCRATCH POOL.

Enter 'RETURN' in the action line and transmit. This will return you to the Home Menu screen.

Please change the text 'ADD' to 'INQ' and re-transmit.

## Interrogating rules

The Inquiry screen should look something like that shown below. Obviously, no update capability is available from this screen.

METATAPELIB Rules Inquiry					
(FLEX) TEST					
RULE INFO					
User/Owner	Pattern	FLEX		and Example	FLEX
Volume Id	Pattern	TEST=		and Example	TESX32APE
File Id	Pattern	FILE=		and Example	FILE000
Hostname	Pattern			and Example	
RETENTION DETAILS					
Expires after : 10 DAYS					
Location :	TAPE RACK	Duration in :	1	Days	
itinerary	VAULT	location	2	Days	
	OFF SITE				
Action : > < ( Return Next Quit )					

The third line of the screen shows the **Rule Identity**. In this case, for tapes, which satisfy the rule provided by the identity '(FLEX) TEST', the following OPAL TAPEDB attributes would be mapped:

```
RULEOWNER   =    FLEX,
RULEID      =    TEST
RULEINDEX   =    0
```

As a slight detour, for more information about these OPAL attributes try the following SUPERVISOR HELP ATTRIBUTE command:

```
TT HELP ATTR RULE=
```



## Navigating TRIM utility

The ACTION line allows single character input to control user activity through the rules database. To navigate through the rules system, certain simple commands are needed.

When available, 'N' or 'NEXT' in this field will show the next eligible rule in the database (subject to usercode restrictions), 'R' or 'RETURN' will always return to the Home Menu screen and 'Q' or 'QUIT' will terminate the TRIM utility.

## Significance of RULEINDEX

Please note that the RULEINDEX attribute mentioned in our example has no significance here; it is only meaningful when, for reporting purposes, it is convenient to logically group rules under one rule identity rule and use the rule index to make those rules unique.

This mechanism makes it very easy to logically group tapes created from the same application but which may have differing retention requirements (daily, monthly, quarterly etc. ). Please refer to the **TP FIND FAMILY** command discussed in **Chapter 5: TRIM commands** for more information.

The pattern information shown is used to select which tapes match this rule (USER, VOLUMEID and FILEID) followed by an example of a tape title, which should match the pattern. In the above case:

**(FLEX) TEST123/FILE000**

Note that the usercode is a very important aspect of the full tape name. Recognising the importance of this, the TAPEDB attribute TITLE will return the tape name in exactly the format shown above, where appropriate (use **TT HELP ATTR TITLE** for on-line help).

In the example shown, all tapes belonging to the (FLEX)TEST rule (or 'Family') have a time-based retention period of 10 days. Tapes matching this rule will get an expiry date calculated from their date of creation using this figure.

Location movement is controlled here, not by time, but by generation. New tapes will be kept in the location 'TAPE RACK' for one generation, then the 'VAULT' for one generation and will remain in the 'OFF SITE' location until they expire.

When expired, tapes will implicitly be moved to a 'SCRATCH POOL' location. As you can see from the example, it is quite acceptable to mix and match **generations** and **days** for controlling expiry and/or locations, if that is desired.

There are no restrictions on how you use these facilities though it should be noted that where both types of retention are used, tapes would not expire until **both** retentions have been satisfied.

Enter 'RETURN' in the action line and transmit. This will return you to the Home Menu screen.

Please change the text 'INQ' to 'MOD' and re-transmit.

## Updating rules

This screen will return you to the rules ADD screen you saw earlier. You are allowed to change any of the retention and pattern details subject to the constraints discussed earlier in building rules.

Change the retention period for this rule from **10 DAYS** to **20 DAYS**. Transmitting this page will return the usual 'DB UPDATED' message. The changed rule is immediately in operation and will be applied to any new tapes, which satisfy the name and owner patterns.

You can abort this 'transaction ' at any time by inputting RETURN and transmitting or by using the SPECIFY key, both of which will take you back to the Home Menu screen.

All changes to TRIM rules are logged and time-stamped in the TAPELOG trace file. The **TP LOG** command allows interrogation of this file or it may be very easily viewed from CANDE even whilst the database is on-line (see the response to the **TP VERSION** command to see the name of the TAPELOG file).

## Using SR and APPLY

Updating an existing rule, which already has its own family of tapes, can have significant effects. If expiry or retention information is changed, in particular, then a **TP APPLY <rule>** must be performed to update all members of the family or the **SR** command can be used to change an individual volume.

If owner, volume or file identifier fields are changed then it may be that members of the existing tape family will not match the rule characteristics any more. Changing such attributes for a rule, which already has a family of tapes, is NOT recommended.

## Rules in action

If the TRIM utility has now been terminated, we can see what effects the new rule will have on the information stored in the database. Under the usercode that you used to create our test rule, from a MARC or CANDE session:

**COPY \*OBJECT/TRIM TO TEST999**

You should now go back to your ODT or COMS window, where you were using **TT TP** commands before.

From that ODT or window, enter the following command:

```
TT TP FIND RULE (FLEX)TEST

----- FIND: RULE info for '(FLEX)TEST' -----

-- TAPE ATTRIBUTES -- PATTERN ----- EXAMPLE --
Usercode info        FLEX                FLEX
Volume id info       TEST###             TEST123
File id info         FILE=                FILE000

-- RETENTION & LOCATION INFO --
Tape EXPIRES after 20 DAYS
Location movement    'TAPE RACK    ' for 1 DAY
                    'VAULT        ' for 2 DAYS
                    'OFF SITE     ' Until expired
```

Note that the retention time is now 20 days, assuming that you made the alteration in the TRIM utility MODIFY screen. The **TP FIND RULE** command allows individual rules to be viewed on-line, without the need to use the TRIM utility and is often useful when testing or verifying a new rule.

Since a time for the 'OFFSITE' location had not been set up, the **TP FIND RULE** command shows duration for that location of 'Until expired'.

Assuming that the TEST999 tape has been successfully created, you can use the **TP TODAY** command from the ODT or COMS window that, if you recall, returns all tapes created in the last 24 hours and will include our TEST999 tape.

The response shows the presence of a new tape, serial R70153, which is showing the correct usercode and name (we created our tape under our rule usercode, FLEX).

```

TT TP TODAY

----- FIND: DATE after 12:15:31 on 25/05/2002 ----- Location ----- Created ---
R70153: (FLEX) TEST999/FILE000                                12:14 25/05/2002
000001: (FLEX) TESTTAPE/FILE000                                11:38 25/05/2002
000008: (FLEX) DBSPACK95144A/FILE000                          11:35 25/05/2002
R70846: (FLEX) META4040108/FILE000                            11:35 25/05/2002
R70845: (FLEX) META4040108/FILE000                            11:26 25/05/2002
R70844: (FLEX) META4040108/FILE000                            11:13 25/05/2002
R70843: (FLEX) META4040108/FILE000                            OFFSITE 23:07 24/05/2002
R70842: (FLEX) META4040108/FILE000                            OFFSITE 22:55 24/05/2002
R70703: (FLEX) META4040108/FILE000                            OFFSITE 21:25 24/05/2002
BOB001: (TAPELIB) METATAPELIB                                1:1 TAPE RACK 17:32 24/05/2002
000093: (FLEX) DBSPACK95143A/FILE000                          TAPE RACK 17:16 24/05/2002
000061: (FLEX) DBSPACK95143AM/FILE000                         TAPE RACK 16:13 24/05/2002
000056: (FLEX) DBSPACK95143AM/FILE000                         TAPE RACK 15:36 24/05/2002
000038: (FLEX) DBSPACK95142A/FILE000                         TAPE RACK 12:40 24/05/2002

```

We can now get information about the TEST999 tape: as previously, you can use TP R70153 as below:

```

TT TP R70153

----- METATAPELIB entry for MT [R70153] -----
Volume : TEST999/FILE000
Cycle   : 1   Reel   : 1
Version : 0   Density : BPI6250
Owner    : FLEX
Location :
Savefactor : 30
Creation info: 12:14:16 on 25/05/2002
Host info : STIRLING1 #6343
Pending loc : TAPE RACK                      Pending date : 25/05/2002
Expires    : 14/06/2002
Volume last accessed : 12:14:16 on 25/05/95
Created by task 0635 : *LIBRARY/MAINTENANCE
Owned by job no 0600 : (FLEX) "CANDE WFL"
Usage count : 14 times                      Last cleaned : 06/04/1994

Marked as : VOLUMED
Rule info : (FLEX) TEST

```

Note that the pending location has been set to TAPE RACK and a pending date to move to this location has been set to 25/05/2002 (today's date), which is also the date of creation of the tape. We can see that a TRIM rule has been found because the "Rule info" field shows as (FLEX)TEST.

Remember that tapes that do not match any existing rules are given a default rule identifier of EXCEPTION. We will be discussing these special rules later in this chapter. Lastly, the tape has already been assigned an expiry date of 14/06/2002 which is in 20 days time.

## Generations and locations

The TRIM utility is very flexible in handling conflicts between retention time/generations and location times/generations and will usually allow the last location field to be empty in both cases. If the total of all location times is not equal to the total retention time in days, or the total of all location generations is not equal to the retention generations, it may be unclear as to what action should be taken by the system when a tape is in its last location.

### Using SR

The **SR** modifier allow rules to be quickly applied to individual tapes without affecting all other generations of the set, if the matched rule has generation-based location movement or retention.

In fact, **SR** will never apply any generation criteria in such cases and will mark the **NOTES** field of the volume with an appropriate message indicating that generation based rules can only be enforced by the use of a **TP APPLY <rule>** command. The **TP APPLY** command will check and enforce both time-based and generation retentions for all members belonging to that rule family.

### Generation-base rule example:

Consider the following example rules:

Family	Retention	Locations	Duration
FAM1	5 days	LOC1	2 days
		LOC2	2 days
FAM2	5 days	LOC1	2 days
FAM3	4 gens	LOC1	1 gen
		LOC2	5 gen
FAM4	4 gens	LOC1	2 gen
		LOC2	
FAM5	6 days	LOC1	2 gen
		LOC2	3 gen
FAM6	6 days	LOC1	2 gen
		LOC2	
FAM7	3 days and 4 gens	LOC1	1 day
		LOC2	

In the previous table, the retention time should be considered as the **MAXIMUM** value of the total time in each location and the specified retention. Similarly, assigned generation retention is considered to be the **MAXIMUM** of the total generations specified for locations and retention generations.

To illustrate this feature, the behaviour of tapes in the will be as follows:

FAMILY	RETENTION ITINERARY
FAM1	LOC1 for 2 days LOC2 for 3 days SCRATCH POOL
FAM2	LOC1 for 2 days LOC2 for 3 days SCRATCH POOL
FAM3	LOC1 for 1 GEN LOC2 for 5 GENS SCRATCH POOL
FAM4	LOC1 for 2 GENS LOC2 for 2 GENS SCRATCH POOL
FAM5	LOC1 for 2 GENS LOC2 for 3 GENS SCRATCH POOL if older than 6 days or PENDING SCRATCH POOL until 6 days old
FAM6	LOC1 for 2 gens LOC2 until 6 days old
FAM7	LOC1 for 1 day LOC2 until more than 4th gen and more than 3 days old

This feature permits some security in the construction of rules.

TRIM will always ensure that a tape cannot expire until it has exceeded the allocated expiry control and the total time that would be spent in all designated locations.

## Authorisation phase

TRIM rules applied to newly created tapes only ever mark location movements as pending. The TRIM system requires actions from the Tape Librarian to confirm or authorise these pending actions and convert pending locations to current locations. Note that this process also includes expired tapes being finally moved to the scratch pool where they get a location of SCRATCH POOL and be marked as PGOK (purge authorised).

The first action of the authorisation phase is a reporting process which generates printed output detailing which tapes are to be moved, sorted by current location. This also allows the librarian to make any last-minute adjustments for specific tapes, which may require special action, or to check that new or revised rules are working correctly. To authorise these actions, it is necessary to use some of the standard Opals, which were compiled when the OPALS/TPDB file, discussed earlier in Chapter 3, was ENTERed.

Usually, this reports phase should be performed on a daily basis, preferably at a time when the production of new tapes is at a minimum. Enter the following command:

```
TT DO TPDB_DAILYREPORTS
```

This is a small subset of specific OPAL programs which scan the METATAPELIB database searching for tape entries which have a valid pending date action (the OPAL attribute PENDINGDATE) **earlier than or equal** to today's date. A valid PENDINGDATE must always be non-zero. For each tape found matching this selection, a special flag and attribute, called REPORTED, is updated for that tape entry.

After the command has finished, a printed report will have been generated from SUPERVISOR. If you have used '**TT USE DESTINATION**' to set up a valid Print System DESTINATION, then the print-out will either have been already printed or will appear in the response to a '**PS SHOW USER SUPERVISOR**' ODT command.

Interrogating the serial number R70153 after the TPDB\_DAILYREPORTS phase has completed, the output is identical except that the tape status entry line now shows:

```
Marked as      :    VOLUMED ,  REPORTED
```



Any other tapes that have been created since TRIM started tracking tape information will also have been marked as **REPORTED**. This is because even though no rules existed, the TRIM system would have been assigned a default rule name, called **EXCEPTION**, to all new tapes. In addition, these tapes are assigned a pending location of **EXCEPTION** and a pending date of their own creation date. The printed daily reports for these tapes will show all tape movements to the **EXCEPTION** location. The '**VOLUMED**' marker refers to tapes that have been WFL VOLUME ADDED on systems, which have the MCP option CATALOGING set.

Note that the reports phase can be repeated many times without affecting any previous reports. Any tapes, which are currently marked as REPORTED, will, however, remain in that state unless a **TP DEFER** command is used to cancel the assignment.

Assuming that all the movement changes have been authorised, **TP CONFIRM** is used to make the pending movements permanent:

**TP CONFIRM**

Volumes confirmed 8 (warnings 0),Links 0,Errors 0

**TP CONFIRM** accesses all REPORTED tapes, resets the REPORTED flag and transfers the pending location to the real location field. Further, if the tape retention is purely time-based (i.e. not generation) then the next pending location and pending date will also be automatically set by **CONFIRM**, as shown below, after interrogating R70153 again:

```

TT TP R70153

----- METATAPELIB entry for MT [R70153] -----
Volume : TEST999/FILE000
  Cycle   : 1   Reel   : 1
  Version : 0   Density : BPI6250
  Owner   : FLEX
  Location : TAPE RACK                Sent date   : 12:17:49 on 25/05/2002
  Savefactor : 30
  Creation info: 12:14:16 on 25/05/2002
  Host info  : STIRLING1 #6343
  Pending loc : VAULT                  Pending date : 26/05/2002
  Expires    : 14/06/2002
  Volume last accessed : 12:14:16 on 25/05/95
  Created by task 0635 : *LIBRARY/MAINTENANCE
  Owned by job no 0600 : (FLEX)"CANDE WFL"
  Usage count : 14 times                Last cleaned : 06/04/1994

Marked as : VOLUMED
Rule info : (FLEX)TEST

```

Note the pending location of VAULT and a pending date set to TOMORROW for that movement since the duration in the TAPE RACK location is only 1 day.

The VOLUMED marker is only applicable on CATALOGING systems and indicates that the tape has been entered into the system volume library where the MCP will maintain additional information about the volume.

Generation-based expiry operates in a different way: although the **CONFIRM** process has the same effects, the next pending location and date are **not** set up until a new generation of the tape family has been created.

### Generation family movement

Let's consider a simple example of a generation-based rule:

Rule id:	METATEST
Usercode pattern:	FLEX
Volume ID pattern:	NEWTEST£££
File ID pattern:	FILE=
Expiry :	2 GENERATIONS
Locations :	TAPE RACK for 1 GEN VAULT for 1 GEN

Note that generations 0 will always be in TAPE RACK and generation 1 will remain in VAULT. Generation 2 will always move to the SCRATCH POOL

The following tables show what happens to each member in the METATEST family shortly before the **CONFIRM** (this means that the daily reports phase will have been complete) and immediately after.

When the first volume, NEWTEST001, is created, it matches the METATEST rule:

Gen	Volume name	Pending Location	Current Location
0	NEWTEST001	TAPE RACK	
CONFIRM			
0	NEWTEST001		TAPE RACK

TRIM confers a pending location of TAPE RACK which becomes the tapes real location after the CONFIRM

Production of NEWTEST002 creation causes a location move of NEWTEST001 to the VAULT because of the 1 GEN limit on TAPE RACK.

Gen	Volume Name	Pending Location	Current location
0	NEWTEST002	TAPE RACK	
1	NEWTEST001	VAULT	TAPE RACK
CONFIRM			
0	NEWTEST002		TAPE RACK
1	NEWTEST001		VAULT

Lastly, creation of NEWTEST003 causes NEWTEST001 to move out to the SCRATCH POOL location and is assigned PGOK, once the CONFIRM has taken place.

Gen	Volume Name	Pending Location	Current Location
0	NEWTEST003	TAPE RACK	
1	NEWTEST002	VAULT	TAPE RACK
2	NEWTEST001	SCRATCH POOL	VAULT
CONFIRM			
0	NEWTEST003		TAPE RACK
1	NEWTEST002		VAULT
2	NEWTEST001		SCRATCH POOL

The PENDING LOCATIONS, where shown, will be automatically assigned by the TRIM system when new tapes are created and the PENDING DATES will be set to the current date.

It is strongly recommended that the daily reports and **CONFIRM** phases are done every day as close together as possible. The closer the two activities are scheduled together, the less likelihood of their being exceptions, though the **CONFIRM** process is very good at handling unexpected changes. Any discrepancies found during **CONFIRM** will be adjusted and reported in the TAPELOG as warnings or errors.

**CONFIRM** always returns a one-line response indicating how many tapes were confirmed, how many tapes were in error etc. This includes the automatic correction of any manual assigned location or expiry assignments performed on tapes, which had pending actions.

You can use the command **TP LOG CON** to provide information and errors about the responses returned from various **TP CONFIRM** commands.

To summarise the authorisation phase, two commands are needed:

```
TT DO TPDB_DAILYREPORTS  
TT TP CONFIRM
```

Getting optimal benefit from TRIM hinges on the sensible and regular usage of these commands. To maintain the Tape Library properly, expired tapes should be returned to the scratch pool for re-use as soon after expiry as possible. Only the **TPDB\_DAILYREPORTS** and **TP CONFIRM** command can provide that capability.

## Linking rules

One common requirement in many installations is the ability to allow the “linking” of one family of tapes to another such that the behaviour of the linked family in many ways match the family to which it is linked. For example, when a daily database dump is due to expire it might be desirable that any COPYAUDIT dumps, relative to that database dump, expire at the same time since these AUDIT dumps are useless without the original database dump. The numbers of audit tapes that would follow a ‘master’ in this case will usually be variable.

This feature can be achieved by a process called ‘**linking**’ where one or more TRIM rules are attached to a ‘master’ rule. In the above example, a rule assigned to the AUDIT dumps could be linked to the rule associated with the original database dump. For example, you might want to set up a TRIM rule for the METATAPELIB database as follows:

```
Rule id:                METADBDUMP
Usercode pattern:      TAPELIB
Volume ID pattern:     METATAPELIB
File ID pattern:
Expiry :               3 GENERATIONS and 20 days
Locations :            OFFSITE1 for 1 GEN
                       OFFSITE2 for 1 GEN
```

In the above rule METADBDUMP, METATAPELIB DMUTILITY dumps will be kept for a minimum of 3 generations or 20 days, whichever occurs later. Note that this rule has been entered by the Tape Librarian usercode and so it does not have an owning usercode and that the File Id pattern must be empty since database dump tapes never have a File Id. Now, building a rule for the corresponding audits using the TRIM utility might look something like the following:

```
Rule id:                METAAUDITS
Usercode pattern:      TAPELIB
Volume ID pattern:     METATAPELIB
File ID pattern:      AUDIT=
Expiry :
Link Rule ID:          METADBDUMP
```

You may have noticed on the TRIM Utility ADD/MODIFY screens several fields at the bottom, which indicates Linking information. No expiry information has been provided; this means that when the master expires, the slave AUDIT tapes will follow unconditionally.

If the METADBDUMP rule is protected with an owner e.g. META then this must be inserted into the usercode field in this part of the form:

**Link Usercode:**       **META**  
**Linked Rule ID:**     **METADBDUMP**

You may if you wish, assign your own location controls for the AUDITs and even your own expiry controls. The creation of the first METATAPELIB dump will establish a connection path for any AUDIT tapes that are subsequently created. The AUDITs will **always link to the most recent generation** of the METADBDUMP rule. You can use the **TP FIND FAMILY METATAPELIB** command to see which volume is currently generation 0.

If no location rules have been specified for the linked rule, as in the above example, the AUDIT tapes will automatically assume the location rules of the master when recorded in the database. That is new AUDITs will assume a pending location of 'OFFSITE1' with a pending date of today. These scheduled movements will be confirmed after the TPDB\_DAILYREPORTS and **CONFIRM** phases.

When the next generation of the METATAPELIB dump is taken, the old generation 0 of METADUMPDB will be immediately marked as pending location 'OFFSITE2' (since generation 0 only resides for 1 generation in 'OFFSITE1'). This effect will eventually ripple through all valid members of the METADBDUMP family during the imminent **CONFIRM** stage.

Information about tape volumes that have become linked to a "master" tape in another family can be extracted in several ways. First a simple interrogation of the linked tape will show the volume to which to which it is attached:

```
TT TP 90

----- METATAPELIB: MT [000090] -----
Volume : METATAPELIB/AUDIT12
Cycle      : 1                      Reel       : 1
Version    : 0                      Density    : BPI1250
Owner      : TAPELIB
Location   : OFFSITE1               Sent date  : 24/05/2002 at 23:30:01
Creation info: 17:16:03 on 24/05/2002
Savefactor : 30
Host info  : STIRLING1 #6343
Expiry date is UNKNWON
Volume last accessed : 23:16:03 on 23/05/2002
Created by task 6446 : *SYSTEM/COPYAUDIT
Owned by job no 6445 : JOB/COPYAUDIT
Usage count : 6 times since first entered

Rule info : METAUDITS      (linked to 000091)
```

Secondly, you can always use a **TP FIND FAMILY** on the linked rule name. For example, using the METAAUDITS rule used above, the following would be a typical response:

```
TT TP FIND FAMILY METAAUDITS

---- FIND:  FAMILY 'DBDUMPLINK' ----- Location ----- Created -- Gen
000090: (TAPELIB)METATAPELIB/AUDIT12      OFFSITE1      24/05/2002    0
      --- ABOVE VOLUMES LINK-ED TO:  '000091' ---
000071: (TAPELIB)METATAPELIB/AUDIT11      OFFSITE2      24/05/2002    1
000062: (TAPELIB)METATAPELIB/AUDIT10      OFFSITE2      24/05/2002    2
      --- ABOVE VOLUMES LINK-ED TO:  '000066' ---
```

Note that, as the rule dictates, generation 0 is currently in the same location as generation 0 of METADBDUMP, which is OFFSITE1. Tape serial numbers 71 and 62 are both linked to the same generation 1 of METADBDUMP and therefore reside in the same OFFSITE2 location.

## Unlabelled tape rules

As mentioned earlier in this chapter, it is possible to establish TRIM rules for unlabelled tapes. Because there are no volume or field identifier patterns that can be matched against an unlabelled tape, TRIM uses the names of the creating task and its parent job to identify the tape. Note that only 28 characters are stored in the database for both these fields.

```
TP FIND RULE (META)UL_TEST

----- FIND:  RULE info for '(META)UL_TEST' -----

-- TAPE ATTRIBUTES -- PATTERN ----- EXAMPLE --
Usercode info      META              META
Job id info        =DUMP/JOB=         DUMP/JOB
Task id info       *SYSTEM/DUMPALL=   *SYSTEM/DUMPALL=

-- RETENTION & LOCATION INFO --
Tape EXPIRES after 2 GENERATIONS
Location movement  'LOCATION 1 ' for 1 GEN
                  'LOCATION 2 ' for 1 GEN
```

Unlabelled tape rules have no restrictions in the type of retention used; generation, time or a mixture of both can control these families.

```

TT TP 89

----- METATAPELIB: MT [000089] -----
UNLABELLED tape
Cycle       : 1                      Reel       : 1
Version     : 0                      Density    : BPI1250
Owner       : META
Location    : LOCATION 1             Sent date  : 24/05/2002 at 23:30:01
Creation info: 17:16:03 on 24/05/2002
Savefactor  : 30
Host info   : STIRLING1 #6343
Expiry date is UNKNOWN
Volume last accessed : 23:16:03 on 23/05/2002
Created by task 6406 : *SYSTEM/DUMPALL
Owned by job no 6405 : JOB/UNLABELLED
Usage count  : 6 times since first entered

Rule info   : (META)UL_TEST

```

UL\_ rules will only be applied against unlabelled tapes and, in the example above, this rule will only match against such tapes which are created by tasks which have the name pattern `"*SYSTEM/DUMPALL="` and jobs which have a pattern of `"=DUMP/JOB="`. The **SR** and **APPLY** commands may be used, as normal, with all unlabelled rules.

## Permanent tapes

A "permanent" tape can be regarded as a tape that will never expire i.e. it cannot become PGOK and move to the SCRATCH POOL location by a CONFIRM action. These permanent tapes can be marked in one of two ways; first, the **PGNEVER** modifier of the **TP <serial>** command (see **Chapter 5: TRIM commands** for more details) which can be used to mark individual tapes to never expire. The **CONFIRM** command will never mark **PGNEVER** tape as **PGOK** regardless of expiration.

This feature can also be controlled through TRIM rules. When setting up a rule with location controls, if the last location duration is set to the value of 999999 days then TRIM will never expire any members of the rule family. This mechanism really only applies to time-based expiration.

If the rule is time-based then TRIM will NOT set up an expiry date. Normal location movements will take place until the tape reaches its last location where it will stay forever. Such tapes can only be manually expired and scratched using the **PGOK** modifier of the **TP <serial>** command.



Consider the following time-based rule:

```
Rule id:           METADBDUMP
Usercode pattern: TAPELIB
Volume ID pattern: METADUMP
File ID pattern:   FILE000
Expiry :
Locations :        OFFSITE1 for 10 days
                   VAULT for 999999 days
```

Tapes in the METADBDUMP family will move into OFFSITE1 for 10 days after creation; thereafter, they will stay permanently in VAULT until manually overridden.

```
TT TP 91

----- METATAPELIB: MT [000090] -----
Volume : METADUMP/FILE000
Cycle      : 1                      Reel      : 1
Version    : 0                      Density    : BPI1250
Owner      : TAPELIB
Location   : VAULT                  Sent date  : 10/01/1996 at 23:30:01
Creation info: 13:44:29 on 30/12/2002
Savefactor : 30
Host info  : STIRLING1 #6343
Expiry date is UNKNOWN
Volume last accessed : 23:16:03 on 23/05/2002
Created by task 9112 : *LIBRARY/MAINTENANCE
Owned by job no 9110: COPY/METATDUMP
Usage count : 12 times since first entered

Rule info : METADBDUMP
```

## Linking delays

When a new generation of METADBDUMP gets created, the AUDIT dumps currently linked to an old generation 0 (now generation 1) will remain unchanged until the daily reports and **CONFIRM** phase have authorised the scheduled movements of the master METADBDUMP tapes.

It is during the **CONFIRM** phase that the TRIM system will establish pending locations/dates for all tapes, currently linked to a master, such that the pending location matches the current master location. The next reports and **CONFIRM** phase, usually the next day, will authorise the movements of the linked tapes.

Typically, if the reports and **CONFIRM** phases are being done once a day then linked tapes will always be one day behind any new location assignments given to their masters.

Similarly, when a master tape expires, moves to the SCRATCH POOL and is marked PGOK, any linked tapes associated with that master will not be destined for the same fate until the following **CONFIRM** phase has been completed.

The response generated to the **TP CONFIRM** command indicates how many linked tapes have been updated during the authorisation phase.

The linked rules can have their own location and/or expiry controls. In such cases, the master's location itinerary will be ignored and the linked tapes will follow their own movement controls. However, the linked tapes will still try to follow the master to the scratch pool, once that master has become PGOK, unless the linked rule has its own expiry criteria.

Providing a linked rule with its own expiry control is limited in its use, however, since the whole point of linking is to force one or more families of tapes to be released at the **same time** as the master.

Because of the enforced time delay involved in releasing linked tapes after the master has already expired, some sites may wish to perform multiple daily reports and **CONFIRM** phases in succession. As a precaution, it is advisable that each generated report is checked as usual to make sure that the scheduled movements are correct. Alternatively, the **TP FIND PENDING** command can be used to view which tapes still have pending movements after a **CONFIRM** has completed.

## EXCEPTION rules

By default, if there are no TRIM rules present in the METATAPELIB database, the TRIM system will allocate a default rule of EXCEPTION for all new tapes created on the system. In this case, this implicit EXCEPTION rule sets up a default location of EXCEPTION and an expiry date constructed from the SAVEFACTOR attribute assigned to the tape.

TRIM provides the capability for a site to define its own EXCEPTION rules, which may be specific for individual usercodes or system-wide. When a user defines a rule with an identifier of EXCEPTION, the TRIM utility will expect the pattern and example fields on the form to remain blank. It is not possible to set up an EXCEPTION rule in any other way because, by their very nature, such rules refer to all tapes, which do NOT match any existing rule. Normal retention and movement details can be set as would be the case for any rule.

In addition, EXCEPTION rules are permitted under individual usercodes. The usercode can be explicitly set using the ADD command from the primary rules handling screen by the Tape Librarian or is implied when the user is logged-on under a conventional usercode. In the latter case, an EXCEPTION rule can be set only for that usercode. As with the standard EXCEPTION rule, no patterns are permitted in usercode, volume and file id fields, these must always be blank.

When the TRIM system is searching the rules database at tape creation time, the following precedence is adopted:

First, search for any non-EXCEPTION rule, which matches the volume

Next, search for an EXCEPTION rule under the USERCODE of the tape owner.

Next, if a non-usercoded EXCEPTION rule exists, this rule will be automatically assigned.

Lastly, if none of the above applies, the TRIM system will set a default rule of EXCEPTION and pending location EXCEPTION.

It is recommended that, at least, a global EXCEPTION rule is created to ensure that some basic retention criteria be assigned for all tapes without rules.

## Testing and changing rules

New or modified rules are always immediately available for use in the TRIM system. New tapes as they are created will be checked against all the rules held in the METATAPELIB database until a match is found. If no match is found, the EXCEPTION rules are then validated. So after changing a rule, it is a good idea to validate that it really is doing what you expect as soon as possible.

TRIM has various aids, which can assist in the validation and testing of rules. It is convenient if you have a tape serial number in the database which is old and unwanted and whose data can be easily changed. Alternatively, you can create a new entry using an update-capable DMINQUIRY. The tape data of that serial number can be easily changed using the TP interface, to match the criteria of your rule. For example,

```
TP 123456 CDATE 12/10/94
TP 123456 VOLID TRIMTEST
```

Would change the creation date of serial 123456 to 10 December 1994 and its volume identifier to the string 'TRIMTEST'. Similarly, you can change all the criteria for a rule in this way (use the modifiers FILEID to change the file identifier, OWNER to change usercode, LOCATION to change the current location if necessary etc. ).

After you have changed the data, how do you assign a rule? You use the **SR** (Search/Set Rule) command for this:

```
TP 123546 SR
```

**SR** behaves in a similar fashion to the way that the TRIM system behaves when a new tape is created, though **SR** on a volume with generation-based retention rules is more limited. **SR** will search all the current rules matching the usercode, volume identifier and file identifier of tape serial number 123456 against each rule in the METATAPELIB database.

As soon as a matching rule is found, the search process stops and the tape library system will assign an expiry date (calculated from the rule), a pending location and pending date (which is always today's date) and a rule usercode, rule identity and rule index.

Note that, for generation-based rules, the **SR** command does not look at any other members of the family; the volume is always treated as generation 0 and the pending location is always set to the first location. Conversely, **SR** will automatically calculate the correct location for tapes whose rules are purely time controlled by using the tape creation date as a base.

Interrogating the tape using the command **TP 123456** will show all this new information available. If you are not happy with the rule, change it through the TRIM utility and then re-use the **SR** command. You can use **SR** as many times as you like since it will only ever work on that individual tape and no other members in the rule family.

If you want to test out the rule for more than one volume (strongly recommended if you have any generation-based expiry or location controls in the rule), then you should use **SR** to assign the rule and therefore add new members to the family. To see all members of a particular rule family, use the command:

**TP FIND FAMILY <ruleid>**

The <ruleid> specification should include the owning usercode of the rule, identity and any rule index, if applicable. Note that you can use wildcards in the <ruleid> if you wish: the system will return all families, which match the rule specification, logically splitting the output for each rule.

If you have used a rule which has generation-based retention or location movement, then you must use the **TP APPLY** command to assign the correct pending locations to each generation in the family. This process will access all members of the current family, including those that may be due for expiry. **However, note that TP APPLY will not inspect tapes, which have been marked as PGOK.**

For example:

**TP APPLY <ruleid>**

The TRIM system will return a message indicating how many tapes have been affected by the **APPLY** command:

#### **12 Volumes updated by APPLY command**

Using **SR** on a tape volume, which matches a "linked" rule assignment, causes TRIM to search for an eligible tape in the master family. TRIM tries to link the tape to a master that has the closest and oldest creation date. If no suitable master tape can be found, two possible actions can occur:

If there are existing members of the master family then the **SR**-ed tape will be marked pending location **SCRATCH POOL** with a pending date of today. The response to the **SR** command in the **TAPELOG** will be modified to indicate that a linkage was not possible.

Alternatively, if there are no members of the master family in the database, linking is again impossible but in this case the link identifier is set to **BADLNK**. The command:

**TP FIND LINKED <serial>**

Can be used to show the subset of tapes that are currently linked to the master '**<serial>**'. Not that tapes which have the bad linkages discussed earlier can be easily seen using **TP FIND LINKED BADLNK**.

The use of **SR** and **APPLY** to a great degree reflects what happens when a tape is created on the system. The assigned pending location and pending date will remain in force for each tape in the family until the "daily reports" and **CONFIRM** phases have been actioned.

As discussed in the **Authorisation Phase** section earlier in this chapter, this reporting phase uses the **TPDB\_DAILYREPORTS** Opals to mark each eligible tape, which has a pending movement, scheduled for today or earlier, as **REPORTED**. To simulate this reporting process, you can use the command:

**TP 123456 REPORTED**

The **CONFIRM** command, which primarily authorises location movements, only operates on tapes which have been marked as **REPORTED**. So, using the **TP <serial> REPORTED** command, it is possible to quickly test out rule changes by doing the **SR**, setting the **REPORTED** flag and invoking **TP CONFIRM**.

If you want to cancel the **REPORTED** flag:

**TP 123456 DEFER**

The **TP CONFIRM** process will select all the **REPORTED** volumes, convert the pending locations to current locations and will change pending dates. If the rule assigned to the tape has no generation controls (i.e. time-based only) and has location assignments, then **CONFIRM** will assign a new pending location and date for the next movement.

For generation-based expiry or movement control, pending actions are only set when the next generation 0 is created for that family.

## TRIM strategy

Most customers will experience one of two possible scenarios for sites using the TRIM Tape Library. Some existing sites may be using older OPAL rule-based systems, which manually applied changes to tape entries at creation time using **TP** commands from OPAL ODTSEQUENCES. Expiration and generation controls are implemented in a similar fashion to TRIM rules by checking owner, volume and file ids. Also, some sites may be migrating from a different A-Series Tape Library system or been running TRIM for some time without having established any rules.

In these cases, the Tape Library database will consist of a mixture of tapes, which have no assigned rules, and more recently created tapes, which have a rule identity of **EXCEPTION** with the likelihood of no expiry information. Where the default rule id of **EXCEPTION** has been used, a pending location of **EXCEPTION** will also have been assigned. You can use **TP FIND LOCATION EXCEPTION** to see these tapes.

If you are intending to convert from an alternative database system, please contact Metalogic for advice and assistance.

The procedures suggested here will cover both the above SITUATIONS. For OPAL-based sites, it is important that as new rules are implemented, the **TP\_RETENTION** Opals are changed to exclude the tapes being handled by the old mechanisms.

### Building a tape library database

Once installed, the TRIM software can be recording new tape and scratch information immediately. However, for those sites that have a large library of tapes and wish to start using TRIM rules as soon as possible, especially generation-based tapes, this presents a problem.

The key is to get as many tapes into the **METATAPELIB** database as soon as possible. One mechanism available to do this is to use the **LOGREADER** utility if you have access to system **SUMLOGs**. **LOGREADER** can process a single **SUMLOG** in batch mode and retrieve all relevant tape creations updating **METATAPELIB** in the process.

If more than one SUMLOG is available, these must be processed in **chronological order**.

To run LOGREADER in batch, it must be run under the Tape Librarian usercode (see **TT USE** command) from a WFL job, MARC or Cande

The name of the SUMLOG file must be run-time file-equated to a logical file, also called SUMLOG.

For example:

```
RUN *METALOGIC/SUPERVISOR/LOGREADER;  
FILE SUMLOG= *SUMLOG/1234/101595/000703 ON LOGPACK
```

Where possible, BOJ and EOJ records will be tracked to extract usercode, job and task names to allow as much information as possible to set up for each tape logged.

Once all available logs have been scanned, the **TP\_PERRECORD** OPALs discussed in **Chapter 6: Getting started with OPAL** are a way of capturing tapes, missed from the log-reading process, and subsequently being used as input on the system. Note that Supervisor will automatically detect new tape creations and scratched tapes.

However, the problem with using **TP\_PERRECORD** is that the original creator usercode will not be set up and this makes the use of TRIM rules difficult. It is not advisable to run the **TP\_PERRECORD** program for too long because of the owner problem but it is a simple way of ensuring that tapes information gets into the database.

If you are using **TP\_PERRECORD**, then it is probably a good idea to regularly check those tapes, which are marked with a usercode of UNOBTAINABLE. If possible, the OWNER fields should be changed to reflect the correct usercode of the creating task.

**It is good practise to stop running these Opals once you have started to use TRIM rules.**



## Review tape expiration

Review tape expiration strategy for the entire system. For time-based expiration, decide how long you want to keep tapes for and which locations you want them to be assigned to at the various stages of their lives. The same procedure should be adopted for generation-based tape families. Decide if you want to use unlabelled tape or tailored EXCEPTION rules (global and/or usercoded). EXCEPTION rules are useful such that all non-rule based tapes can have a default expiry and location, if desired.

## New rules and EXCEPTION tapes

If you are serious about setting up TRIM rules, it is a good idea to cancel any currently assigned pending movements for the aforementioned EXCEPTION tapes. If you have been using an OPAL-rules based system with the TRIM system, there could be many hundreds of tapes that have the correct locations and expiry provided by the OPAL rules but with non-zero pending dates.

Because an important part of making new rules effective is the ability to action the daily reports and TP CONFIRM phases as soon as possible, it is important to cancel all outstanding pending movements or the effects of the old OPAL rules will be altered.

To avoid this problem, **it is strongly recommended** that you take the following course of action. First make the following SUPERVISOR OPAL programs (or they may be found in the file OPALS/TPDB, found on the METALOGIC release tape):

```
TT DEFINE + SITU TPDB_RESET(TAPEDB) :
    RULEID EQL "EXCEPTION" AND PENDINGDAY GTR 0

TT DEFINE + ODTS TPDB_RESET(TAPEDB) :
    ODT("TT TP ", SERIALNO, " PEND NEVER") ;

TT EVAL TPDB_RESET DO TPDB_RESET
```

The EVAL will scan the database and reset the pending date and pending location of all EXCEPTION tapes.

## Making an example rule

Select the first TRIM rule you wish to establish; if you are using rules for the first time try to pick a set of tapes whose numbers are small e.g. weekly dumps. Consider the following family of tapes, which have an owner of 'META' and the following name pattern:

**DBSPACK£££££?/FILE000**

Where £££££ is a variable 5 character numeric string (in this case a Julian date) followed by any single character. The tapes are always created by the META usercode.

In our example, we want to keep the dumps for 3 generations with the following locations:

<b>Generation 0 will be in STIRLING</b>	<b>Duration: 1 gen</b>
<b>Generation 1 will be in PERTH</b>	<b>Duration: 1 gen</b>
<b>Generation 2 will be in GLASGOW</b>	<b>Duration: 1 gen</b>

Generations older than 2 will be returned automatically to the SCRATCH POOL and be marked as PGOK (purge authorised).

To create the rule, we'll call ours DBSPACKDUMP, use the TRIM utility running under the Tape Librarian usercode. Because this tape is produced from an important application, we would not normally give this rule an owner usercode or a rule index so that only the Tape Librarian can change the rule.

Patterns of 'META', 'DBSPACK£££££?' and 'FILE=' are provided for Usercode, Volume Id and File Id fields respectively. The rule is set up with retention of 3 generations and 3 location movements, each generation in each location for 1 cycle only.

Any new 'DBSPACK' tapes would pick up the new rule if the patterns match correctly.

When building rules, it's important to try to make the patterns used as unique as possible. From a COMS window or ODT, the command 'TP FIND RULE DBSPACKDUMP' would confirm the details of the rules that have been set up.

## Building a rule family

To find what eligible DBSPACK dump tapes are in the database, use the command

```
TP FIND NAME (META) DBSPACK#####=
```

Which will find all tapes that match the above pattern. Make a note of all the tape serial numbers or get a printed listing (use the prefix **TT PRINT** to get a SUPERVISOR generated print-out).

Use the **SR** (Search/Set Rule) on each serial number in the list e.g.

```
TT TP 123456 SR
```

```
VOL:123456 rule applied - generation rules not done
```

Both commands and responses will appear in the TAPELOG. Note that **SR** does not affect other generations of a generation-based rule. After **SR**-ing all serial numbers, we can see the new family using:

```
TP FIND FAMILY DBSPACKDUMP
```

Individual **TP** commands will show them all with the same pending location of STIRLING. For generation-based rules, the **SR** command does NOT look at any other members of the rule family and this is signified by the **SR** response. The **NOTES** entry for each tape also suggests that you should do an **APPLY** command to consolidate all members of the rule family:

```
TP APPLY DBSPACKDUMP
```

If the new rule family has a large number of members, the TRIM system may take some time responding but you will get a response saying

```
'N volumes affected by APPLY command'
```

Where 'N' is the number of volumes actually updated during the **APPLY** process. This number need not match exactly the number of members in the rule family. Individual interrogations of each generation of the family should now show an appropriate pending location i.e. generation 1 should have a pending location of PERTH, generation 2 should be GLASGOW, generation 3 should be EDINBURGH and generation 4 or later should be SCRATCH POOL.

The pending dates will all be today's date.

## Reporting and CONFIRM phases

To make these pending changes permanent, it is necessary to mark all members of the family as REPORTED (i.e. '**TP 123456 REPORTED**'). This process can be done in two ways. Each volume can be manually marked using the above **TP** command from any Supervisor COMS window or ODT. Alternatively, you can **EVALUATE** the TPDB\_RESET Opals, discussed earlier, and followed by a **DO** of the TPDB\_DAILYREPORTS ODTSEQUENCE. The initial **EVALUATE** is necessary to **DEFER** all outstanding pending EXCEPTION movements and avoids confusion.

After the reporting phase, the movements will be authorised by the **TP CONFIRM** command. You will get a response indicating how many volumes were confirmed and how many errors; warnings and linking operations were encountered. In the DBSPACK example, note that generations 4 or older will then have moved to the SCRATCH POOL and been marked PGOK.

These tapes will now NOT appear in a **TP FIND FAMILY <ruleid>** command response because of their PGOK setting, even though the rule identity will still appear in the **TP <serial>** response. You should note that this exclusion of PGOK tapes from rules families is likely to change in future releases.

## EXCEPTION rules

It's probably a good idea to make an EXCEPTION rule at some point. This allows you, at the very least, to put default expiration and location information for new tapes as they are created. Do not use the TPDB\_RESET Opal programs after you have set up an EXCEPTION rule so it is sound practice to try and convert as many of your tapes as possible that require TRIM rules **before** setting up any EXCEPTION rules.

## Modifying rules

Modifying an existing rule is easy to do using the TRIM utility. However, if you do decide to change the retention or location criteria for a rule, you must remember to update the whole family using a **TP APPLY** command. Be careful of increasing the number of retention generations; if the rule has been in force some time, then there may be now valid generations, which have been moved to the SCRATCH POOL. You will have to manually cancel the **PGOK** assignment (using the **PGOK-** modifier) for each tape prior to executing the **TP APPLY** command.

## Using "linked" rules

If you use DMSII databases and regularly produce COPYAUDIT dumps, it is strongly recommended that you adopt the use of linked rules. In the case of database dumps, this allows the relevant audit tapes, which are only usable with the last full database dump, to be automatically released when the database dump has gone.

Note that you may have more than one linked rule to a 'master' rule.

## De-implementing old retention systems

If you are running OPAL-based rules, please remember to update your TP\_RETENTION ODTSEQUENCE to eliminate the tapes now covered by any new rule, which you bring into operation. Eventually, when all the old rule mechanisms have been replaced, the only reason for running a TAPELABEL OPAL is for filtering purposes.

## Caring for your TAPELOG

Use the **TP LOG** command to check the testing and to look for any problems, especially when issuing **CONFIRM** commands. Because the TAPELOG can grow very large over time, it is recommended that you close the TAPELOG at least once a week at a convenient time.

A suggestion using the following AFTER, for example late on Sunday nights:

**TT AF + 2330 ON SUNDAY: TT TP LC CLOSE**

Work is scheduled to enhance the accessibility of the **TP LOG** interface to allow full access by time or date and to interrogate off-line TAPELOGs.

## Regular CONFIRMs

When you have the rules established and working as planned, please remember to do the **TPDB\_DAILYREPORTS** and **CONFIRM** phases every day. Neglecting these activities over a long period of time can negate the effectiveness of TRIM. Metalogic recommend that the activity **TT DO TPDB\_DAILYREPORTS** is set up on a daily basis by the **SUPERVISOR AFTER** command at the same each day.

The **TP CONFIRM** command ideally should be actioned as soon as possible after the reports have been checked but it is not absolutely essential. At least in the early stages of using TRIM rules, you should check the generated printout for any errors or inconsistencies before the **CONFIRM**.

If there is a considerable gap between the reports and the **CONFIRM** phase, new tapes created in a generation based rule family can be affected because it is likely that their current pending movements are now incorrect.

In normal operation, the TRIM system adjusts this automatically but will not do it for tapes that are marked as **REPORTED**. The **CONFIRM** command will flag such tapes with a warning in the TAPELOG and will establish a new pending location and date should it be deemed necessary.

## Software changes and enhancements

The TRIM software is continually being enhanced and faults addressed. Metalogic strongly recommend that you read the supplied fixes and changes documentation which are regularly mailed to all customers every 3-4 months, as well as the **METANOTES/=** files automatically loaded by the **INSTALL** utility during installation of a new release.

## Chapter 5: TP commands

*What with excellent browsing and sluicing and cheery conversation and what-not, the afternoon passed quite happily.*

P. G. Wodehouse, My Man Jeeves

The primary interface used by the TRIM software to handle user commands is maintained through the use of the TT command either at an ODT, SUPERVISOR-COMS window or REMOTESPO. The TT commands available are a strict subset of those for the full SUPERVISOR system and they are similar in form to standard ODT commands. Similar to that implementation, input may be free-format and spaces are optional, unless needed to resolve ambiguities.

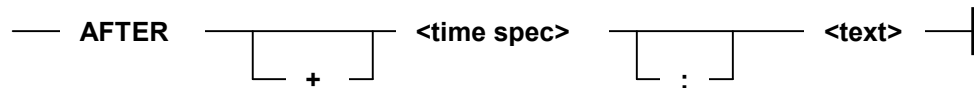
This chapter discusses the various commands, all available using the **TP** prefix, that may be used to interrogate and change Tape Library information. All the currently available commands are listed below with a short description of its functionality. The **HELP TP** command, as opposed to the less preferred **TP HELP** synonym, allows access to all Supervisor and TRIM on-line help.

At most, only the first 4 characters of a command need to be entered. Synonyms exist for many of the commands and all are accessible from an ODT or COMS window, subject to certain security considerations. The table below is logically divided into two sections: the first section covers inquiry and modification of a specific tape volume and a single serial number is always needed for these commands. The second section covers general reporting commands plus those that are responsible for authorising and controlling the effects of TRIM rules.

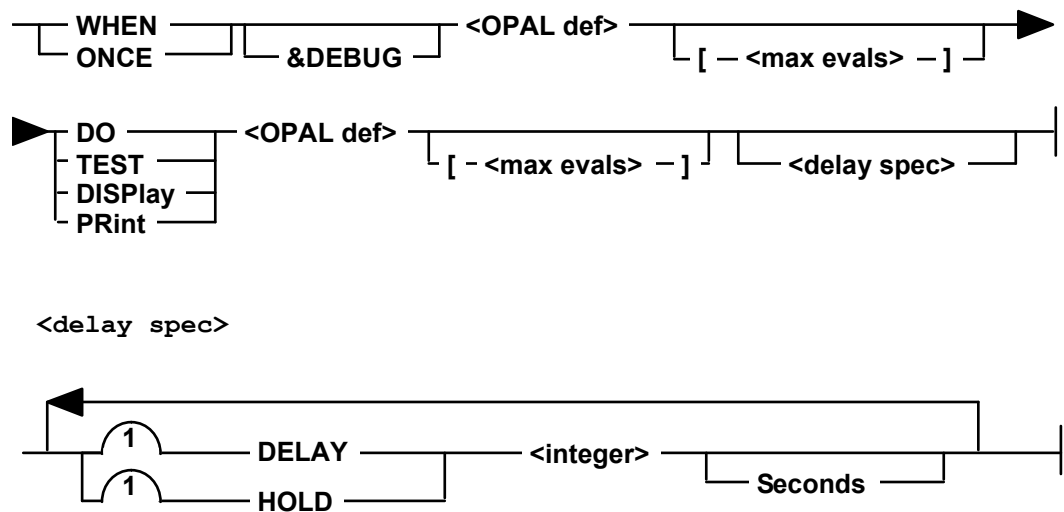
## Railroad Diagrams

Railroad diagrams are used throughout this manual to describe the syntax of the various TRIM and SUPERVISOR commands. Many of these can become quite complicated and confusing due to the number of options available within some commands. In order to simplify the diagrams to a degree, a convention has been adopted for commands, which have synonyms that are abbreviations of the full command keyword. For example, **AF** is a valid synonym for the SUPERVISOR **AFter** Command.

To show the above behaviour, the letters comprising the valid contracted synonym are shown in upper case and the rest of the keyword shown in lower case:



Where a command has an alternative that is not an abbreviation, then the alternative is shown as a branch on the diagram:



In the above diagram for the WHEN command, there is a possible keyword of DISPLAY. This can be given in full as DISPLAY or abbreviated to DISP. WHEN itself has an alternative form of ONCE.



## Tape modification commands

Each of the following commands require the following prefix:

**TT TP <serial>**

The <serial> in question must be a valid tape volume serial number currently present in the database. Note that a **TP <serial>** command **without** a modifier will return a full page of information detailing all current database information relating to the provided tape serial number.

The current list of modification commands:

BOX	<text>	Changes the box number of the tape
CDATE	<date>	Changes the creation date of the tape
CLEANED		Mark a tape as just cleaned
CTIME	<number>	Changes the creation time of the tape
DAMAGED	+ or -	Turn on the damaged flag
DEFER		Postpone pending movement action
DELETE		Delete a tape from the database (Librarian only)
DESTROYED		Mark a tape as destroyed
EXPIRY	<date>	Set the tape expiration date
FILEID	<text>	Changes the tape file identifier of the tape
HOSTNAME	<text>	Changes the origin hostname of the tape
JOBID	<text>	Changes the job name of tape creator
JOBNO	<number>	Changes the job number of tape creator
LOCATION	<text>	Changes the tape location immediately
LOST	+ or -	Turn on the lost flag
MIXID	<text>	Changes the task name of tape creator
MIXNO	<number>	Changes the task number of tape creator
NOTE	<text>	Changes the user-specific notes for a tape
OWNER	<text>	Changes the owner (usercode) of the tape
PEND	<date> NEVER	Changes the current pending movement date
PGNEVER	+ or -	Turn on 'keep forever' flag
PGOK	+ or -	Changes the purge-authorised flag
PLOC	<text>	Changes the pending location movement
POOL	<text>	Changes the SCRATCHPOOL attribute
REPORTED		Mark tape as detectable by CONFIRM
RETURN	<date>	Set the return date of a tape (from offsite)
SEND	<date>	Set the send date of a tape (to offsite)
SR		Search rules and match tape volume info
SYSNO	<number>	Changes the system serial number
UNASSIGN		Changes an allocated tape to unassigned
VOLID		Changes the volume identifier

## OPAL attributes and METATAPELIB DASDL fields

Many of the above update commands directly change values in the database entry for the specified tape. The current DASDL symbol file, used to compile the METATAPELIB database and generate tailored software, is listed in **Appendix C: METATAPELIB DASDL**. The discussion of these fields is beyond the scope of this manual as it is generally unnecessary for the site to require their own programmatic access to the METATAPELIB database.

Each database field has a corresponding OPAL attribute belonging to the TAPEDB and, to a lesser extent, TAPELABEL contexts. Usually, the same identifier is used for both the database field and the corresponding OPAL attribute if possible.

In this chapter, each tape modification command will detail the relevant database field and the corresponding TAPEDB attribute affected by its usage. The type of the attribute e.g. Boolean, String, Real, will also be shown.

For example, the description of the **PGOK** modifier will include:

<b>DASDL field:</b>	<b>PGOK</b>	
<b>TAPEDB attribute:</b>	<b>PGOK</b>	<b>(Boolean)</b>

The set of OPAL attributes belonging to the TAPEDB and TAPELABEL contexts are discussed in more details in the **Commonly used OPAL attributes used in Supervisor** reference manual, which accompanies this document.

## Rule reporting and authorisation

These commands help to provide fast access to a variety of database information to generate on-line or printed reports.

The current list of reporting commands:

APPLY	Re-applies a rule to all members of family.
CLOSE	Shuts down database for off-line dump.
CONFIRM	Confirms all pending actions on REPORT-ed tapes.
DUMP	Initiates dump of database to flat files
EXPIRED	Variant of TP FIND EXPIRED for non-scratch tapes
FIND DATE	Shows a list of tapes created within a date range.
FIND EXPIRED	Shows a list of expired tapes.
FIND FAMILY	Shows a list of tapes belonging to a specified family.
FIND LINKED	Shows all tapes currently "linked" to a master.
FIND LOCATION	Shows a list of tapes matching location criteria.
FIND NAME	List of tapes matching user name criteria.
FIND PENDING	Shows a list of tapes with a current pending action.
FIND RULES	Returns information about individual rules
FIND SCRATCH	Shows a list of tapes which currently have the PGOK attribute TRUE.
HELP	Provides on-line help information.
LC	Logs the supplied text into the current TAPELOG.
LOG	Display recent log information from the TAPELOG.
NET	Control networking protocols and master-slave configuration
OPEN	Makes the METATAPELIB database available after a TP CLOSE.
SCRATCH	Synonym for TP FIND SCRATCH
SER	Synonym for TP FIND NAME
SHOW	Identical to TP <serialno>
TBC	Shows a list of tapes that require cleaning.
TODAY	Shows a reverse creation time ordered tape list.
VERSION	Shows version and run-time information for TAPELIBUPDATER and OPALTAPELIB

## Modifying Tape Information

*Man's yesterday may ne'er be like his tomorrow  
Nought may endure but mutability.*

Percy Bysshe Shelley, Mutability (1816)

There are two ways to modify information manually in the METATAPELIB database - by using the various DMSII facilities, such as INQUIRY or ERGO, directly on the database or by using the **TP** command interface directly into the TAPELIBUPDATER process.

Most **TP** commands are accepted from unrestricted terminals, but TRIM will only allow the designated Tape Librarian to physically delete a tape from the database. All other modification commands are accepted from an ODT, COMS window or Supervisor ODTSEQUENCES, unless otherwise stated.

Each command is not executed by SUPERVISOR itself but passed to the TAPELIBUPDATER task for processing. TAPELIBUPDATER calls entrypoint procedures in the OPALTAPELIB library to physically update tape information in the METATAPELIB database. Like top-level Supervisor commands, this subset of **TP** commands has their own on-line help facilities.

All changes made via any **TP** command are logged into TRIM's TAPELOG file. The originating command and any associated result message will be logged.

For example:

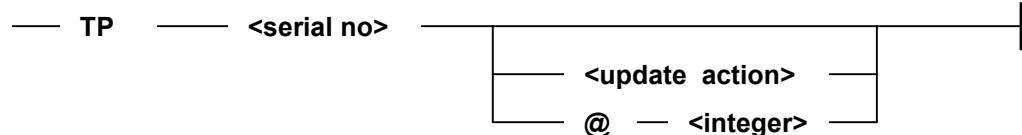
```
TP LOG ODT
----- 24/07/95 -----
13:16:34 ODT: 123456
12:20:28 ODT: FIND FAMILY METAT=
12:20:20 ODT: 123456 PGOK
11:00:07 ODT: META02 LOCATION BAHAMAS
10:30:22 ODT: META02 CDATE 15/12/1994
```

The **TP LOG ODT** or **TP LOG MSG** commands can return information about all TRIM commands issued within a window of the past 7 days or 3000 lines of screen output.

## TP serial number commands

*'Whom are you?' he asked, for he had attended business college.*

George Ade, 'The Steel Box', 1898



Each tape recorded into the METATAPELIB4 database must have a unique 6-character serial number. To get access to tape information for a specific tape volume, the TRIM system will accept the serial number prefixed by the text '**TP**'. All operator interrogation and modification commands require the presence of this field before any command will be recognised.

Using **TP** with a serial number alone, a page of database information is about the tape with the specified serial, is returned to the caller.

**TP SHOW <serial no>** is equivalent to **TP <serial no>**.

A typical interrogation response is shown below

```

TP 12

----- METATAPELIB: MT [000012] -----
Volume :DISK93244B/FILE000
Cycle      : 1                      Reel      : 1
Version    : 0                      Density   : BPI1250
Owner      : FLEX
Location   : TAPE RACK              Sent date  : 03/11/1994
Creation info: 09:05:16 on 01/09/1993
Savefactor : 30
Host info  : STIRLING1 #6343
Expiry date is UNKNOWN
Volume last accessed : 11:44:25 on 28/12/1994
Created by task 5012 : *LIBRARY/MAINTENANCE
Owned by job no 5011 : DUMP/DISK/93244B
Usage count  : 7 times              Last cleaned : 09/03/1992

Marked as   : VOLUMED
Rule info   : DISKDUMP
Notes      : TESTING ONLY
  
```

If the '@ <integer>' syntax is used, the history for a previous generation is shown in abbreviated form. The generations are numbered with the current version at zero and later generations with increasing numbers.

For example, to get historical information for generation 1 of serial number "000012" then:

```
TP 12 @1

----- METATAPELIB: MT [000012] GEN 1 -----
Volume :DBSPACK93132AM/FILE000
  Owner      : FLEX
  Creation info: UNKNOWN
  Host info   : #6343
  Created by task 8593
This label was overwritten 08:30:50 on 01/09/1994
```

A maximum of 5 generations is maintained by TRIM, though this can be controlled using the Metalogic INSTALL utility, using the TRIM config.

Note that the other modifiers cannot be used if the generation syntax is used, so that the information on older generations is **read-only** via the **TP** interface. <serialno> alone and '<serialno> @' are allowed from any source.

Specifying an **<update action>** modifier allows individual field update of a particular database entry, selected by serial number. This functionality allows diverse attributes such as tape owner, volume identifier, etc to be dynamically changed.

Multiple **<update action>**s may appear in one command input. In such cases, update modifiers, which use text-based fields, must be surrounded by single quotes, i.e.

```
TT TP 600032 OWNER 'META' LOCATION 'TIMBUKTU'
```

Otherwise the use of quotes is optional.

The description of each individual command modifier will be discussed in the rest of this section. Where applicable, examples of each command, with appropriate responses, will be given.

## BOX modifier

*He had forty-two boxes, all carefully packed,  
With his name printed on each:  
But, since he omitted to mention the fact,  
They were all left behind on the beach.*

Lewes Carroll, The Hunting of the Snark

— TP — <serial no> — BOX — <text> —

The **BOX** modifier sets the value of the BOXNUM field for the specified serial number. This field may be used at the discretion of the site as an indication of how or where individual tapes may be located by internal site procedures. The TRIM rule system does not set or use this field at any time.

The supplied <text> field may not exceed 6 characters.

**DASDL field:** BOXNUM  
**TAPEDB attribute:** BOXNUM (String)

There is no variant of the **TP FIND** command to show all tapes, which carry a valid BOXNUM attribute. However, a TP <serialno> response shows the BOXNUM field if one is present and user-supplied Opal TAPEDB SITUATIONS can be used to scan the database to search and report on tapes with specific attributes (see **Chapter 6: Getting started with OPAL** for more information).

### Example

```
TP X00034 BOX MYBOX
VOL: X00034 put in BOX MYBOX
```

## CDATE modifier

*I think it's wonderful you could all be here for the 43rd anniversary of my 39th birthday. We decided not to light the candles this year - we were afraid Pan Am would mistake it for a runway.*

Bob Hope, on his 82nd birthday, 1985



The **CDATE** modifier sets the CREATIONDATE attribute associated with the specified serial number. This attribute is intended to indicate when the tape was created and should not normally be changed. If the CREATIONDATE is changed, the **SR:** command must be performed if the tape has a time-based rule and an **APPLY** command, if the tape belongs to a generation-based rule.

<b>DASDL field:</b>	<b>CREATIONDATE</b>	
<b>TAPEDB attributes:</b>	<b>CREATIONDATE</b>	(String)
	<b>CREATIONDAY</b>	(Integer)

If the Supervisor option **USDATES** is set (for US sites), the **<date>** modifier expects a date format of MM/DD/YY. If the option is reset, you should use DD/MM/YY format. Note that in both cases, the year can be 2- or 4-digits values.

The **NEVER** modifier cancels any current creation date. The attribute is set to zero and will not be displayed in the response to a TP interrogation command.

### Examples:

```

TP 45 CDATE12/19/95
VOL: 000045 CREATION date set to 12/19/2002

TP X00036 CDATE NEVER
VOL: X00036 CREATION date has been cancelled
  
```



## CLEANED modifier

*C-l-e-a-n, clean, verb active, to make bright, to scour. W-i-n, win, d-e-r, der, winder a casement. When the boy knows this out of the book, he goes and does it.*

Charles Dickens, Nicholas Nickleby (1839)



Use of the **CLEANED** modifier indicates that the physical media has just been cleaned. The **CLEANDATE** attribute is set to the current date, the **USAGE** count to zero and the **DAMAGED** attribute is set to false. Each new use of the tape thereafter will cause the usage count to be incremented by 1.

<b>DASDL field:</b>	<b>CLEANDATE</b>	
<b>TAPEDB attribute:</b>	<b>CLEANDATE</b>	<b>(String)</b>
	<b>CLEANDAY</b>	<b>(Integer)</b>

If the **<unitno>** modifier is used, TRIM will check for the availability of the physical unit and will require an unlabelled, write-enabled tape to be mounted on that unit.

If so, and the serial number does not exist in the METATAPELIB database, or it exists, is marked as scratch, and has a usage count greater than zero, TRIM will automatically SN the tape at BPI6250 to the serial number specified in the command.

### Examples

```
TP A12300 CLEANED
VOL: A12300 CLEANDATE set to 12/07/2002
```

```
TP A12301 CLEANED 50
Cannot SN MT50-must be an unlabelled tape
```

## DAMAGED modifier

*And of course I've got this terrible pain in all the diodes down my left hand side.*

Douglas Adams, Hitch Hiker's Guide to the Galaxy



The **DAMAGED** modifier indicates whether the tape has been physically damaged in some way or is suspect. The tape keeps this damaged status even if it is subsequently re-used. To reset the status, use the "-" syntax or use the **CLEANED** modifier.

**DASDL field:** **DAMAGED**  
**TAPEDB attribute:** **DAMAGED** (Boolean)

If using Metalogic's Full Flex package, the FAMILYMANAGER utility will always attempt to make another backup of any file resident on a tape marked as **DAMAGED**.

FLEX may also set the **DAMAGED** flag if it detects a damaged tape using the FLEX Inquiry or MERGETAPE utilities.

### Examples:

**TP X00320 DAMAGED +**  
**VOL: X00320 marked as DAMAGED**

**TP X00320 DAMAGED -**  
**VOL: X00320 now repaired**

## DEFER modifier

*Tomorrow, I'll think of some way to get him back. After all, tomorrow is another day.*

Margaret Mitchell, *Gone with the Wind* (1936) ch. 57 (closing words)

— TP — <serial no> — DEFER —————|

The **DEFER** modifier is used to reset the **REPORTED** flag for the specified tape volume. This will prevent a subsequent **CONFIRM** command from changing the status of the specified tape after it has already been processed by the daily reporting phase.

DASDL field:           REPORTED  
TAPEDB attribute:   REPORTED   (Boolean)

**DEFER** is most useful when testing and verifying new or revised rules, or to abort a scheduled tape movement because of an unexpected problem.

### Example

TP 14 DEFER  
VOL: 000014 has been DEFER-ed

## DELETE modifier

*The Polis as Polis, in this city, is Null an' Void!*

Sean O'Casey, Juno and the Paycock

---

— TP — <serial no> — DELETE —

---

The **DELETE** command totally deletes all references, including history, of the selected serial number in the METATAPELIB database. Unlike the other <modifiers>, **this command may only be used by the Tape Librarian usercode**. This modifier does not correspond to any valid OPAL attributes.

Two possible ways exist to use the Tape Library usercode with this command. The most common way is to prefix the command with the FOR modifier, allowing the inheritance of a usercode to any command. So, from any COMS Supervisor window or ODT:

### Examples

```
TT FOR TAPELIB/PASSWORD TP X00035 DELETE
VOL: X00035 deleted from DB
```

Alternatively, the Supervisor **TERM USER** command can be used from a COMS Supervisor window to establish a session usercode, which will implicitly attach a usercode to every command issued from the terminal. The **TT TERM** command will **not** work for system ODTs but Supervisor will always respect their current TERM specifications.

### TERM USER

A log-on screen will appear, similar to the MARC log-on screen, allowing input of the Tape Librarian usercode/password so that future command input will be tagged with that usercode.

Once the **TERM USER** command has been used, with the Tape Librarian usercode, the **DELETE** command may be used without any **FOR** prefix.

## DESTROYED modifier

*Passion, you see, can be destroyed by a doctor. It cannot be created.*

Peter Shaffer, Equus

— TP — <serial no> — DESTROYED —

The **DESTROYED** modifier sets the DESTROYED flag in the database for the specified serial number. If the system has the CATALOGING option set, the TRIM system will issue a WFL VOLUME DESTROYED command to update the system Volume Library.

**DASDL field:** DESTROYED  
**TAPEDB attribute:** DESTROYED (Boolean)

The DESTROYED attribute cannot be manually reset using any TP command and will only be cleared when a new tape has been created with that same serial number.

### Example

TP BOB067 DESTROYED  
VOL: BOB067 marked as DESTROYED

## EXPIRY modifier

*CUSTOMER: I wish to complain about this parrot what I purchased not half an hour ago from this very boutique.*

*SHOPKEEPER: Oh yes, the Norwegian Blue — what's wrong with it?*

*CUSTOMER: I'll tell you what's wrong with it — it's dead that's what's wrong with it.*

*SHOPKEEPER: No, no — it's resting.... It's probably pining for the fjords....*

*CUSTOMER: It's not pining — it's passed on! This parrot is no more! It has ceased to be! It's expired and gone to meet its maker! This is a late parrot! It's a stiff! Bereft of life it rests in peace — if you hadn't nailed it to the perch it would be pushing up the daisies! It's rung down the curtain and joined the choir invisible! this is an ex-parrot!*

Monty Python's Flying Circus (BBC TV programme, 1969)



The **EXPIRY** modifier allows the user to change the current expiration date of the specified tape. The expiry date is used by the TRIM system to indicate when the PGOK attribute should be set and the tape returned to the SCRATCH POOL location for re-use.

<b>DASDL field:</b>	<b>EXPIRY</b>	
<b>TAPEDB attribute:</b>	<b>EXPIRYDATE</b>	(String)
	<b>EXPIRYDAY</b>	(Integer)

As a side-effect, the PENDINGDATE attribute of the tape will also be set to the new expiration date, allowing the manual change to be reported during the daily reports phase. Note that this can only occur if the date is **earlier** than any currently assigned PENDINGDATE.

The **NEVER** modifier cancels any current expiry date. The attribute is set to zero and will not be displayed in the response to a TP interrogation command.

If the Supervisor option **USDATES** is set (for US sites), the <date> modifier expects a date format of MM/DD/YY. If the option is reset, you should use DD/MM/YY format. Note that in both cases, the year can be 2- or 4-digits values.

**Examples:**

TP X00022 EXPIRY 12/1/96  
VOL: X00022 EXPIRY date set to 12/01/1996

TP 15 EXPIRY NEVER  
VOL: 000015 EXPIRY date has been cancelled

## FILEID modifier

*O thou! whatever title suit thee,  
Auld Hornie, Satan, Nick, or Clootie.*

Robert Burns, 'Address to the Deil' (1786)

---

TP      <serial no>      FILEID      <text>

---

The **FILEID** modifier is used to change the file identifier of the tape volume name specified by the given serial number.

**DASDL field:**            **FILEID**  
**TAPEDB attribute:**    **FILEID**      (**String**)

Since TRIM rules use the file identifier as part of the matching process, you may need to use the **SR** command to assign a new or revised rule. If the new rule is generation-based, you should use **APPLY** to update all members of the rule family.

The text field may not exceed 17 characters in length. If <text> is an empty string then the FILEID attribute will be also be set to an empty string.

### Example:

Assuming that the tape name for serial number X00012 is

**METATAPELIB/AUDIT21**

Is incorrect and the file identifier should be '2AUDIT21' instead of 'AUDIT21'

**TP X00012 FILEID 2AUDIT21**  
**VOL: X00012 FILEID changed to 2AUDIT21**



## HOSTNAME modifier

*Mankind is divisible into two great classes: hosts and guests.*

Sir Max Beerbohm, 'And Even Now' (1920) 'Hosts and Guests'

```
— TP — <serial no> — HOSTNAME — <text> —
```

The **HOSTNAME** modifier is used to change the name of the HOSTNAME that created this current version of the tape.

The <text> field may not exceed 17 characters and should conform to Unisys naming conventions.

<b>DASDL field:</b>	<b>HOSTNAME</b>	
<b>TAPEDB attribute:</b>	<b>HOSTORIGIN</b>	<b>(String)</b>

Note that the HOSTNAME field can be used for TRIM rule matching, the **SR** and **APPLY** commands should be used to update the rule family.

**Example:**

```
TP META02 HOSTNAME STIRLING
VOL: META02 HOSTNAME set to STIRLING
```



## JOBID modifier

*Gizza job.... I can do that.*

Yosser Hughes, Boys from the Blackstuff by Alan Bleasdale

— TP — <serial no> — JOBID — <text> —

The **JOBID** modifier is used to change the name of the job that created this current version of the tape. Similarly, the **TASKID** modifier may be used to change the name of the task that created this tape. The <text> field may not exceed 28 characters and should conform to Unisys naming conventions though TRIM will attempt to make sense of the name if it is invalid.

**DASDL field:** JOBID  
**TAPEDB attribute:** JOBID (String)

Note that the JOBID field is used to match special TRIM rules for unlabelled tapes (rule names are prefixed by the identifier 'UL\_') and, if a JOBID attribute is changed for an unlabelled tape, the **SR** and **APPLY** commands should be used to update the rule family.

### Example:

```
TP META02 JOBID (META)WFL/CREATE/COPY
VOL: META02 JOB ID set to (META)WFL/CREATE/COPY
```

## JOBNO modifier

*I am not a number, I am a free man*

The Prisoner, 1960's British TV series

— TP — <serial no> — JOBNO — <integer> —

The **JOBNO** modifier allows the user to change the job number of the program or job that originally created the tape, specified by the given serial number. The <integer> field may take a value between 1-9999 and cannot exceed 4 digits.

**DASDL field:** JOBNO  
**TAPEDB attribute:** JOBNO (Integer)

The **JOBNO** attribute for a tape volume is of particular importance when counting generations for a generation-based TRIM rule. The TRIM system considers multiple tapes created from the same job (as dictated by **JOBNO**) to be members of the same generation.

Note that tapes with the same creator job number must be adjacent chronologically within the family. Using the **JOBNO** modifier for such tapes may require the use of the **APPLY** command to recalculate current locations for all members of the rule family.

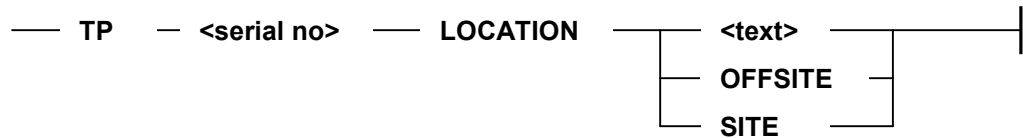
### Example:

**TP A99999 JOBNO 5678**  
**VOL: A99999 JOBNUMBER set to 5678**

## LOCATION modifier

*For my part, I travel not to go anywhere, but to go. I travel for travel's sake. The great affair is to move.*

Robert Louis Stevenson, Travels with a Donkey



The **LOCATION** modifier sets the latest location of the specified serial number to that specified by the <text> field.

**DASDL field:**            **LOCATION**  
**TAPEDB attribute:**   **LOCATION**   (**String**)

The **LOCATION** attribute is maintained by the TRIM rules system and should not normally be changed. When changing **LOCATION**, the **SENDDATE** and **SENDTIME** Opal attributes are automatically updated to reflect the date and time of the change (see also the **SEND** modifier).

The <text> for **LOCATION** may not exceed 12 characters. Please note spaces inserted inside the text string will be respected and retained.

Changing the location of a tape in this way can be futile if the tape in question is assigned to a rule that has its own defined location controls. If a pending action has already been marked against the tape (i.e. the **PENDINGDATE** is less than or equal to today), the next scheduled reports and **CONFIRM** phase will always override the changed location with that calculated from the rule.

Similarly, the **APPLY** command will always verify the current location of each tape in the family, setting up a new pending location if necessary. To make a location change permanent in such cases, any pending actions must be cancelled (see the **PDATE** command used with the **NEVER** modifier ).

On CATALOGING capable systems running the Full FLEX package, the locations OFFSITE and SITE have special significance. As expected, the tape attribute LOCATION will be changed to OFFSITE or SITE and will also have the following actions:

If the MCP is 4.1 or later, the Volume Library entry for the tape is marked with the volume offsite flag set to TRUE if the LOCATION is OFFSITE, or it is set to FALSE if the LOCATION is set to SITE. TRIM simulates the appropriate WFL VOLUME OFFSITE or ONSITE command to achieve this.

A tape's OFFSITE or ONSITE status can be seen in the standard response to the ODT command PV MT<serial>.

The FLEX statistics file also marks the volume as being OFFSITE or SITE. This feature is used by both the MERGETAPE and FAMILYMANAGER utilities when creating and loading tapes.

#### Examples:

```
TP 25 LOCATION TAPE RACK
VOL: 00025 sent to TAPE RACK
```

```
TP A00012 LOCATION SITE
VOL: A00012 sent to SITE
```

```
TP 51 LOCATION OFFSITE
VOL: 000051 sent to OFFSITE
```

## LOST modifier

*For when the One Great Scorer comes to mark against your name,  
He writes—not that you won or lost—but how you played the Game.*

Grantland Rice, Only the Brave



The **LOST** modifier is used to indicate that the tape has been “lost” and is temporarily unavailable. Note that the current **LOCATION** is not modified.

**DASDL field:**            **LOST**  
**TAPEDB attribute:**   **LOST**        (**Boolean**)

The **LOST** field will be automatically reset if the tape is used or updated. If using the Full **FLEX** package, the **FAMILYMANAGER** utility will attempt to make another backup of any file currently resident on a **LOST** library maintenance tape.

### Examples:

**TP 12 LOST +**  
**VOL: 000012 marked as LOST**

**TP 12 LOST-**  
**VOL: 000012 rediscovered**

## MIXID modifier

*Titles are but nick-names, and every nick-name is a title.*

Thomas Paine, The Rights of Man

— TP — <serial no> — MIXID — <text> —

The **MIXID** modifier is used to change the name of the task that created the specified tape. . The <text> field may not exceed 28 characters and should conform to Unisys naming conventions. Note that unlike many of the other modifiers, TASKID is **not** a synonym for MIXID.

**DASDL field:** TASKID  
**TAPEDB attribute:** TASKID (String)

Note that the TASKID attribute is used to match special TRIM rules for unlabelled tapes (rule names prefixed by 'UL\_') and, if a TASKID attribute is changed for a specific unlabelled tape using the **MIXID** modifier, the **SR** and **APPLY** commands may be used to update the rule family.

Example:

```
TP META02 MIXID (META) SYSTEM/DUMPALL
VOL: META02 TASK ID set to (META) SYSTEM/DUMPALL
```



## MIXNO modifier

*Well, if I called the wrong number, why did you answer the phone?*

James Thurber, New Yorker cartoon, 5 June 1937

— TP — <serial no> — MIXNO — <integer> —

The **MIXNO** modifier allows the user to change the task number of the program/WFL task that created the tape as specified by the serial number. The <integer> field may take a value between 1-9999 and cannot exceed 4 digits.

<b>DASDL field:</b>	<b>TASKNO</b>	
<b>TAPEDB attribute:</b>	<b>MIXNUMBER</b>	(Integer)
	<b>MIXNO</b>	(Integer)

Unlike the **JOBNUMBER** attribute, **MIXNUMBER** is **not** used by the TRIM system when counting generations within a generation-based rule family. The **SR** and **APPLY** commands are not required after changing the value of this field.

**Example:**

```
TP A99999 MIXNO 5678
VOL: A99999 MIXNUMBER set to 5678
```

## NOTE modifier

*O wretched fool!*  
*That liv'st to make thine honesty a vice.*  
*O monstrous world! Take note, take note, O world!*  
*To be direct and honest is not safe.*

William Shakespeare, Othello

— TP — <serial no> — NOTE — <text> —

The **NOTE** modifier permits the setting of a free-format text field within the specified tape entry available for any convenient user-selected purpose. The <text> field may be a string up to 72 characters in length.

<b>DASDL field:</b>	<b>NOTES</b>	
<b>TAPEDB attribute:</b>	<b>NOTES</b>	<b>(String)</b>

If the Full FLEX package is being used with TRIM, the MERGETAPE utility will unconditionally update this field with specific information about the release of tapes.

### Example:

```
TP META04 NOTE CUSTOMER TAPE SENT 14/04/2002
VOL: META04 NOTED: CUSTOMER TAPE SENT 14/04/2002
```

## OFFSITE modifier

*'Glorious, stirring sight!' murmured Toad, never offering to move. 'The poetry of motion! The real way to travel! The only way to travel! Here today—in next week tomorrow! Villages skipped, towns and cities jumped—always somebody else's horizon! O bliss! O poop-poop! O my! O my!'*

Kenneth Grahame, *Wind in the Willows*

— TP — <serial no> — OFFSITE —

The **OFFSITE** modifier is a synonym for 'LOCATION OFFSITE'. It is used to indicate that a physical tape has been sent to an off-site storage location.

As before, with the **LOCATION** modifier, changing a tape location automatically updates the **SENDDATE** and **SENDTIME** attributes.

Note that if Metalogic's Full FLEX package is installed on the system, sending tapes off-site changes the tape entry in the system Volume Library. Please see the **LOCATION** modifier for more details.

Example:

```
TP A00045 OFFSITE
VOL: A00045 sent OFF SITE
```

## OWNER modifier

*An ill-favoured thing, sir, but mine own.*

William Shakespeare, As You Like It

— TP — <serial no> — OWNER — <text> —

The **OWNER** modifier is used to change the name of the specified tape's "owning" usercode

<b>DASDL field:</b>	<b>OWNER</b>	
<b>TAPEDB attribute:</b>	<b>OWNER</b>	<b>(String)</b>
	<b>USERCODE</b>	<b>(String)</b>

The **OWNER** field is very important because of its involvement in the matching process when searching the rules database after an **SR** command or tape creation. For this reason, a tape entry should be **SR**-ed followed by an **APPLY** (if the rule is generation-based) whenever the **OWNER** of a tape is changed using this modifier.

The <text> string may not exceed 17 characters.

Since SUPERVISOR does not get the usercode of the task that created the tape from the MCP, SUPERVISOR will track internally tasks creating tapes and hold usercode information until the MCP has sent a tape label event. On rare occasions, SUPERVISOR may not be able to do this, often because the system has been extremely busy. This may also occur when the LOGREADER utility has run after a restart or halt-load, and LOGREADER is unable to determine the usercode from the SUMLOG. In such cases, a default owner usercode of 'UNKNOWN' will be assigned and can be seen in the TAPELOG or in the response to a **TP <serialno>** inquiry.

If the usercode is incorrect, then it is likely that the rule associated with the tape will be incorrect as well. Using the **OWNER** modifier to correct the usercode followed by an **SR** will resolve the problem.

You can easily monitor tapes that have 'UNKNOWN' usercode by using the **FIND NAME** command with a wildcard:

**TP FIND NAME (UNKNOWN)=**

Example:

**TP A00423 OWNER META**  
**VOL: A00423 OWNER set to META**

## PEND modifier

*The man is quiet, but firm as a rock, yet sensitive to the first imperceptible sign of impending changes. He does not delay in taking action.*

I Ching (BC 1150)



The **PEND** modifier establishes a new pending date for a scheduled location movement for the specified tape.

<b>DASDL field:</b>	<b>PENDING-DATE</b>	
<b>TAPEDB attribute:</b>	<b>PENDINGDATE</b>	(String)
	<b>PENDINGDAY</b>	(Integer)

A valid pending location should have already been set for the specified serial number by using the **PEND** modifier. If the **NEVER** modifier is used, the pending date is set to zero, the current pending location is also removed and the **REPORTED** attribute is reset.

The **PEND** modifier can be used to assist in the cancellation of a pending tape movement usually handled by **CONFIRM**.

If the Supervisor options **USDATES** is set, the date format MM/DD/YY should be used and, if reset, the alternate date format DD/MM/YY.

### Examples:

```

TP 21 PENDING 14/12/2002
VOL: 000021 PENDING date set to 14/12/2002
  
```

```

TP X00032 PENDING NEVER
VOL: X00032 PENDING date cancelled
  
```

## PGNEVER modifier

*Wood may remain ten years in the water, but it will never become a crocodile*

Mongolese proverb



The **PGNEVER** modifier indicates that the specified tape volume can never expire or be eligible for purging (also known as “permanent”).

DASDL field:	PGNEVER	
TAPEDB attribute:	PGNEVER	(String)

Further, TRIM inhibits the use of the **PGOK** modifier to mark any tape as “scratchable” if it is currently marked as **PGNEVER**. The **PGNEVER** modifier should be used with ‘+’ to set and ‘-’ to reset. When used with the Full FLEX package, the **PGNEVER** flag has the additional meaning that this tape can never be a candidate for merging by the MERGETAPE utility.

When the **CONFIRM** process encounters a tape which has the **PGNEVER** attribute set and it's pending movement is to the SCRATCH POOL, the TRIM system will cancel the pending movement and date and the tape's location will remain unchanged.

**Examples:**

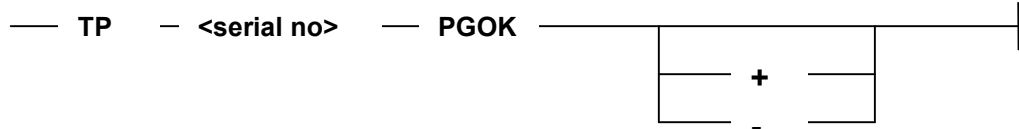
```
TP 11 PGNEVER +
VOL: 000011 PERMANENT (NEVER purge) set

TP 11 PGNEVER -
VOL: 000011 PERMANENT (NEVER purge) reset
```

## PGOK modifier

*I'll purge, and leave sack, and live cleanly, as a nobleman should do.*

William Shakespeare (Henry IV, Part 1)



The **PGOK** modifier sets or resets the purge authorisation flag (also known as PGOK) for the specified serial number in the database. The '+' modifier will set the flag, '-' to disable; if neither is present, then '+' is assumed.

<b>DASDL field:</b>	<b>PGAUTHORISED</b>
<b>TAPEDB attribute:</b>	<b>PGOK (Boolean)</b>

The PGOK attribute, when set, informs the TRIM software that this tape can be purged unconditionally unless the TAPEMANAGER software is active (see **Chapter 7:TAPEMANAGER implementation** for more information on purge restrictions when TAPEMANAGER is active). The value of the PGOK attribute is used by the TRIM rules system to signify that a tape has expired because of retention limits imposed by the rule. It is normally set during the **CONFIRM** process and usually accompanied by a location movement to SCRATCH POOL.

### Examples:

```
TP A00012 PGOK +
VOL: A00012 purge authorised
```

```
TP 45 PGOK -
VOL: 000045 purge no longer authorised
```

If the Full FLEX system is available on a CATALOGING capable MCP, trying to set the PGOK flag for a tape volume that has backup references is not allowed.

```
TP META93 PGOK
VOL: 000045 cannot mark PGOK: tape has backup refs
```



## PLOC modifier

*Like Webster's Dictionary,  
We're Morocco Bound*

Song from the film 'Road to Morocco'

— TP — <serial no> — PLOC ————— <text> ————— |

The **PLOC** modifier permits the dynamic change of the current pending location for the specified serial number (this is the OPAL attribute, PENDINGLOCATION ) to that specified by the <text> field.

<b>DASDL field:</b>	<b>PENDING-LOCATION</b>	
<b>TAPEDB attribute:</b>	<b>PENDINGLOCATION</b>	<b>(String)</b>

The PENDINGDATE will be automatically changed to today's date and the REPORTED attribute will be reset.

As with the **LOCATION** modifier, **PLOC** will only accept a text field up to 12 characters in length. Spaces inserted in the string will be respected. If the tape has a valid TRIM rule, changing the current pending location will be detected by the **CONFIRM** phase. **CONFIRM** will issue a warning and will, in most cases, authorise the move but will automatically assign the authorised next location, as determined from the rule. Similarly, an **SR** or **APPLY** command will also re-assign any manually changed pending location.

If the pending location of a tape is required to be manually changed (for example, to allow a tape to be sent to a one-time only location), the **PEND** modifier should be used to establish a new pending date either before or after the **CONFIRM**.

**Example:**

```
TP 11890 PLOC BAHAMAS
VOL: 11890 will be sent to BAHAMAS
```

## REPORTED modifier

*The report of my death was an exaggeration.*

Mark Twain, New York Journal, 2 June 1897

— TP — <serial no> — REPORTED —

The **REPORTED** modifier marks the selected volume as being eligible for action by a **CONFIRM** command though the tape must also have valid **PENDINGDATE** and **PENDINGLOCATION** assignments.

**DASDL field:** **REPORTED**  
**TAPEDB attribute:** **REPORTED** (Boolean)

This **REPORTED** attribute is automatically set by the daily report OPAL programs but may be subsequently cancelled using the **DEFER** modifier. In fact, any OPAL **SITUATION** of type **TAPEDB** which uses the **PENDING** subcontext will automatically set the **REPORTED** flag when inspecting each eligible tape in the **PENDING** subset (i.e. the tape must have a valid pending location and a pending date earlier or equal to today's date).

For example:

```
TT DEFINE + SITUATION MAKE_REPORTED (TAPEDB=PENDING) :
    TRUE
```

Note that the **REPORTED** attribute is always automatically reset should a **TP PLOC** modification command be used on any individual tape.

Example:

```
TP A00123 REPORTED
VOL: A00123 is now marked as REPORTED
```

## RETURN modifier

*There was a young lady named Bright,  
Whose speed was far faster than light;  
She set out one day  
In a relative way  
And returned on the previous night.*

Arthur Buller, Punch 19 Dec. 1923, 'Relativity'



The **RETURN** modifier allows the setting of a special date field indicating the date that the specified tape was "returned". This date value is available for use at the site's discretion and need not specify when a tape actually returned to its current location. Note that RETURNDATE is NOT currently used by the TRIM system.

**DASDL field:** RETURNDATE  
**TAPEDB attribute:** RETURNDATE (Boolean)

If the Supervisor option **USDATES** is set, the date format MM/DD/YY should be used, if reset the DD/MM/YY format is required.

The **NEVER** modifier cancels any current RETURNDATE assignment by setting the date to 0 and will not be displayed in the response to a TP interrogation.

### Examples:

```
TP 15 RETURN 12/12/95
VOL: 000015 will be returned on 12/12/2002
```

```
TP 15 RETURN NEVER
VOL: 000015 RETURN date has been cancelled
```

## SEND modifier

*I sent the club a wire stating, please accept my resignation. I don't want to belong to any club that will accept me as a member.*

Groucho Marx

```

— TP — <serial no> — SEND — [ <date> ]
                                   [ NEVER ]

```

The **SEND** modifier updates date fields within METATAPELIB representing the date that the specified was send to its current location. These attributes, called **SENDDATE** and **SENDTIME**, are also updated by the TRIM system whenever the **BOX**, **LOCATION** and **OFFSITE** modifiers are used and are set to the current date and time. The **SITE** modifier or a new creation of any tape volume automatically resets these fields to null.

The **SENDTIME** and **SENDDATE** attributes are NOT updated by the **CONFIRM** command at any time.

<b>DASDL field:</b>	<b>SENDDATE</b>	
<b>TAPEDB attribute:</b>	<b>SENDDATE</b>	(String)
	<b>SENDDAY</b>	(Integer)

If the Supervisor option **USDATES** is set , the date format MM/DD/YY should be used, if reset the DD/MM/YY format is required.

The **NEVER** modifier cancels any current **SENDDATE** assignment by setting the date to 0 and will not be displayed in the response to a TP interrogation.

### Examples:

```

TP 12 SEND 13/12/2002
VOL: 000012 SENT on 12/12/2002

TP A00456 SEND NEVER
VOL: A00456 SEND date has been cancelled

```

## SITE modifier

*Home James, and don't spare the horses.*

Fred Hillebrand, title of song (1934)

— TP — <serial no> — SITE —

The SITE modifier is a synonym for 'LOCATION SITE'. It is used to indicate that a physical tape has been sent back to an on-site storage location. Changing a tape location will automatically update the SENDDATE and SENDTIME fields in the database.

Note that if Metalogic's Full FLEX package is installed on the system, sending tapes on-site may change the tape entry in the system Volume Library. **Please see the LOCATION modifier for more details.**

Example:

```
TP A00045 SITE
VOL: A00045 sent ON SITE
```

## SR modifier

*Rowe's Rule: the odds are five to six that the light at the end of the tunnel is the headlight of an oncoming train.*

Paul Dickson, Washingtonian 1978

— TP — <serial no> — SR —

The **SR** (Set Rule) modifier will apply TRIM's rule searching algorithms against the specified serial number. The TRIM system will use the owner, tape volume and file identifiers and creation date fields to determine whether a matching rule is found.

If the **SR** operation finds a match, the database record for the specified tape is updated with identity information about the assigned rule.

**Database fields:**      **RULE-USER, RULE-ID, RULE-INX**  
**TAPEDB attributes:**   **RULEOWNER, RULEID, RULEINDEX**

The **RULEID** attribute refers to the name or identity of the rule and the **RULEOWNER** attribute refers to the owning usercode of the rule. Remember that if no matching rule is found, the default **EXCEPTION** rules (usercoded or un-usercoded) will be applied.

The **SR** modifier will also assign an expiry date, if the rule is time-based, and a default pending location and date, if a location movement list exists in the rule. For generation-based rules, **SR** will not consider other generations in the family so each **SR**-ed tape will always be considered as though the tape had just been created i.e. it is the most recent generation.

When **SR**-ing a tape which is assigned a generation-based rule, the following change is unconditionally made to the **NOTES** attribute and can be seen in the response to a **TP <serial>** query:

**Note: Use 'TP APPLY <rule fam>' to action GEN rules**

To correctly update all entries in a generation-based rule family, the **APPLY** command must be issued on the appropriate family. This is only necessary if the **NOTES** attribute for the tape corresponds to that shown above.

The SR command will automatically set the PGNEVER flag for any matching rule that uses a NEVER retention.

**Examples:**

```
TP 61 SR
VOL: 000061 Rule applied
```

```
TP A12056 SR
VOL: A12056 Rule applied -- GEN rules not done --
```

## SYSNO modifier

*I must create a system, or be enslaved by another man's.  
I will not reason and compare: my business is to create.*

William Blake, 'Jerusalem'

— TP — <serial no> — SYSNO — <integer> —

The **SYSNO** modifier is used to change the assigned serial number of the A-Series system on which this tape was created.

**Database field:**        **SYSNO**  
**TAPEDB attribute:**   **SYSNO**

The SYSNO attribute is not currently used by the TRIM software so changing this setting has no effect on TRIM's rule-handling functionality.

The value of <integer> may not exceed 65535.

**Example:**

**TP A00023   SYSNO 6343**  
**VOL: A00023 SITENO set to 6343**



## UNASSIGN modifier

*Whenever I hear the word culture...I release the safety-catch of myBrowning!*

Hanns Johst, 'Schlageter' (1933) act 1, sc. 1

—— TP —— <serial no> —— UNASSIGN ——

The **UNASSIGN** modifier allows a tape that has been previously assigned (e.g. requested for a tape mount) to be returned to normal. This only applies if the tape is SCRATCH or PGOK and has been returned, as the next available tape, by the OPAL function GETSCRATCH for use by the calling OPAL program.

Typically, **UNASSIGN** would be needed if the proposed usage of the tape did not occur.

The current list of assigned tapes can be seen with the **TP ASSIGN** interrogation command.

## VOLID modifier

*The life of every man is a diary in which he means to write one story, and writes another; and his humblest hour is when he compares the volume as it is with what he vowed to make it.*

Sir J. M. Barrie, 'The Little Minister'

— TP — <serial no> — VOLID — <text> —

The **VOLID** modifier command is used to change the volume identifier of the tape name currently assigned to the specified serial number.

Database field:	<b>VOLID</b>	
TAPEDB attribute	<b>VOLID</b>	(String)
	<b>VOLUMEID</b>	(String)

Since the TRIM system uses the volume identifier as part of the rule matching process, you may need to use the **SR** command to assign a new or revised rule. If the new rule is generation-based, you should use **APPLY** to update all members of the rule family.

The text field may not exceed 17 characters in length.

### Example:

Assuming that the tape name for serial number X00012 is

**DBSPACK95123/AUDIT21**

Is incorrect and the file identifier should be 'METATAPELIB' instead of 'DBSPACK95123'

**TP X00012 VOLID METATAPELIB**  
**VOL: X00012 VOLUMEID changed to METATAPELIB**

If the volume id of a tape has been altered by a **TP . .VOLID** command, then subsequent attempts to purge the tape would normally be denied if TAPEMANAGER purge protection is active. This is because of a mismatch between the actual label and database volume identifiers.

To resolve this SITUATION, when the VOLID modifier is used to change a tape name, the previous volume id will be stored in the unused ACCESSCODE field of the database record. If present, the old label will be displayed in the response to a TP <serial> command.

```
TP 1

----- METATAPELIB: MT [000001] -----
Volume : TESTNAME/FILE000
Cycle      : 1                      Reel      : 1
Version    : 0                      Density    : BPI1250
Owner      : FLEX
Location   :
Created at  : 14/07/2004
Savefactor : 30
Host info   : STIRLING #1           Old volume  : FLEXCOPY2004153
Pending loc : EXCEPTION             Pending date : 14/07/2004
Expiry date is UNKNOWN
Volume last accessed : UNKNOWN
Created by task 03744 : DATA_ENTRY
Owned by job no 03743 : DATA_ENTRY
Usage count : NONE

Marked as   : NON-VOLUMED
Rule info   : EXCEPTION
```

If TAPEMANAGER is active then the old volume id will also be checked automatically against the actual tape name when a purge request is verified.

---

## Reporting and rule processing commands

The TRIM system has very powerful reporting tools, allowing information retrieval about all tapes in the METATAPELIB database in a variety of ways. Most of these commands use the **FIND** prefix followed by another modifier indicating the DMSII path used to retrieve data. All commands prefixed by **TT PRINT** will divert all output to a printer backup file.

When used from a COMS window or REMOTESPO, each **FIND** command will return a maximum of 200 tapes and may be paged from the datacom station. This limit does NOT apply if the TT PRINT prefix is used and all tapes found will be returned. Most of the FIND commands will take wildcard patterns for the target used in the search.

It should be noted that Opal programs are capable of generating all these reports. All the fields in each database entry can be easily retrieved and manipulated as Opal attributes. It should be noted that by default all Opal attributes are **read-only**. However, as already discussed in the previous section, there is a large subset of database fields that may be changed dynamically using TP <serial> modifiers.

Other commands discussed in this section are critical to the effectiveness of TRIM rule mechanisms; in particular, **CONFIRM**. This command authorises all pending rule actions, which involve location movements and marking expired tapes as purge authorised (i.e. the PGOK flag is set) for those tapes that have been released by the system. Also, the **APPLY** command can be used to quickly update tape families that have had their rules changed by the TRIM utility.

## APPLY command

*You must remember this, a kiss is still a kiss,  
A sigh is just a sigh;  
The fundamental things apply,  
As time goes by.*

Herman Hupfeld, As Time Goes By (1931 song)

— TP ——— APPLY ———— <rule family> ————|

The **APPLY** command is used to ensure that all tapes volumes currently members of the selected <family id> rule have correct location and expiry assignments. **APPLY** will verify all members of the family, processing each in turn from the most recent to the oldest.

For generation-based families, tapes which are older than the oldest possible generation maintained by the rule will be marked pending location SCRATCH POOL and expiry date will be marked as today. These tapes will be released during the next **CONFIRM** phase.

**APPLY** is particularly useful when the characteristics of a TRIM rule have changed, since this allows any new or revised controls to be easily propagated throughout all members of the family. If new tapes need to be included as part of the family, they will have to have already been **SR**-ed.

**APPLY** will check all members in the family as seen in the response to a **FIND FAMILY** command. Note that <rule family> may be composed of a rule owner, identity and index.

Examples:

```
TP APPLY BLANKRULE
No matching volumes affected by this rule
```

```
TP APPLY (META)EXAMPLERULE/1
12 volumes affected by APPLY command
```

## ASSIGN command

*My Dear One is mine as mirrors are lonely.*

W.H.Auden, 'The Sea and the Mirror--Miranda' (1944)

— TP ——— ASSIGN ———

The ASSIGN command returns a list of tapes currently marked as 'assigned' in the METATAPELIB4 database. Such tapes are only ever marked by calls of the OPAL function GETSCRATCH, which may be used by OPAL, scripts to return the next available SCRATCH or PGOK tape, according to SCRATCHPOOL and DENSITY.

The <density> modifier allows **ASSIGN** to filter the returned volumes by tape density.

```
TP ASSIGN
----- FIND: PGOK ASSIGNEDF -----
----- FIND: PGOK (FMT36TRK) ASSIGNED -----
C12724: A S S I G N E D to POWEREDGE          at 14:42 20/04/2004

----- FIND: PGOK (BPI1250) ASSIGNED -----
000110: A S S I G N E D to POWEREDGE          at 12:31 31/03/2003
000111: A S S I G N E D to POWEREDGE          at 09:08 29/01/2004
077172: A S S I G N E D to at 10:18 30/07/2001

----- FIND: PGOK (BPI11000) ASSIGNED -----
----- SCRATCH POOL: DBPOOL -----
DSI070: A S S I G N E D to POWEREDGE          at 10:53 16/04/2003

----- FIND: PGOK (FM TDLT35) ASSIGNED -----
----- SCRATCH POOL: DLT35 -----
DLT003: A S S I G N E D to POWEREDGE          at 11:11 16/12/1999

----- FIND: PGOK (BPI1250) ASSIGNED -----
----- SCRATCH POOL: LONGPOOLNAME -----
Z00028: A S S I G N E D to INSPIRONMCP        at 12:30 31/03/2003

----- FIND: PGOK (BPI6250) ASSIGNED -----
----- SCRATCH POOL: STIRLING -----
R70890: A S S I G N E D to POWEREDGE          at 11:11 16/12/1999
----- 12 ENTRIES FOUND -----
```

Tapes that have been assigned by GETSCRATCH are marked for automatic loads by tape robot systems such as DSI and StorageTek™.

## CLOSE command

*Goodbye to all that.*

Robert Graves, title of autobiography

---

### TP CLOSE

---

The **CLOSE** command issues a close down message to the METATAPELIB database usually to allow off-line dumps to take place. The TRIM system will inform the operator that the database has been successfully closed. This command is only valid for MASTER systems.

Once closed, all **TP** commands, except **OPEN**, **LOG** and **VER**, will be rejected with the message:

**\*\*\* METATAPELIB database is unavailable \*\*\***

While the database is down, the TRIM system will still be receiving tape events and possibly TP commands issued from Supervisor ODTSEQUENCES. These inputs are queued by TAPELIBUPDATER and will be re-processed when the database has returned on-line by using the **TP OPEN** command. An entry will be made into the TAPELOG prior to the re-processing indicating how many inputs and events were queued.

**10:52:19 Msg: Re-processing 11 queued events/TP commands**

If the METATAPELIB database is closed whilst the TAPEMANAGER system is active, all operator and programmatic tape purge commands will fail, returning an error message into the TAPELOG:

**10:52:04 Err: DENY PG OF 000021 (Database offline)**

The **CLOSE** command should always be used rather than an operator DS. If the TAPELIBUPDATER task is DS-ed, it is possible that pending tape event information may be lost.

The database can only be re-opened using the **TP OPEN** command.

```
TP CLOSE
METATAPELIB database has been closed
```

It is strongly recommended that, like all DMSII database systems, the METATAPELIB database and audit trail be regularly backed up.

With the advent of DMUTILITY on-line dumping capabilities, it is not necessary to take the database off-line using **TP CLOSE**. However, the Metalogic INSTALL utility will issue a **TP CLOSE** command during the installation of a new release.

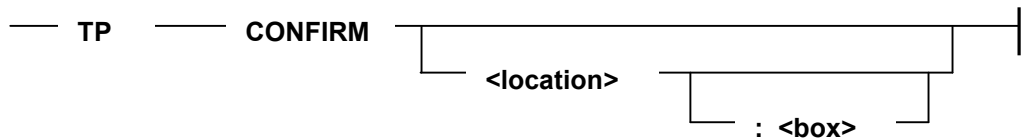
Alternatively, the **TP DUMP** command also allows the dumping of the entire METATAPELIB database into flat files. Please refer to the DUMP command for more information.



## CONFIRM command

*Possessed, as are all the fair daughters of Eve, of an hereditary propensity, transmitted to them undiminished through succeeding generations, to be 'soon moved with the slightest touch of blame'; very little precept and practice will confirm them in the habit, and instruct them in all the maxims, of self-justification.*

Maria Edgeworth, Letters for Literary Ladies (1795)



The **CONFIRM** command causes outstanding pending actions to be performed for all tape volumes which have the REPORTED attribute set. A tape should only be marked as REPORTED if it has a valid pending location scheduled (set by the PENDINGLOCATION attribute) and a pending date less than or equal to the current date (held in the PENDINGDATE attribute).

Volumes are marked as REPORTED by the **TP <serialno> REPORTED** command or, implicitly, by using an OPAL SITUATION whose context is TAPEDB=PENDING.

Note that the Metalogic-supplied Opal programs, called TPDB\_REPORT, are responsible for determining those tapes that need to be marked as REPORTED. These programs are automatically executed as part of the daily reports phase that requires the invocation of a Supervisor ODTSEQUENCE:

```
TT DO TPDB_DAILYREPORTS
```

This command will generate a printed report detailing all scheduled pending tape movements and should be manually checked prior to a **CONFIRM**. It is intended that this mechanism will be superseded by a top-level **TP** command in a later release.

**CONFIRM** checks that the movement is valid and subsequently changes the pending location to the actual location and resetting the pending date to null.

During the **CONFIRM** process, TRIM logs any errors and warnings into the TAPELOG. These should be periodically checked, using the command **TP LOG CON**, to ensure that no problems have been reported.

If a tape moves to the SCRATCH POOL location as a result of the **CONFIRM** process, the PGOK flag will be automatically set.

Optionally, the **CONFIRM** command will accept a location modifier which restricts authorisation to those tapes with a pending location that matches the specified location. Further, the use of the '**:<box>**' option, in conjunction with the **<location>** modifier, allows a user-selected box number field to be set. This provides the site with extra user-selected controls over tape movements, if required.

The BOXNUMBER attribute for each confirmed volume will not be changed after a **CONFIRM** unless **<box>**modifier was NOT supplied with the original command. Previously, manually assigned box information was automatically deleted during the **CONFIRM**.

#### Examples:

```
TP CONFIRM
Volumes confirmed 22 (warnings 1),Links 4,Errors 0
```

To confirm all tape movements with a pending location of 'VAULT' to box BOX123:

```
TP CONFIRM VAULT:BOX123
Volumes confirmed 1 (warnings 0),Links 0,Errors 0
```

## CONFIRM errors and warnings

The **CONFIRM** command may generate various warning and information messages which are visible from the TAPELOG using the **TP LOG CON** command. The TRIM-generated response always indicates how many errors or warnings were generated during the **CONFIRM** process. Even if warnings are detected, **CONFIRM** will always move a tape to the location indicated in the daily report but will automatically establish a new pending location and date for the correct location.

Explanations of the possible messages that can be encountered are shown below. Each entry is typical of how it appears in the TAPELOG.

**17:43:29 Msg: Volumes confirmed 1 (warnings 1),Links 0,Errors 0**

This is an information message indicating the number of tapes correctly processed, plus any errors and warnings. If warnings or links are reported then it is advisable to run another daily report and **CONFIRM** as soon as possible. The links entry refers to the number of tapes detected which are linked, via a TRIM rule, to master tapes which have changed location status during the **CONFIRM** process. As a side effect of **CONFIRM**, these linked tapes will have been set up with new pending locations and dates inherited from their masters.

**17:43:29 Con: TRIM39 confirmed not PGOK should be VAULT**

The above message should be considered as a warning. The location of the tape has been changed to SCRATCH POOL as requested but as the current location should have been VAULT, the pending location will have been changed to VAULT and the PGOK will NOT be not set. Typically, this message will be seen for tapes, which have been processed by a **TP <serial> PLOC** command.

**17:39:52 Con: TRIM40 confirmed while reported, should be STIRLING**

This message is also a warning. Here, a new tape, belonging to a generation-based rule family, has been created in the interval between the production of the daily report and the **CONFIRM**. The pending location of the TRIM40 tape, which will now be a later generation in the family, should really be in STIRLING. **CONFIRM** will authorise the current movement but will establish a pending location of STIRLING and a pending date set to today's date.

**16:54:46 Con: TRIM55 confirmed no PGOK (PGnever)**

In this case, TRIM55 has been moved to the SCRATCH POOL but the PGNEVER flag is set which means that this tape is never eligible for purging. As a consequence, **CONFIRM** will automatically assign a pending location of the last valid location, as provided by the TRIM rule. In normal operations, tapes with the PGNEVER flag set should never get a pending location of SCRATCH POOL at any time. If this has happened, it is most likely that the PGNEVER flag was set after the tape was already pending scratch or the **PLOC** modifier was used to manually alter the pending location.

**16:54:46 Con: TRIM54 not confirmed (no longer pending)**

This is the only case where the tape volume is not confirmed to its pending location. In the time since it has been marked as REPORTED, the pending date has been changed and is no longer valid. This will also increment the errors total as seen in the CONFIRM response.

**10:28:01 Con: TRIM56 Location via Link BAHAMAS**

This information message indicates that volume TRIM56 has had its pending location changed to BAHAMAS since the volume it was linked to has already moved there during the CONFIRM. These linked tapes will then move to the same location as the master after the next daily reports and CONFIRM phases. This message also causes the links total, seen in the CONFIRM response, to be incremented by one.

**10:04:35 Con: TRIM57 confirmed should be OFFSITE**

Usually, this message will be seen after a TRIM rule has been changed and an **APPLY** has not been used to update all members of the family or a PLOC command has been used. In the above case, TRIM57 has been moved to its pending location but that location is incorrect, according to the rule. As in other previous cases, a new pending location has been changed to the OFFSITE location and the pending date set to today's date.

## DUMP command

*Where is the Life we have lost in living? Where is the wisdom we have lost in knowledge?  
Where is the knowledge we have lost in information?*

T.S.Eliot, 'The Rock' (1934) pt. 1

— TP — DUMP —

The **DUMP** command will read the entire METATAPELIB4 database (including history information and writes the data to a series of flat files under the TAPE LIBRARIAN usercode on the database CONTROL file family.

```
TP DUMP
METATAPELIB4 dumped under (TAPELIB) on VMSA
```

The files have the general names, held under the database usercode:

```
(TAPELIBUSER)METATAPELIB/DUMP/HIST
(TAPELIBUSER)METATAPELIB/DUMP/MEDIA
(TAPELIBUSER)METATAPELIB/DUMP/RULES
```

And can be found on the database CONTROL file family. These files can be used by the Metalogic INSTALL utility to rebuild a reinitialized database.

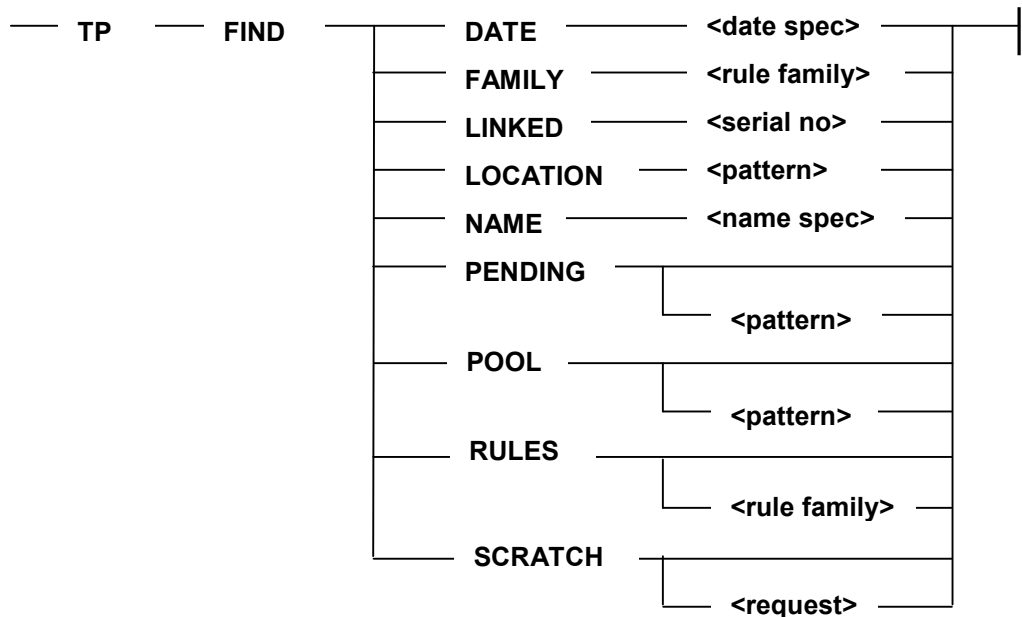
These files can be used to restore the METATAPELIB database very quickly without having to resort to a DMUTILITY dump. The Metalogic INSTALL utility can be used to automatically re-initialize and reload the database, using file generated by the DUMP command.

**U META/INSTALL NEWDMS RESTORE**

Please refer to the Metalogic INSTALL manual for more information on this feature.

## FIND command

*Knowledge is of two kinds. We know a subject ourselves, or we know where we can find information upon it.*  
 Samuel Johnson, Boswell Life



The **FIND** command allows the user to perform fast, flexible searches of the METATAPELIB database searching on various tape attributes such as NAME, LOCATION or CREATIONDATE. Some of these commands allow wild card patterns to be used to enhance these searches.

A maximum of 200 tapes will be returned by all **FIND** commands, except **RULES**, unless prefixed by **TT PRINT**, when all matching tape volumes will be recorded to a printer backup file.

### Examples:

```
TT PRINT TP FIND NAME =
TP FIND LOC STIRLING=
TP FIND DATE 12/12/93-13/03/94
TT PRINT TP FIND FAMILY DBSPACK=
```

For information on each **FIND** command, you should use the appropriate **HELP** command e.g. 'TP HELP FIND NAME', 'TP HELP FIND DATE' etc.

## Patterns and wildcard strings

Wildcard characters have been discussed earlier in **Chapter 4: TRIM rules management**, but it will be covered again here. With the **FIND** command, wildcard strings are available for identifying a tape volume, a TRIM rule identifier or a location.

The combination of these patterns uniquely identifies the rule to which a set of tape volumes can belong. Each pattern field can contain a mixture of normal alphanumeric characters or wildcard characters.

The available wildcard characters are as follows:

=	Matches any string of characters.
~	Matches any string of characters except "/".
#	Matches a single numeric character.
@	Matches a single alpha character.
&	Matches a single alpha-numeric character.
?	Matches any single character.

Note the '#' symbol often uses the alternative '£' character on many keyboards.

### Examples:

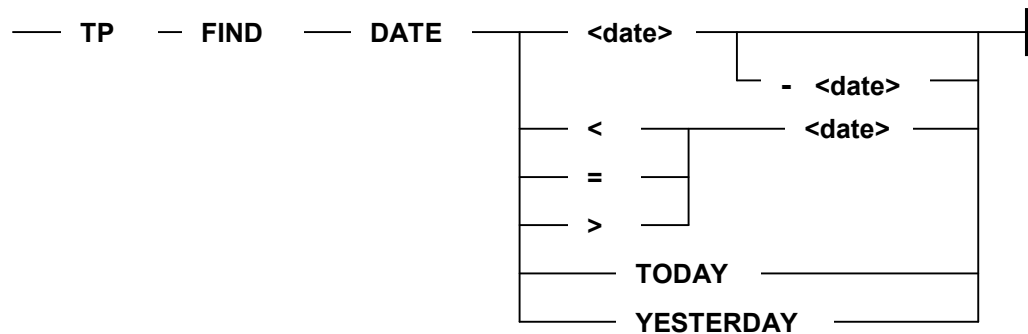
```
(META) ABC/ .  
(M=) DUMP. . /FILE. #  
SCRATCH P=  
METATAPE??/AUDIT?=-
```

## FIND DATE command

*Franklin: Have you ever thought, Headmaster, that your standards might perhaps be a little out of date?*

*Headmaster: Of course they're out of date. Standards are always out of date. That is what makes them standards.*

Alan Bennett, *Forty Years On* (1969) act 2



The **FIND DATE** command is able to perform fast searches of the METATAPELIB database for tapes that match a user provided creation date specification.

The **<date>** modifier used should be in DD/MM/YY or MM/DD/YY format, subject to the setting of the Supervisor option **USDATES**. Note that the TRIM software will accept four-digit years.

For example:

```

TP FIND DATE 12/7/95-17/7/95

----- FIND: DATE from 12/07/2002 to 17/07/2002 --- Location ----- Created ---
R70881: (META) META4242002/FILE000          META EXCEPT 16:14 17/07/2002
R70880: (META) META4242002/FILE000          META EXCEPT 10:40 17/07/2002
R70879: (META) META4242002/FILE000          META EXCEPT 10:18 17/07/2002
R70878: (META) META4242002/FILE000          META EXCEPT 09:50 17/07/2002
BOB002: (TAPELIB) METATAPELIB                1:1 TAPE RACK 08:12 17/07/2002
000047: (FLEX) DBSPACK95198A/FILE000         TAPE RACK 00:00 17/07/2002
000019: (FLEX) DBSPACK95195A/FILE000         TAPE RACK 21:41 14/07/2002
R80207: (META) META4242002/FILE000          META EXCEPT 15:45 14/07/2002
000060: (FLEX) DBSPACK95194A/FILE000         TAPE RACK 21:38 13/07/2002
000100: (FLEX) DBSPACK95194AN/FILE000        TAPE RACK 14:23 13/07/2002
000099: (FLEX) DBSPACK95194AM/FILE000        TAPE RACK 13:54 13/07/2002
000093: (FLEX) DBSPACK95193A/FILE000        TAPE RACK 21:39 12/07/2002
000087: (FLEX) NEWFLEXTAPE/FILE000          UNSPECIFIED 13:25 12/07/2002
BOB001: (TAPELIB) METATAPELIB                1:1 TAPE RACK 08:14 12/07/2002
  
```



All reports are sorted by date of creation in **descending** order, such that the most recent tapes will always be seen at the beginning of the page.

The **FIND DATE** command can use various date filters to enhance the searching, using ranges e.g. 12/1/94 - 15/1/94 or using a specified date with '>' (all tapes created **after** the target date) or '<' (all tapes created **before** the target date).

```
TP FIND DATE> 18/7/2002

----- FIND: DATE from 18/07/2002 to 21/07/2002 --- Location ----- Created ---
000013: (FLEX) DBSPACK95201A/FILE000          TAPE RACK      21:39 20/07/2002
R70883: (META) META4242002/FILE000          META EXCEPT  15:21 20/07/2002
UUAGEN: (META) METATRIP4040109/FILE000      OFFSITE       14:21 20/07/2002
R80174: (META) METATRIP4242002/FILE000      OFFSITE       14:05 20/07/2002
000011: (FLEX) DBSPACK95200A/FILE000          TAPE RACK      08:46 20/07/2002
BOB003: (TAPELIB) METATAPELIB                1:1 TAPE RACK   08:09 19/07/2002
000048: (FLEX) DBSPACK95199A/FILE000          TAPE RACK     22:59 18/07/2002
IPP001: METATAPE                            1:0 META EXCEPT 16:13 18/07/2002
R70882: (META) META4242002/FILE000          META EXCEPT  10:08 18/07/2002
```

The **TODAY** modifier has special significance in that it always returns all tapes created in the last 24 hours, regardless of when the command is issued.

The **YESTERDAY** modifier is similar but only returns volumes created yesterday i.e. from 00:00-23:59.

```
TP FIND DATE TODAY

----- FIND: DATE after 12:00:45 on 20/07/2002 --- Location ----- Created ---
000013: (FLEX) DBSPACK95201A/FILE000          TAPE RACK      21:39 20/07/2002
R70883: (META) META4242002/FILE000          META EXCEPT  15:21 20/07/2002
UUAGEN: (META) METATRIP4040109/FILE000      OFFSITE       14:21 20/07/2002
R80174: (META) METATRIP4242002/FILE000      OFFSITE       14:05 20/07/2002
```

**TP TODAY** is a convenient synonym for **TP FIND DATE TODAY**.

Response to a screen device is limited to 200 tapes, the field 'REPORT TRUNCATED' will be seen at the end of the page if this occurs. As with other **FIND** commands, the **TT PRINT** prefix, which will produce a printed report, does not exhibit this behaviour.

## Examples:

```
TP TODAY
TT PRINT TP FIND DATE 12/12/93
TP FIND DATE 1/12/93 - 15/4/94
TP FIND DATE 11/10/93
TP FIND DATE > 15/4/94
TP FIND DATE TODAY
TT PRINT TP TODAY
TP FIND DATE YESTERDAY
```

## FIND FAMILY command

*And so I give you our toast. From that young man upstairs who has had the impudence to make me a great-uncle, to Mother and Father on their Golden Wedding; through four generations of us, and to those who have gone, and those who are to come. To the family—that dear octopus from whose tentacles we never quite escape, nor, in our inmost hearts, ever quite wish to.*

Dodie Smith, Dear Octopus (1938) p. 120

— TP — FIND — FAMILY ——— <rule family> —————|

<rule family>

—————|  
 | <user pattern> | <id pattern> |—————|  
 | | / <index pattern> |

The **FIND FAMILY** command allows the retrieval of information for tapes that have a rule identifier, which matches the <rule family> specification. This is similar to <name spec>, used in the **FIND NAME** command, in that the rule owner, identifier and index fields can be used as patterns within the <wildcard family> assignment.

Response to a screen device is limited to 200 tapes, the field 'REPORT TRUNCATED' will be seen at the end of the page if this occurs. As with other **FIND** commands, the **TT PRINT** prefix, which will produce a printed report, does not exhibit this behaviour.

The sample output below shows a typical response for one of the special unlabelled tape rules (rule identifier is always prefixed by UL\_).

TP FIND FAM (META)UL_TEST								
-----	FIND: FAMILY	'(META)UL_TEST'	-----	Location	-----	Created	--	Gen
TRIM72:	(META)	-- UNLABELLED TAPE --		LOCATION 1		02/04/2002		0
TRIM71:	(META)	-- UNLABELLED TAPE --		LOCATION 2		14/03/2002		1

Note that **FIND FAMILY** will output a generation number for each tape as part of the report line. The report is grouped by the full rule identity (including rule index) and ordered by descending creation date within each subset.

Note that, unlike the FILEID pattern, a rule index can only ever be numeric. If members of all indices belong to the same rule identity, then '=' should be used for the rule index. The example below shows this sort of behaviour showing the split out between three different families TRIMTESTK/1, TRIMTESTK/2 and TRIMTESTK/3.

```
TP FIND FAM TRIM=/=

----- FIND: FAMILY 'TRIMTESTA/=' ----- Location ----- Created -- Gen
TRIM53: (TRIMTEST) TRIMTESTJ/FILE000      LINK 1          31/01/1996    0
    --- ABOVE VOLUMES LINK-ED TO: 'TRIM11' ---
TRIM52: (TRIMTEST) TRIMTESTJ/FILE000      LINK 2          21/01/1996    1
    --- ABOVE VOLUMES LINK-ED TO: 'TRIM10' ---
TRIM51: (TRIMTEST) TRIMTESTJ/FILE000      LINK 2          13/01/1996    2
    --- ABOVE VOLUMES LINK-ED TO: 'TRIM09' ---

----- FIND: FAMILY 'TRIMTESTK/=' ----- Location ----- Created -- Gen
TRIM68: (TRIMTEST) TESTKYEAR/FILE000      ONE            31/01/1996    0
TRIM60: (TRIMTEST) TESTKDAY/FILE000        ONE            30/01/1996    1
TRIM59: (TRIMTEST) TESTKDAY/FILE000        TWO            29/01/1996    2
TRIM58: (TRIMTEST) TESTKDAY/FILE000        TWO            28/01/1996    3
TRIM57: (TRIMTEST) TESTKDAY/FILE000        TWO            27/01/1996    4
TRIM56: (TRIMTEST) TESTKDAY/FILE000        TWO            26/01/1996    5
TRIM55: (TRIMTEST) TESTKDAY/FILE000        TWO            25/01/1996    6
TRIM69: (TRIMTEST) TESTKYEAR/FILE000        TWO            03/01/1996    7
TRIM67: (TRIMTEST) TESTKYEAR/FILE000        TWO            03/01/1996    7
TRIM64: (TRIMTEST) TESTKMONTH/FILE000      TWO            02/12/2002    8
TRIM63: (TRIMTEST) TESTKMONTH/FILE000      TWO            02/11/2002    9
TRIM62: (TRIMTEST) TESTKMONTH/FILE000      TWO            31/01/2002   10
```

Note that because of the special usage of "/" as part of the pattern, TRIM merges all members of each rule family in the TRIMTESTK set and presents them as though all the tapes belong to a specific rule.

Where members of a rule family are "linked" to tapes from a master rule, the **FIND FAMILY** command display this linkage information as part of the response. In the above screen, the tapes TRIM53, TRIM52 and TRIM51 from rule family TRIMTESTK/= are linked to tapes TRIM11, TRIM10 and TRIM09 respectively. Similarly, the **FIND LINKED TRIM94** command would also show which tapes are currently linked to a master, which may include members from more than one family.

#### Examples:

```
TP FIND FAMILY (META) RULE/=
TP FIND FAMILY RULE/1
TT PRINT TP FIND FAMILY METATEST
TP FIND FAMILY =MYRULE
```

## FIND LINKED command

*It is the secret sympathy,  
The silver link, the silken tie,  
Which heart to heart, and mind to mind,  
In body and in soul can bind.*

Sir Walter Scott, The Lay of the Last Minstrel (1805)

— TP — FIND — LINKED ——— <serial no> —————|

The **FIND LINKED** command allows the user to determine which tapes are currently linked, via TRIM rules, to an individual tape designated by the <serialno> modifier. Note that more than one rule family may be attached to a master link rule. The tapes will be returned in random order and generation information is not given. Note that the **FIND FAMILY** command will also give linkage information if the rule id provided is a linked rule.

```
TP FIND LINKED TRIM11

----- FIND: TAPES LINK-ED TO 'TRIM11'-- Location ----- Created ----- Expiry
TRIM49:TRIMTESTI/FILE000          ONE          14:42 28/01/1996   UNKNOWN
TRIM53:TRIMTESTJ/FILE000          LINK 1       14:43 31/01/1996   UNKNOWN
TRIM53:TRIMTESTJ/FILE000          LINK 1       14:43 31/01/1996   UNKNOWN
```

The list of tapes generated by this report is returned in descending chronological order of creation.

Response to a screen device is limited to 200 tapes, the field 'REPORT TRUNCATED' will be seen at the end of the page if this occurs. As with other **TP FIND** commands, the **TT PRINT** prefix, which will produce a printed report, does not exhibit this behaviour.

### Examples:

```
TP FIND LINKED A00012
TT PRINT TP FIND LINKED 81
```

## FIND LOCATION command

*Any old place I can hang my hat is home sweet home to me.*

William Jerome 1865–1932, title of song (1901)

— TP — FIND — LOCATION ——— <location pattern> ——— |

The **FIND LOCATION** command allows searching of the database for a specific or wildcard location string. The <location pattern> field must not be greater than 12 characters and should only consist of alphanumeric characters. Note that unless a trailing '=' has been provided, all spaces used in the pattern will be respected.

Response to a screen device is limited to 200 tapes, the field 'REPORT TRUNCATED' will be seen at the end of the page if this occurs. As with other **FIND** commands, the **TT PRINT** prefix, which will produce a printed report, does not exhibit this behaviour.

Similar to the **NAME** variant of **FIND**, search performance will always be improved if a location pattern is a string that has leading non-wildcard characters.

For example, to search for locations which are similar to SCRATCH POOL,

**TP FIND LOC SCR=**

Is significantly faster than

**TP FIND LOC =ATCH P=**

This is because of the way that TRIM searches the METATAPELIB key data in the relevant DMSII set; DMSII set searches are significantly faster if the leading part of the key is known.

An example command:

```
TP FIND LOC L=

----- FIND: LOCATION 'LOC 2' ----- Created ----- Expiry -
TRIM93: (TAPELIB)DBDUMP/FILE000          15:47 31/10/1994 UNKNOWN

----- FIND: LOCATION 'LOC 3' ----- Created ----- Expiry -
TRIM89: (TAPELIB)DBDUMP/FILE000          12:14 06/10/1994 PERMANENT
TRIM90: (TAPELIB)LINK/AUDIT              12:33 06/10/1994 UNKNOWN
TRIM92: (TAPELIB)LINK/AUDIT              12:37 06/10/1994 UNKNOWN

----- FIND: LOCATION 'LOCATION 1' ----- Created ----- Expiry -
TRIM72: (META) -- UNLABELLED TAPE --      12:51 02/04/2002 UNKNOWN

----- FIND: LOCATION 'LOCATION 2' ----- Created ----- Expiry -
TRIM71: (META) -- UNLABELLED TAPE --      12:51 14/03/2002 UNKNOWN

----- FIND: LOCATION 'LOC2' ----- Created ----- Expiry -
TRIMQ2: (TRIMTEST)TRIMTESTL/FILE000      09:45 03/05/2002 UNKNOWN
TRIMQ3: (TRIMTEST)TRIMTESTM/FILE000      09:46 03/05/2002 UNKNOWN
TRIM82: (FLEX)LINKINX0/FILE000           12:51 23/05/1993 UNKNOWN
TRIM83: (FLEX)LINKINX1/FILE000           12:51 23/05/1993 UNKNOWN
```

Examples:

```
TP FIND LOC SCRATCH POOL
TP FIND LOC =
TT PRINT TP FIND LOC STIRLING
TT PRINT TP FIND LOC SCR= P=
```

*You mentioned your name as if I should recognise it, but I assure you that, beyond the obvious facts that you are a bachelor, a solicitor, a Freemason, and an asthmatic, I know nothing whatever about you.*

Sir Arthur Conan Doyle, *Return of Sherlock Holmes* (1905)

&lt;name pattern&gt;

A complex example is shown below; note that using (=) as the usercode or owner pattern will also return non-usercoded (\*) tapes.

TP	FIND NAME (=)T=/FILE###	Location	Created	Expiry
-----	FIND: User ' ' -----	-----	Created -----	Expiry -
TEST4	:T/FILE000	OFFICE	15:39 21/04/1994	UNKNOWN
TAPE13	:TAPEMANAGER/FILE000		00:00 01/01/1994	UNKNOWN
TAPE14	:TAPEMANAGER/FILE000		00:00 01/01/1994	UNKNOWN
-----	FIND: User 'FLEX' -----	Location -----	Created -----	Expiry -
R70287	:TEST/FILE000	OFFICE	09:39 13/05/1993	UNKNOWN
R70286	:TEST/FILE000	OFFICE	09:33 13/05/1993	UNKNOWN
R70284	:TEST/FILE000	OFFICE	14:53 12/05/1993	UNKNOWN
-----	FIND: User 'META' -----	Location -----	Created -----	Expiry -
TAPE02	:TAPEMANAGER/FILE000	SCRATCH POOL	15:52 18/11/1994	UNKNOWN
TRIMX3	:TEST/FILE123	OFFSITE	09:08 03/05/2002	UNKNOWN
TRIMX2	:TEST/FILE000	OFFSITE	09:06 03/05/2002	UNKNOWN
R70527	:TEST/FILE000	OFFICE	14:58 21/06/1994	UNKNOWN
IPP021	:TEST/FILE000	OFFICE	15:05 20/05/1994	UNKNOWN
IPP022	:TEST/FILE000	OFFICE	15:27 18/05/1994	UNKNOWN
-----	FIND: User 'TRIMTEST' -----	Location -----	Created -----	Expiry -
TRIM60	:TESTKDAY/FILE000	ONE	12:50 02/04/2002	UNKNOWN
TRIM59	:TESTKDAY/FILE000	TWO	12:50 01/04/2002	UNKNOWN
TRIM58	:TESTKDAY/FILE000	TWO	12:50 31/03/2002	UNKNOWN



Response to a screen device is limited to 200 tapes, the field 'REPORT TRUNCATED' will be seen at the end of the page if this occurs. As with other **FIND** commands, the **TT PRINT** prefix, which will produce a printed report, does not exhibit this behaviour.

Search performance will always be improved if the wildcard specifications include patterns, which start with non-wildcard characters.

For example, **(ME=)ABC/=** is a significantly faster search than **(=ETA)=BC/=** because of the way that TRIM searches the METATAPELIB key data in the relevant DMSII set.

Tapes will be returned, grouped by usercode if appropriate, and sorted by name within each usercode. Where tape names match, entries will be sorted in descending chronological order of creation date and time.

Note also, in the following example, how ALL tapes regardless of their owner will be returned if no usercode pattern has been provided.

```
TP FIND NAME METATAP=

----- FIND: User ' ' ----- Location ----- Created ----- Expiry -
IPP001:METATAPE                      1:0 META EXCEPT 16:13 18/07/2002 UNKNOWN

----- FIND: User 'TAPELIB' ----- Location ----- Created ----- Expiry -
BOB003:METATAPELIB                   1:1 TAPE RACK      08:09 19/07/2002 16/08/2002
BOB002:METATAPELIB                   1:1 TAPE RACK      08:12 17/07/2002 14/08/2002
BOB004:METATAPELIB                   1:1 TAPE RACK      08:10 10/07/2002 07/08/2002
038118:METATAPELIB/AUDIT2            OFFICE            09:35 22/04/1994 UNKNOWN
BOB008:METATAPELIB/AUDIT2            OFFICE            10:30 16/10/1993 UNKNOWN

----- FIND: User 'UNKNOWN' ----- Location ----- Created ----- Expiry -
BOB001:METATAPELIB                   1:1 UNSPECIFIED 08:14 12/07/2002 UNKNOWN
```

In the above example, note the BOB001 tape that is owned by the usercode UNKNOWN. Usually, this will be seen very rarely and occurs after the LOGREADER process has been run and a 'missed' tape has been detected from the SUMLOG.

For this to happen, two events must have occurred. First, SUPERVISOR must have been inactive for a period of time whilst the tape was created and secondly, LOGREADER was unable to determine the usercode of the creating task from the SUMLOG because the appropriate BOJ or EOJ records could not be found.

In the event that such UNKNOWN usercode records are found, changing the owning usercode with the **TP OWNER** modifier followed by a **TP SR** and lastly, a **TP APPLY** will rectify the problem.

**TP SER** is a convenient synonym for **TP FIND NAME**.

**Examples:**

```
TT PRINT TP FIND NAME =  
TT PRINT TP SER =  
TP FIND NAME (META) DBSPACK=  
TP SER (META) DBSPACK=  
TP FIND NAME (ME=) DBS??=/FILE
```

## FIND PENDING command

*I never think of the future. It comes soon enough.*

Albert Einstein ,in an interview, given  
on the Belgenland, December 1930

— TP — FIND — PENDING

<location pattern>

The **FIND PENDING** command allows searching of the database for all tapes which have a PENDINGLOCATION attribute matching the optional pattern and a PENDINGDATE attribute less than or equal to today's date.

The <location pattern> cannot exceed 12 characters and spaces will be respected in the pattern.

Further, if no pattern is provided then all tapes with any pending location and a valid pending date will be marked with the REPORTED attribute set to TRUE. This latter case will generate the same tapes provided by the **TPDB\_DAILYREPORTS** ODTSEQUENCE.

```
TP FIND PENDING

----- PENDING MOVE: 'STIRLING' to 'LOC2' ----- Created ----- Pending
TRIM84: (FLEX) LINKINX1/FILE000 12:51 24/05/2002 TODAY
TRIM83: (FLEX) LINKINX1/FILE000 12:50 23/05/2002 TODAY

----- PENDING MOVE: 'VAULT' to 'LOC2' ----- Created ----- Pending
TRIM84: (META) DISKDUMP1/FILE000 12:51 24/05/2002 TODAY
TRIM83: (META) DISKDUMP2/FILE000 12:50 23/05/2002 TODAY

----- PENDING MOVE: 'LOC2' to 'SCRATCH POOL' ----- Created ----- Pending
TRIM84: (FLEX) LINKINX1/FILE000 12:51 06/12/1994 TODAY
TRIM83: (FLEX) LINKINX1/FILE000 12:50 05/12/1994 TODAY
```

The main report is broken down into subgroups governed by the current values for pending location and current location. The main key is pending location that is sorted in alphabetical order.

By using the **<wildcard location>** field, the report may be filtered:

```
TP FIND PENDING SCR= POOL

----- PENDING MOVE: 'LOC2          ' to 'SCRATCH POOL' ----- Created ----- Pending
TRIM84: (FLEX) LINKINX1/FILE000                12:51 06/12/1994  TODAY
TRIM83: (FLEX) LINKINX1/FILE000                12:50 05/12/1994  TODAY
...
```

Response to a screen device is limited to 200 tapes, the field 'REPORT TRUNCATED' will be seen at the end of the page if this occurs. As with other FIND commands, the **TT PRINT** prefix, which will produce a printed report, does not exhibit this behaviour.

#### Examples:

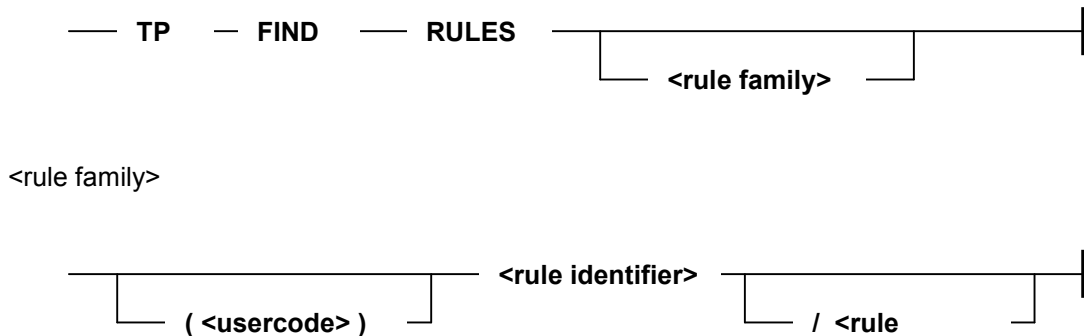
```
TT PRINT TP FIND PENDING =
TP FIND PENDING =LOC=
```



## FIND RULES command

*There are two golden rules for an orchestra: start together and finish together. The public doesn't give a damn what goes on in between.*

Sir Thomas Beecham



The **FIND RULES** command allows on-line interrogation of all or an individual TRIM rules, currently held in the METATAPELIB database. For an “all rules” report, the rules are grouped by owning usercode and the output details usercode, volume and file identifier patterns plus some brief retention information.

TP FIND RULES				
----- FIND: RULES for * ----- Patterns -----				Info -
BOB	TAPELIB	METATAPELIB		28 Days
BOB2	AA	AA	AA	--
DBDUMP	META	DBDUMP	=	5 Days
DBDUMPLINK	META	DBAUDIT#	=	--
DISKDUMP	FLEX	DISK=	FILE=	--
DUMMY	TRIMTEST	DUMMY	FILE=	--
EXCEPTION				--
EX1	BADUSER	AAAA@	BBBB#	30 Days
EX2	=	AAAAA	BBBBB	30 Days
FAMILYMANAGER	=	DBSPACK=	FILE=	--
IANSRULE/1	META	IPP=	FILE=	1 Gen
IANSRULE/2	META	BOBS=	FILE=	90 Days
LINKINX	FLEX	LINKINX0	FILE=	--
LINKINX/1	FLEX	LINKINX1	FILE=	--
METATAPES	FLEX	META	FILE=	--
METATRIP	=	METATRIP=	FILE=	--
NEWRULE	META	METATEST	=	12 Days
OLDMETATAPES	=	META	FILE=	--
PACKDUMP	FLEX	PACK	FILE=	--
TEST	TAPELIB	DBDUMP	FILE=	3 Gen
TESTLINK	TAPELIB	LINK	AUDIT	--

If an individual rule is selected, the <rule identity> must consist of the complete rule name including usercode and index if required. As of this time, no wildcard patterns are permitted. TRIM will return full information about the rule including location details.

```
TP FIND RULE DBDUMP

----- FIND: RULE info for 'DBDUMP' -----

-- TAPE ATTRIBUTES -- PATTERN ----- EXAMPLE --
Usercode info          META              META
Volume id info         DBDUMP.           DBDUMP12
File id info           =                 FILE000

-- RETENTION & LOCATION INFO --
Tape EXPIRES after     4 GENERATIONS or 5 DAYS
Location movement     'MYLOC1           ' for 1 GEN
                      'MYLOC2           ' for 1 GEN
                      'MYLOC3           ' for 2 GENS
```

For rules which have a linkage to a 'master' rule, the output is slightly different, showing information about the linkage as well as location and expiry data.

```
TP FIND RULE DBDUMPLINK

----- FIND: RULE info for 'DBDUMPLINK' -----

-- TAPE ATTRIBUTES -- PATTERN ----- EXAMPLE --
Usercode info          META              META
Volume id info         DBAUDIT#          DBAUDIT1
File id info           =                 FILE000

-- RETENTION & LOCATION INFO --
Tape EXPIRES after     >> NOT SPECIFIED <<
Location movement     'AUDLOC1           ' for 1 GEN
                      'AUDLOC2           ' for 1 GEN
                      'AUDLOC3           ' for 2 GENS

Rule is LINKED to      : DBDUMP
```

A limit of 200 rules will be returned in the “all rules” report but prefixing the command with a **TT PRINT** will bypass this restriction.

#### Examples:

```
TT PRINT TP FIND RULES
TP FIND RULE (META)DBDUMP/1
TP FIND RULE EXCEPTION
```

## FIND SCRATCH command

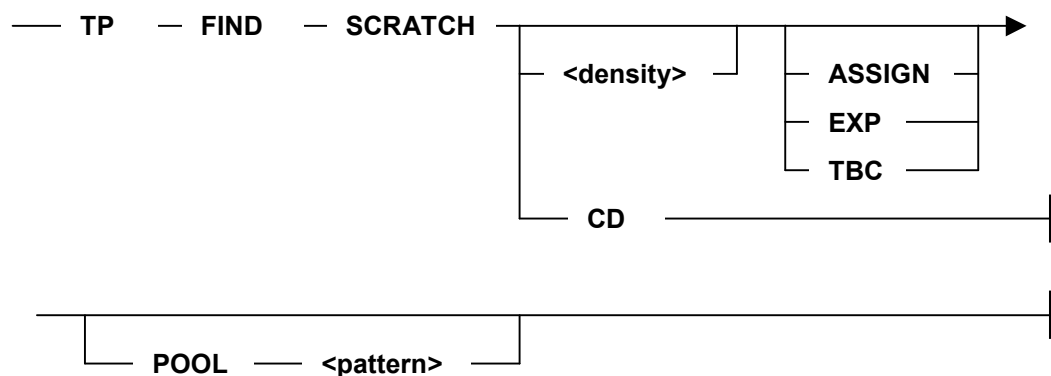
*Lear: Dost thou know me, fellow?*

*Kent: No, sir; but you have that in your countenance which I would fain call master.*

*Lear: What's that?*

*Kent: Authority.*

William Shakespeare, King Lear



The **FIND SCRATCH** command will generate a list of all purge authorised (i.e. PGOK attribute is TRUE) and scratch tapes, currently known to the METATAPELIB database. This report, if generated without a **<density>** modifier, will be grouped according to density and by ascending serial number within each subgroup.

The user may optionally provide his own specifications to filter the subset of tapes returned. To see all scratch or PGOK tapes:

```

TP FIND SCR

----- FIND: PGOK (1250) ----- Location ----- Created ----- Expired
IFN470: S C R A T C H                SCRATCH POOL 11:39 16/06/2002
000036:TESTBOB/FILE000                SCRATCH POOL 09:24 24/03/2002 22/06/2002
000055:DBSPACK95090A/FILE000          SCRATCH POOL 08:15 03/04/2002 17/05/2002
000085:DBSPACK95166A/FILE000          SCRATCH POOL 21:36 15/06/2002 13/07/2002
000096: S C R A T C H                SCRATCH POOL 08:48 13/07/2002

----- FIND: PGOK (FMT36TRK) ----- Location ----- Created ----- Expired
BLLOG7:CMIGTAPE/FILE000                SCRATCH POOL 00:00 20/09/1993 19/12/1993
BRISTB:METAOECD/FILE000                SCRATCH POOL 00:00 09/02/1985 10/05/1985
IFN471:RMVTAPE/FILE000                META EXCEPT 12:02 14/06/2002 12/09/2002

----- FIND: PGOK (FMT36TRK) ----- Location ----- Created ----- Expired
METAL :BADENTERSCHED/FILE000            SCRATCH POOL 00:00 22/11/1993 20/02/1994
META31:METAPRIVLIB/FILE000            SCRATCH POOL 00:00 18/02/1984 18/05/1984
  
```



Response to a screen device is limited to 200 tapes, the field 'REPORT TRUNCATED' will be seen at the end of the page if this occurs. As with other **FIND** commands, the **TT PRINT** prefix, which will produce a printed report, does not exhibit this behaviour.

## Density modifier

The **<density>** modifier, if specified, should be a valid Unisys density valid in the loaded MCP. TRIM will permit densities to be used without the 'BPT' or 'FMT' prefix but the full density name is also permitted. For example, these densities are valid:

**800,1600,6250,BPI38000,BPI11000, DLT36, 36TRK, FMTDLT10**

Using a density specification greatly increases the speed of the search.

```
TP FIND SCR DLT35

----- FIND: PGOK (1250) ----- Location ----- Created ----- Expired
000036:TESTBOB/FILE000          SCRATCH POOL 09:24 24/03/2002 22/06/2002
000042:WORK95178AF/FILE000      UNSPECIFIED 12:38 27/06/2002 UNKNOWN
000055:DBSPACK95090A/FILE000    SCRATCH POOL 08:15 03/04/2002 17/05/2002
000064:DBSPACK95167AN/FILE000   SCRATCH POOL 10:19 16/06/2002 13/07/2002
000041: S C R A T C H          SCRATCH POOL 08:48 17/07/2002
000006: S C R A T C H          SCRATCH POOL 15:55 14/07/2002
```

## CD modifier

The CD modifier allows PGOK or SCRATCH CDs to be reported. These CD volumes are only generated by the Metalogic FLEX software, indicating the availability of a CD volume for re-use.

```
TP SCR CD

----- FIND: PGOK (CD) -----
CD0003:CD00003F2000166B          SCRATCH POOL 00:00 14/06/2000 02/04/2003
CD0013:CD00013F2000168A          SCRATCH POOL 00:00 16/06/2000 24/04/2003
CD0037:CD00037F2000185A          SCRATCH POOL 00:00 03/07/2000 23/04/2003
CD0043:CD00043F2000195A          SCRATCH POOL 00:00 13/07/2000 23/04/2003
CD0044:CD00044F2000196A          SCRATCH POOL 00:00 14/07/2000 02/04/2003
CD0092:CD00092F2000273A          SCRATCH POOL 00:00 29/09/2000 24/04/2003
....
```

## EXP qualifier

The **EXP** qualifier will only report those tapes that are scratch tapes (i.e. appears as S C R A T C H) or marked as PGOK and have expired (i.e. the EXPIRY attribute is non-zero and less than or equal to today's date).

```
TP FIND SCR 1250 EXP 995

----- FIND: PGOK (1250) EXPIRED ----- Location ----- Created ----- Expired
IFN470: S C R A T C H                      SCRATCH POOL 11:39 16/06/2002
000036:TESTBOB/FILE000                     SCRATCH POOL 09:24 24/03/2002 22/06/2002
000055:DBSPACK95090A/FILE000               SCRATCH POOL 08:15 03/04/2002 17/05/2002
000064:DBSPACK95167AN/FILE000              SCRATCH POOL 10:19 16/06/2002 13/07/2002
000085:DBSPACK95166A/FILE000              SCRATCH POOL 21:36 15/06/2002 13/07/2002
000096: S C R A T C H                      SCRATCH POOL 08:48 13/07/2002
000041: S C R A T C H                      SCRATCH POOL 08:48 17/07/2002
000006: S C R A T C H                      SCRATCH POOL 15:55 14/07/2002
```

## POOL modifier

The POOL modifier enhances the search for SCRATCH or PGOK tape volumes that have a SCRATCHPOOL matching the pattern provided. The POOL modifier can **only** appear at the end of the command. ).

```
TP SCR DLT35 POOL META

----- FIND: PGOK (FMTDLT35) META ----- Location ----- Created ----- Expired
----- SCRATCH POOL: META -----
DDL001:TAPESERVER01/FILE000                META EXCEPT 00:00 16/07/1999 UNKNOWN
DDL002:TAPESERVER02/FILE000                META EXCEPT 00:00 16/07/1999 UNKNOWN
----- 2 ENTRIES FOUND -----
```

## TBC qualifier

The **TBC** (To-Be-Cleaned) modifier will only report those tapes that are considered to be due for cleaning. TRIM determines this by examining the ACCESSCOUNT attribute, and if that exceeds a site-controlled maximum value, the tape will appear in this report. ACCESSCOUNT returns the number of times a tape has been created or used.

A special MAGUS configuration variable, TL\_MAXUSES, can be easily changed to change the limit for this count. By default, TL\_MAXUSES is set to 10 but its value may be changed using the Metalogic INSTALL utility from CANDE:

#### U META/INSTALL CONFIG

The configuration setting is available from the TRIM menu option.

The value of the ACCESSCOUNT attribute is always set to zero after the tape has been marked as "cleaned" by use of the **TP CLEANED** command.

TP SCR is a synonym for	TP FIND SCRATCH
TP TBC is a synonym for	TP FIND SCRATCH TBC
TP EXP is a synonym for	TP FIND SCRATCH EXP

For convenience, note that both **TP TBC** and **TP EXP** will also take a **<density>** modifier following the command.

#### Examples:

```
TP FIND SCR 1250
TP FIND SCR 38000 EXP
TP SCR
TP EXP 6250
TT PRINT TP TBC
TT PRINT TP FIND SCR 1250 EXP
```



Note that the keyword modifier may be any **TP** command including serialnumber commands such as **PLOC**, **PGOK** etc.

Most **HELP** responses are multiple pages, so a **TT NS** will usually be seen in the top left hand corner of the screen. In order to show the original command, this has been omitted from the following screen.

```
TT TP HELP FIND LOC

*** METATAPELIB : Help for FIND LOCATION ***

----- TP -- FIND ----- LOCATION --- <pattern> -----!

The TP FIND LOCATION commands allows searching of the database for a
specific or wildcarded LOCATION. The <pattern> must not be greater
than 12 characters. Note that unless a trailing '=' is used, all
spaces used in a <pattern> will be respected. Please use 'TP HELP
PATTERN' for more information.

Examples:      TP FIND LOC =
               TT PRINT HELP LOC STIRLING
               TT PRINT HELP LOC STIR=

A maximum of 200 tapes will be returned in the report, please use the
'TT PRINT' prefix to generate a full printed list. The report will
be sorted by LOCATION order and by SERIALNO within each LOCATION.
```

Examples:

```
TP HELP FIND NAME
TP HELP SCRATCH
TP HELP PLOC
```

## LC command

*Comment voulez-vous gouverner un pays qui a deux cent quarante-six variétés de fromage?*

Charles de Gaulle

---

— TP — LC — <text> —

The **LC** (Log Comment) command causes the supplied **<text>** string to be logged directly into the current TAPELOG file. This is provided for site usage in a similar way to the implementation of the Unisys LC ODT command. Text may not exceed 200 characters. The message will appear as an ODT command and therefore is visible via using a **TP LOG ODT** command:

**11:15:56 ODT: LC THIS IS A TEST MESSAGE**

The special modifier, **CLOSE**, will automatically close the current TAPELOG and open a new file. The previous TAPELOG has the date and time appended to its original title; an example is shown below where the date is in military format (YYMMDD) and the time is HHMMSS:

**(TAPELIB)METATAPELIB/TAPELOG/950726/000101**

It is a good practise to release the TAPELOG once a week or more. Failure to do this on a regular basis will mean that the log file will continue to grow larger. If possible, release and backup of these old TAPELOG files should be performed in a similar way to other system log files.

If the full SUPERVISOR package is available, LC can be scheduled to execute automatically (using the **AFTER** command) as shown below.

### Examples:

```
TP LC <text>
TP LC CLOSE
TT AF +0400 ON SUNDAY:TT TP LC CLOSE
```

## LOG command

*I always say, keep a diary and some day it'll keep you.*

Mae West, Every Day's a Holiday (1937 film)

TP	LOG
ALL	
CON	
CTL	
ERR	
EVT	
LC	
LGR	
MSG	
ODT	
RMT	

The **LOG** command allows read only access to the METATAPELIB TAPELOG file returning relevant tape library activity from the last 7 days, or a maximum of 300 lines of text. If you wish to view beyond this limit, you should use CANDE or MARC text editor facilities accessing the TAPELOG as a normal data file.

**LOG** takes various modifiers that can filter the information returned to the caller. These modifiers are described below:

ALL	Returns all entries in the log to current limits
CON	Returns all error and warning messages generated by the TP CONFIRM command
CTL	Returns all DSI or StorageTek™-related TRIM messages
ERR	Returns all error messages due to DMSII or generated programmatically.
EVT	Returns all event information e.g. tape creations, tape volume purges
LC	Returns all tape log entries generated by the TP LC command
LGR	Returns all messages and event notices detected by the LOGREADER utility
MSG	Returns all solicited and unsolicited TRIM messages
ODT	Returns all TP commands input via COMS window or ODT
RMT	Returns all events, messages and commands processed from TRIM slave system (only seen on MASTER hosts)

It should be noted that all commands, non-interrogation output and information messages are logged to the TAPELOG; the only exception is the logging of individual **LOG** commands. The output will be returned in descending time order.

Many of the messages written to the TAPELOG have been simplified and made consistent, each logged entry has a valid context prefix, like '**Msg:**', '**Evt:**' etc. It is likely that this command will be considerably enhanced to take for example, date and time search parameters in future releases.

The text of each generated message is usually self-explanatory. Some examples are shown below:

```
TP LOG EVT

----- 26/07/95 LOG -----
08:41:44 Evt: CREATED [TESTXX] STATSTAPE by
08:39:32 Evt: SCRATCH-ed [TESTXX] SCRATCH tape
08:08:06 Evt: CREATED [BOB001] METATAPELIB by TAPELIB(not SCR)
----- 25/07/95 LOG -----
23:28:59 Evt: CREATED [000085] DBSPACK95206A by FLEX
17:23:23 Evt: SCRATCH-ed [000085] DBSPACK95166A
----- END OF LOG SNAPSHOT -----
```

```
TP LOG MSG

----- 26/07/95 LOG -----
08:39:30 Msg: ALLOW PG OF TESTXX (Program CLOSE PURGE)
08:07:48 Msg: Usage [BOB001] METATAPELIB by TAPELIB
08:07:11 Msg: Usage [BOB001] METATAPELIB by TAPELIB
----- 25/07/95 LOG -----
23:31:23 Msg: Volumes confirmed 4 (warnings 0),Links 0,Errors 0
17:23:21 Msg: ALLOW PG OF 000085
16:12:44 Msg: VOL: 000001 GENERATION @4 not found in METATAPELIB
----- END OF LOG SNAPSHOT -----
```

#### Examples:

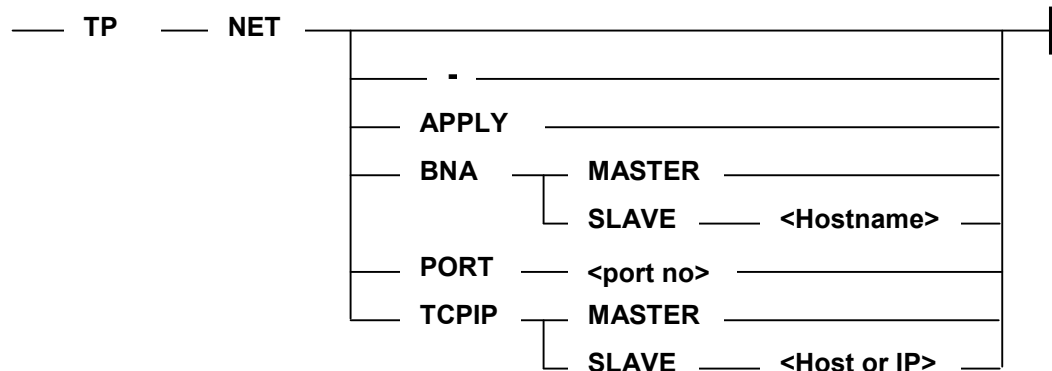
```
TP LOG
TP LOG CON
TP LOG MSG
TT PRINT TP LOG EVT
```



## NET command

*O what a tangled web we weave, When first we practise to deceive!*

'Marmion' (1808) canto 6, st. 17



The TP **NET** configuration command has been implemented to assist with network setup in a multi-host, master-slave environment. The user can readily configure a system to use BNA or TCPIP and run as a master or slave.

The BNA and TCPIP variants dictate the network protocol to be used by TRIM for inter-system communication. Both BNA and TCPIP *\*require\** either MASTER or SLAVE to indicate if this system is to be a TRIM master or slave.

If SLAVE is specified, a valid hostname or IP address (TCPIP only) must be provided.

If TCPIP is used then an optional PORT number can be assigned otherwise the default 44444 will be used.

Any changes to the network environment effected by the NET command will be marked as pending until a TP NET APPLY command is issued. If a change from master to slave is made, the METATAPELIB4 database will be immediately closed and any current network connections will be terminated.

The PORT modifier is only valid if the TCPIP protocol is in use or a network change to TCPIP is pending. PORT allows the optional change of the TCPIP port number to a value other than 44444.

Any pending actions can be cancelled using the '-' modifier:

**TP NET -**

Both the TP VER and TP NET command will show both the current network configuration and any pending changes. For example:

**TP NET BNA MASTER**

**TRIM Networking Configuration:**

**MASTER system**

**Using native TCPIP as network provider (port #55555)**

**No slave hosts are currently connected**

**METATAPELIB database is ON-LINE**

**Network Config changes are pending:**

**Network protocol will use BNAv2**

**System will be configured as >> MASTER <<**

Changing network configuration is very easy to do with the TP NET APPLY command; TRIM handles the close and restart of the TRIM subsystem without operator intervention.

Please note that switching or migrating a system from a slave to a master environment also requires that a valid METATAPELIB4 database configuration exists on the 'new' master. If appropriate, this means that the INSTALL utility must have been run to set up the TRIM environment and the database loaded from the previous master system.

Please refer to the Metalogic INSTALL manual for more details on setting up the TRIM run-time environment.

**Other examples:**

**TP NET BNA MASTER**

**TP NET TCPIP SLAVE POWEREDGE PORT 33333**

**TP NET TCPIP PORT 55555**

**TP NET APPLY**

## OPEN command

*Roderick Spode? Big chap with a small moustache and the sort of eye that can open an oyster at sixty paces?*  
P. G. Wodehouse, The Code of the Woosters

— **TP** — **OPEN** —

The **OPEN** command is responsible for opening the METATAPELIB database after a **TP CLOSE** has been previously issued. When the TRIM system is invoked after a halt-load or a Supervisor restart, this is not normally necessary because the database will be opened automatically.

An entry will also be written into the TAPELOG to signify that the database has been successfully opened:

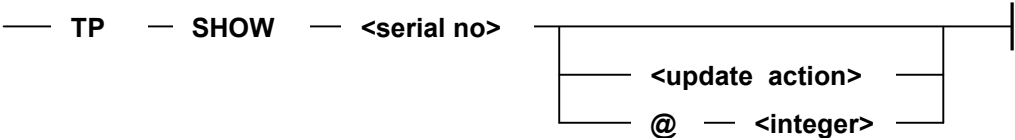
```
TP OPEN
METATAPELIB database is now available
```

The current state of the METATAPELIB database can always be easily determined by using the **TP VERSION** command.

# SHOW command

*There's no business like show business.*

Title of song, Irving Berlin (1946)



The **SHOW** command is identical to a simple TP **<serialno>** interrogation command, allowing generation selection, plus all other modification commands.

**SHOW** may be regarded as noise, allowing access to tapes that have names matching TP **<serialno>** modifiers.

## Examples:

**TP SHOW APPLY PGOK**

The above would set the PGOK flag for a tape called 'APPLY '.

## VERSION command

*It was Cohan who first said to a newspaperman (who wanted some information about Broadway Jones in 1912), 'I don't care what you say about me, as long as you say something about me, and as long as you spell my name right.'*

John McCabe, 'George M. Cohan'

— TP ——— VERSION —————

The **VERSION** command returns a variety of information about current software versions, plus networking and run-time options. The command is permitted from any station without restriction.

```
TP VER

Metalogic OPALTAPELIB library 50.500.003
  Compiled 12:30:12 on 09/07/2004
Metalogic TAPELIBUPDATER module 50.500.005
  Compiled 13:05:06 on 06/07/2004

TRIM Networking Configuration:
  MASTER system
  Using native TCPIP as network provider (port #44444)
  Connected slave host:
    #2 METAMCPA (10.0.0.20, SEG=1456)
  METATAPELIB database is ON-LINE

Runtime options:
  USDATES      : RESET
  CATALOGING   : SET
  LIBRARIAN    : TAPELIB
  Visibility   : ALL      Access : MODIFY
TAPELOG title is '(TAPELIB)METATAPELIB/TAPELOG ON DEV'
```

Version information about the TAPELIBUPDATER process and OPALTAPELIB library is shown, along with run-time options and the name of the TAPELOG.

In the above response, the TRIM networking environment has been configured as a TCPIP MASTER system (via the **TP NET** command). A slave system, METAMCPA with an IP address of 10.0.0.20, is currently linked to this master passing events commands and responses between the two hosts.

A similar command issued from the slave METAMCPA system might show:

```
TPVER

Metalogic OPALTAPELIB library 50.500.003
  Compiled 12:58:09 on 06/07/2004
Metalogic TAPELIBUPDATER module 50.500.005
  Compiled 13:05:06 on 06/07/2004

TRIM Networking Configuration:
  SLAVE system
  Using native TCPIP as network provider (port #44444)
  MASTER system assigned to POWEREDGE

TAPELOG title is '(TAPEDB)METATAPELIB/TAPELOG ON WORK'
```

Please note that the versions and compile date and time information are of importance when reporting a problem to Metalogic since these items can be used to determine whether the fault has already been fixed or not.

## Run-time Options

The **USDATES** option, as previously covered, is set using the Supervisor **TT SO** command and controls the presentation of all dates in either DD/MM/YY (option is RESET) or MM/DD/YY (option is SET). The CATALOGING option indicates that the MCP system CATALOGING (option 23 on MCP 4.1) has been set on this system.

If CATALOGING is set, and the Metalogic Full FLEX package is available, various extra features are available to integrate the FLEX and TRIM systems and have already been covered elsewhere.

Lastly, the TAPE LIBRARIAN usercode is here set to TAPELIB. This usercode has overall control of all TRIM rules and can delete tape volumes from the METATAPELIB database. The Supervisor **TT USE** command must be used to change its current setting.

The name of the TAPELOG is also shown on this screen. Although the **TP LOG** command can retrieve information from the last 48 hours of the TAPELOG file, it is sometimes useful to scan the file using CANDE editing commands. The response to the **TP VERSION** command offers a reminder of where to find this file; note that it is established through the use of the TL\_TAPELOG configuration variable, maintained by the INSTALL utility.

## Chapter 6: Getting started with OPAL

*Don't panic.*

Douglas Adams, Hitch Hiker's Guide to the Galaxy

The Supervisor **DEFINE** command is used to maintain the library of OPAL programs within Supervisor's \*SCHEDULE file, to compile new programs and to query programs stored in the program directories that Supervisor maintains. All OPAL programs must be given a unique name (identifier) when they are defined. Each name (identifier) consists of 17 characters or less in length and may contain the underscore "\_" character. Let us take the opportunity to briefly summarise the various features of OPAL programming.

Together, SUPERVISOR and OPAL combine to form a powerful knowledge-based, expert system. OPAL works with three types of algorithms – Information Retrieval, Pattern Recognition, and Report Formatting. These three algorithm types match the functional areas of an expert system: receiving input from its environment; determining an action or response; and delivering an output to its environment.

This natural division of behaviour – information input, direct action, and communication – is mirrored in SUPERVISOR. It looks for a given set of circumstances (an OPAL SITUATION) and then responds by communicating (an OPAL DISPLAY) or by direct action (an OPAL ODTSEQUENCE). In the TRIM world, OPAL programs can enhance the capabilities of the Tape Library system by applying fast, powerful database update and retrieval using simple interfaces. It is necessary, however, to review some of the basics that have already been discussed in earlier chapters.

### OPAL programs

There are three kinds of OPAL programs: SITUATIONs, ODTSEQUENCEs and DISPLAYs. The **DEFINE** command creates, modifies, deletes and interrogates OPAL programs in the program dictionary in the SCHEDULE file that is maintained by SUPERVISOR.

This chapter gives a brief overview on getting started with OPAL programs. It is strongly recommended that the **Metalogic OPAL Reference** manual be used as reference when reading examining the OPAL examples provided here.

## OPAL SITUATIONS

*As someone pointed out recently, if you can keep your head when all about you are losing theirs, it's just possible you haven't grasped the SITUATION.*

Jean Kerr, Please Don't Eat the Daisies (1957)

A SITUATION is a Boolean expression that indicates whether an event has occurred or a condition is present. SUPERVISOR maintains a dictionary of these in the SITUATION Dictionary where the expression associated with each SITUATION is stored. Any number of SITUATIONS may be entered into the dictionary.

In the TRIM world, OPAL SITUATIONS are used to monitor the production of tapes (using TAPELABEL and TAPEDB contexts) and physical tape units (the PER context). An example TAPELABEL SITUATION to filter tapes whose serial numbers start with "B" is shown below:

```
DEFINE + SITUATION TL_FILTER(TAPELABEL) :  
    NOT SERIALNO HDIS "B"
```

This SITUATION, called **TL\_FILTER**, has a context of **TAPELABEL** and uses a string attribute called **SERIALNO** to check the serial number of a specific tape. If the head of the string returned by the attribute does not begin with "B" (the HDIS operator does this, i.e. "Head IS") then this SITUATION will return TRUE.



## OPAL ODTSEQUENCES

*I have striven not to laugh at human actions, not to weep at them, nor to hate them, but to understand them.*  
Baruch Spinoza, Dutch philosopher, from 'Tractatus Politicus'(1677)

An OPAL ODTSEQUENCE is a sequence of statements, separated by semicolons, which describe one or more actions that need to be taken. Often it is necessary to group together a sequence of system commands that are understood to be one logical operational function or to simulate one or more ODT inputs. SUPERVISOR maintains a dictionary called the ODTSEQUENCE Dictionary, where the statements associated with each ODTSEQUENCE are stored.

To accompany the SITUATION shown in the previous section, also called **TL\_FILTER**, the following ODTSEQUENCE will execute the special TAPERECORD statement that updates the METATAPELIB database with all known information concerning the tape that has just been created.

```
DEFINE + ODTSEQUENCE TL_FILTER(TAPELABEL) :  
    TAPERECORD ;
```

If this ODTSEQUENCE was now linked to the **TL\_FILTER** SITUATION via the WHEN command, TRIM would automatically record **all** newly-created labelled or scratch tapes in the METATAPELIB database **except** those whose serial number starts with "B".

```
TT WHEN TL_FILTER DO TL_FILTER
```

Note that the TRIM software does already automatically record all tape activity without the need for any TAPELABEL OPAL programs (as long as the Supervisor option TAPELIB has been set). However, this action is disabled if a TAPELABEL WHEN...DO is active. These simple filtering OPAL programs permit user control over which tapes should be excluded from the database.

## OPAL DISPLAYs

*It was about eleven o'clock in the morning, mid October, with the sun not shining and a look of hard wet rain in the clearness of the foothills. I was wearing my powder-blue suit, with dark blue shirt, tie and display handkerchief, black brogues, black wool socks with dark blue clocks on them. I was neat, clean, shaved and sober, and I didn't care who knew it.*

Raymond Chandler, 'The Big Sleep' (1939)

An OPAL DISPLAY program provides the capability to write user tailored reports to the system ODT, COMS window terminal or line printer. The syntax is based on the syntax of the Unisys **ADM EVENT PRINTLABEL** system command. All references to DISPLAYs are made by a symbolic name determined at the time of definition and must be unique to one DISPLAY. SUPERVISOR also maintains a dictionary of these names, called the DISPLAY Dictionary.

A simple DISPLAY that might look like this:

```
DEFINE + DISPLAY TL_FILTER(TAPELABEL) :  
    SERIALNO, ,TITLE 20, ,CREATIONDATE
```

The above program can be invoked stand-alone or linked to a SITUATION using the **EVAL** or **WHEN** commands. Some examples:

```
TT DISP TL_FILTER B00001  
TT WHEN TL_FILTER DISPLAY TL_FILTER  
TT EVAL TL_FILTER DISPLAY TL_FILTER
```

This DISPLAY program, when invoked, will display a single line of output on the originating station showing the serial number of the tape, its volume identifier (TITLE) and its creation date as a string (DD/MM/YY or MM/DD/YY depending on the setting of USDATES)

## OPAL Contexts

All OPAL programs have a context. A context is the domain in which the information available is founded. Information in an OPAL program is supplied by reserved functions known as *attributes*. Each of which has its own context.

Only the following contexts are available within the TRIM environment:

Context	Gives information about...
PER	Peripheral units
TAPEDB	Tapes, CDs, CD Images, Disk Farms
TAPELABEL	Tapes, CDs, CD Images, Disk Farms

An OPAL SITUATION whose context is TAPEDB or TAPELABEL can be used in two ways. If invoked with the **WHEN** command, a TAPELABEL SITUATION will be triggered each time a tape is created or scratch-ed on the system.

### TAPELABEL vs. TAPEDB

Similarly, a TAPEDB SITUATION linked by a **WHEN** command will be triggered after a TAPELABEL notice has been processed by the TRIM system and recorded on the METATAPELIB database.

The difference is that the attribute subset is much greater for a TAPEDB event and includes rule-based attributes such as RULEID, RULEOWNER or location and tape attributes such as PENDINGLOCATION and EXPIRYDATE. Usually, TAPELABEL based **WHENS** are used for filtering; TAPEDB based **WHENS** are used for reporting, printing labels or exception handling where the TRIM rules system may not be able to handle special cases.

If TAPEDB and TAPELABEL SITUATIONS are used with the **EVAL** command, then the METATAPELIB database will be scanned searching for tapes that return TRUE when the OPAL SITUATION is evaluated. The TAPELABEL context can only scan the database sequentially, in serial number order, examining every record and this can be an expensive process.

## TAPEDB sub-contexts

The TAPEDB context has subcontexts that allow access to METATAPELIB through different DMSII sets:

BYNAME	Scans BYNAME set, ordered by owner and name
BYFAMILY	Scans BYFAM set, ordered by rule
BYLOCATION	Scans BYLOCATION set, ordered by location
SCRATCH	Scans SCRATCHTAPE set, all PGOK or SCRATCH tapes ordered by serial number
PENDING	Scans PENDING subset, all tapes with a pending movement date less than or equal to today's date, ordered by location
BYTS	Scans BYTS set, ordered by creation date and time

Accessing tape information from TAPEDB SITUATIONs will usually be faster if the appropriate subcontext is used since most of the above sets are smaller subsets of the entire database.

If ODTSEQUENCEs and DISPLAy are invoked stand-alone (with **DO** or **' / '** respectively), then a valid tape serial number must be provided as a parameter to extract information for only that tape.

Within the TRIM software, the PER context may only be used to extract system information about tape units (for SITUATIONs, it must be used with a subcontext of MT i.e. PER=MT). PER SITUATIONs, if linked by the WHEN statement, will be triggered every time a tape unit changes status.

Alternatively, if used with the EVAL command, all valid tape units can be scanned. We shall see examples of all types of program later in this chapter.

All the above OPAL programs may access SYSTEM attributes from any context e.g. HOSTNAME, MCP), but other attributes may only be accessed from the appropriate context. Thus OPALs with a PER context can only directly access PER and SYSTEM attributes. Attributes can be listed on hardcopy by entering:

**TT PRINT ATT**

Individual or subsets of attributes can be interrogated using the HELP ATT command.

## DEFINE command

To create, modify or just view OPAL programs, you must use the DEFINE command. DEFINE may be used from a SUPERVISOR COMS window, any system ODT, or MARC.

The syntax of the DEFINE Command for query purposes is

**DEF ? <OPAL program type> <program id>**

To list all programs in the SITUATION, ODTSEQUENCE and DISPLAY dictionaries enter:

**TT DEF ? DEF**

Each OPAL program is listed in alphabetical order within type, beginning with SITUATIONS and continuing with ODTSEQUENCES and DISPLAYS. SUPERVISOR will respond with something similar to:

```
TT +
-- SITUATIONS FOR * ----- OPAL ----- LAST USED ----- CREATED ----- VER
TPDB_ALLMT      PER=MT-      420.08      Never      10:34,03/11/95
TPDB_AUTOPG     PER=MT      420.08      09:00,09/11/95*10:34,03/11/95      3
TPDB_BYFAM      TAPEDB=BYFAMILY 420.08      Never      10:34,03/11/95
TPDB_BYLOCATION  TAPEDB=BYLOCATION 420.08      Never      10:34,03/11/95
TPDB_BYSERIAL   TAPEDB      420.08      Never      10:34,03/11/95
TPDB_DELETE     TAPEDB      420.08      23:52,08/11/95 16:09,03/11/95
TPDB_FILTER     TAPELABEL      420.08      Never      10:34,03/11/95
TPDB_GENS       TAPEDB=BYFAMILY 420.08      Never      10:34,03/11/95
TPDB_LABEL      TAPEDB      420.08      Never      10:34,03/11/95
TPDB_PGREWIND   MSG      420.08      Never      10:35,03/11/95
TPDB_REPORT     TAPEDB=PENDING 420.08      23:01,08/11/95 10:35,03/11/95
TPDB_RESET      TAPEDB      420.08      Never      10:35,03/11/95
TPDB_SCRATCH    TAPEDB=SCRATCH 420.08      Never      10:35,03/11/95
TPDB_STATS      TAPEDB      420.08      10:40,03/11/95 10:35,03/11/95
TPDB_SUMMARY    TAPEDB=BYFAMILY 420.08      Never      10:35,03/11/95
-- ODTSEQUENCES FOR * ----- OPAL ----- LAST USED ----- CREATED ----- VER
TPDB_AUTOPG     PER      420.08      09:00,09/11/95*10:34,03/11/95      1
TPDB_AUTOPG     PER      420.08      09:00,09/11/95*10:34,03/11/95      3
TPDB_BYFAM      TAPEDB      420.08      Never      10:34,03/11/95
TPDB_BYLOCATION  TAPEDB      420.08      Never      10:34,03/11/95
```

Each program in the dictionary is grouped alphabetically by program type (e.g. SITUATION). The OPAL context associated with the DEFINE is also displayed along with usage and timestamp information.

Note the **TT +** in the home position of the screen; as with multi-page ODT commands, the 'TT +' command returns the next page of output.

The following SUPERVISOR command shows all programs held in the SITUATION Dictionary:

```
TT DEF ? SITU

-- SITUATIONS FOR * ----- OPAL ----- LAST USED ----- CREATED ----- VER
TPDB_ALLMT      PER=MT-      420.08      Never      10:34,03/11/95
TPDB_AUTOPG     PER=MT       420.08      09:00,09/11/95*10:34,03/11/95      3
TPDB_BYFAM      TAPEDB=BYFAMILY 420.08      Never      10:34,03/11/95
TPDB_BYLOCATION   TAPEDB=BYLOCATION 420.08      Never      10:34,03/11/95
TPDB_BYSERIAL    TAPEDB      420.08      Never      10:34,03/11/95
TPDB_DELETE     TAPEDB      420.08      23:52,08/11/95 16:09,03/11/95
TPDB_FILTER     TAPELABEL    420.08      Never      10:34,03/11/95
TPDB_GENS       TAPEDB=BYFAMILY 420.08      Never      10:34,03/11/95
TPDB_LABEL      TAPEDB      420.08      Never      10:34,03/11/95
TPDB_PGREWIND   MSG         420.08      Never      10:35,03/11/95
TPDB_REPORT     TAPEDB=PENDING 420.08      23:01,08/11/95 10:35,03/11/95
TPDB_RESET      TAPEDB      420.08      Never      10:35,03/11/95
TPDB_SCRATCH    TAPEDB=SCRATCH 420.08      Never      10:35,03/11/95
TPDB_STATS      TAPEDB      420.08      10:40,03/11/95 10:35,03/11/95
TPDB_SUMMARY    TAPEDB=BYFAMILY 420.08      Never      10:35,03/11/95
```

There may be a need to limit the programs listed. A partial list can be obtained by using a wildcard pattern in the program id. field of the command. The equal sign (=) means match any string where it occurs. The **DEFINE** interrogation form of this command:

```
TT DEF ? DEF TP=REPORT=
```

Will produce a listing of all programs in all dictionaries containing the leading string "TP" and following string "REPORT" in the rest of the identifier name.

## DEFINE interrogation

Listing or checking for the presence of an OPAL program is quite simple. The syntax is

```
TT DEF ? <type of OPAL program> <program id>
```

If the OPAL program does not exist within the type specified, SUPERVISOR will respond with

```
NAME OF A <type of OPAL program> EXPECTED
```

If SUPERVISOR finds a program with that identifier then the program source will be displayed. If the following is entered:

```
TT DEF ? SITU TPDB_EXCEPTION
```

Then SUPERVISOR might respond with:

```
TT DEFINE + SITUATION TPDB_EXCEPTION(TAPEDB=BYFAMILY) :  
    RULEID="EXCEPTION"
```

It is important to note that SUPERVISOR has replaced the question mark (?) in the command with a plus sign (+) in the response to make it easier to modify the program. Care must be taken since this plus sign (+) means that no check is made to prevent overwriting of an existing program (but see also the Supervisor option **DEFINERESP** referred in the section on the **TT SO** command in the **Metalogic Supervisor Reference** manual).

Also, when using the abbreviated form of the command and/or the type of OPAL program, SUPERVISOR will change the abbreviations to their full spelling. DEF is changed to DEFINE, SITU changed to SITUATION.

## Sample OPAL Program

When creating a new OPAL program, it is good practice to first check for the presence of the same type of OPAL program with the same program name. A simple example of creating a new OPAL program follows. (Do not be concerned about understanding this program, it will be covered later in this section.) Programs can be entered in free form beginning at the home position and transmitting after the last statement.

This particular OPAL program is of type SITUATION. (Remember, a SITUATION describes a condition to look for.) The name of this OPAL is **TPDB\_FINDER** and it will search for all the tapes in the database that have "META" somewhere in the title. At the completion of this program we will have a list of serial numbers where this condition has been met.

If you wish to enter this program in your system, you might want to prefix the program name with your initials (**iii\_** in the text below) so as to avoid overwriting any existing programs.

```
TT DEF SITU iii_TPDB_FINDER(TAPEDB) :
    "META" ISIN TITLE
```

SUPERVISOR will respond with:

```
TT DEF SITU iii_TPDB_FINDER(TAPEDB) :
    "META" ISIN TITLE
SITUATION DEFINITION ENTERED
```

Note in the this example, the response does not have a plus sign (+), the **DEFINE** command was not changed from DEF to DEFINE and the OPAL program type was not changed from SITU to SITUATION. This is done for safety. Since we are attempting to create a new OPAL program, we don't want to accidentally overwrite an existing one. Had there been a SITUATION named **iii\_TPDB\_FINDER**, the response would have been

```
TT DEF SITU iii_TPDB_FINDER(TAPEDB) :
    "META" ISIN TITLE
THAT NAME HAS BEEN DEFINED ALREADY
```

Telling us to pick a different name.



Never put in the plus sign (+) yourself – at least not at first. Use the inquiry command

```
DEF ? <type of OPAL program> <program id>
```

and let SUPERVISOR put in the plus sign (+) for you. It's safer!

To modify an existing program, first do an inquiry of the program using the **TT DEF ?** syntax, e.g.

```
TT DEF ? SITU iii_TPDB_FINDER
```

SUPERVISOR will respond by listing the program and substituting a plus sign (+) in place of the question mark (?). Modify the program and then transmit from the end of the program. Remember to place the cursor anywhere after the last statement. SUPERVISOR will respond with

```
<type of OPAL program> DEFINITION ENTERED
```

When creating or modifying an OPAL program, the choice of indentation style is yours. OPAL programs are always free form. You can edit up to 24 lines directly on the ODT or COMS window. When the 24 line limit is reached or if the program becomes too cumbersome, you can save the last compiled version to a disk file, then edit it with CANDE or EDITOR. If you expect a SITUATION, ODTSEQUENCE or DISPLAY program to exceed the 24 line screen limit, you can use a CANDE file from the beginning. The CANDE file should begin with the DEFINE Command, e.g.

```
TT DEF SITU iii_TPDB_FINDER(TAPEDB) :
```

and end with a back slash "\". Note that the use of the **TT** prefix is optional here since it is treated as noise when the command is processed by SUPERVISOR. The ENTER command allows loading of OPAL programs that have been externally created into the \*SCHEDULE file.

To delete an OPAL program from the directory, the **DEF -** syntax is used. SUPERVISOR will respond with the message

```
<type of OPAL program> DEFINITION DELETED
```

If the program was created from a CANDE file, only the entry in the directory of OPAL programs is removed in SUPERVISOR while the CANDE file remains intact. A typical request for a program to be deleted would appear as:

```
TT DEF - SITU iii_TPDB_FINDER
```

SUPERVISOR's response would be

```
SITUATION DEFINITION DELETED
```

To accompany the above SITUATION, the following ODTSEQUENCE, also called **TPDB\_FINDER**, can be used to display some information about a tape when the SITUATION evaluates to true.

```
TT DEFINE + ODTS TPDB_FINDER(TAPEDB) :  
    SHOW(SERIALNO 6,, TITLE 40,, CREATIONDATE);
```

The SHOW statement is a simple way of displaying information to the screen in preference to the DISPLAY statement. SHOW will also automatically perform a line insert operation before printing a line of text so that any previous output is preserved and visible. The '.,' string is shorthand for a single space in each output string.

The numbers following each of the attributes are used to indicate the width of each field, i.e. pad out the SERIALNO attribute to 6 characters, the TITLE attribute for the next 40. This make the output appear tabular on the screen which we will review in the next few sections..

## EVALUATE Command

*When I consider how my life is spent,  
I hardly ever repent.*

Ogden Nash, Reminiscent Reflection.

The **EVALUATE** command will evaluate a SITUATION once and only once. Then, for each element of the context that the SITUATION evaluates true, it will pass that element to an optional associated DISPLAY or ODTSEQUENCE.

The basic syntax for the EVALUATE Command takes several forms:

```
TT EV <SITUATION id>
TT EV <SITUATION id> DISPLAY <DISPLAY id>
TT EV <SITUATION id> DO <ODTSEQUENCE id>
TT EV <SITUATION id> TEST <ODTSEQUENCE id>
```

For example,

```
TT EV TPDB_FINDER DISP TPDB_FINDER
```

Or can be scheduled using the SUPERVISOR AFTER command:

```
TT AF 1130: TT EV TPDB_CHECKIT DO TPDB_DOIT
```

The command

```
TT EV ?
```

Produces a display of the current status of all WHENs, ONCEs, DOs, EVs and DISPLAYs. SUPERVISOR's response will look something like this:

```
TT EV ?
----- SUPERVISOR WHEN STATUS (LIMIT = 40, ACTIVE = 7) -----
001 8428 WHEN PER_MT DO PER_MT RUNNING 001
      TIMES:CPU=00:00:00,IO=00:00:00,ET=01:33:32
      21 evals, 33 Restarts
W 000 8427 WHEN TPDB_AUTOPG DO TPDB_AUTOPG RUNNING 002
      TIMES:CPU=00:00:00,IO=00:00:00,ET=01:33:35
      1 eval, 5 Restarts
W 000 8426 EVAL TPDB_SCAN DO TPDB_SCAN RUNNING 003
      TIMES:CPU=00:00:12,IO=00:00:16,ET=00:00:38
      1028 evals, 286 ODTs entries
001 8425 WHEN META_SMLLBL DO META_SMLLBL RUNNING 006
      TIMES:CPU=00:00:00,IO=00:00:00,ET=01:33:40
      0 evals, 64 Restarts
```

The **EVALUATE** command without an ODTSEQUENCE or DISPLAY is used to debug all types of SITUATIONS. EVALUATE runs the SITUATION program once and then returns whether the SITUATION is currently true or false. SUPERVISOR's response is displayed at the requesting terminal. If the evaluation returns false, it displays:

**SUPERVISOR/<SITUATION name>:RETURNED FALSE**

When the evaluation returns true, different responses will be given depending on the context of the SITUATION.

Context	Response
PER	SUPERVISOR/<situ id>:RETURNED TRUE FOR n UNITS:<unit list>
TAPEDB	SUPERVISOR/<situ id>:RETURNED TRUE FOR n TAPES:<serial list>
TAPELABEL	SUPERVISOR/<situ id>:RETURNED TRUE FOR n TAPES:<serial list>

**TT EV TPDB\_FINDER**

Will EVALUATE this TAPEDB SITUATION and cause SUPERVISOR to respond with

**THE EVALUATE WILL BE DONE**

The METATAPELIB database will now scanned in its entirety which may take some time if there area significant number of tapes present. When completed SUPERVISOR will display the results:

**TT EV TPDB\_FINDER**

**SUPERVISOR/TPDB\_FINDER:RETURNED TRUE FOR 5 TAPES:**  
**000012,000018,000389,004241,004266**

## DO command

*Never do to-day what you can put off till to-morrow.*

Punch Magazine, Vol. xvii, p.241 (1849)

The **DO** command is used to run or stop an ODTSEQUENCE without a controlling SITUATION. Parameters are required to pass information to all types of TRIM ODTSEQUENCE.

Context	Parameters
PER	Unit type and number list
TAPEDB	Serial number list
TAPELABEL	Serial number list

The command:

**TT DO <ODTSEQUENCE> <param list>**

is the syntax used to execute the ODTSEQUENCE unconditionally.

**TT DO + <ODTSEQUENCE> <param list>**

Will first check for a running DO of the same name. If one is running SUPERVISOR will respond with

**ODTSEQUENCE ALREADY RUNNING**

Otherwise the ODTSEQUENCE will begin to execute. The results will be displayed at the initiating terminal. For example, assume the tape serial number "000012" was present in the METATAPELIB database, then

TT DO TPDB_FINDER 000012		
000012	(META)META4242002/FILE000	12/12/1994

The "DO -" (minus sign) form of this command is used to terminate an ODTSEQUENCE that was started with a **DO** or **TEST**. SUPERVISOR will respond with the message:

**THE DO WILL BE DEACTIVATED**

## TEST (DO) Command

*Examinations are formidable even to the best prepared, for the greatest fool may ask more than the wisest man can answer.*

Charles Caleb Colton, Lacon.

The **TEST** form of the **DO** command is used to debug an ODTSEQUENCE. In TEST mode, system commands are not passed on to the MCP for processing but are displayed as messages on the ODT. For example, consider the following ODTSEQUENCE

```
TT DEFINE + ODTSEQUENCE TPDB_FIXIT (TAPELABEL) :
    ODT ("TT TP " , SERIALNO , " EXPIRY " , DATETOTEXT (NEWDATE (TODAY , 10) , DDMMYY) ) ;
```

This program changes the EXPIRY attribute of the specified tape to be changed to the current date plus 10 days; note that both TAPELABEL and TAPEDB contexts will work using the same attributes. Using the TEST command is a good way to verify that the generated command appears correct:

```
TT TEST TPDB_FIXIT 000012
```

will cause the ODTSEQUENCE to begin executing. SUPERVISOR will display

```
THE DO WILL BE DONE
```

at the beginning of the program's execution. All system, Supervisor or TRIM commands will then be displayed. To view all the displays, use the system's MSG command:

```
MSG
---Mix-Time----- MESSAGES -----
* 7302 13:43 SUPERVISOR/TPDB_FIXIT:ODT:TT TP 000012 EXPIRY 20/11/95
```

Alternatively, the temporary replacement of the ODT statement with the OPAL SHOW statement is an excellent way of visually displaying complex commands. SHOW returns the command as a simple message back to the originating terminal and is therefore much more convenient than TEST. When testing is complete, it will be necessary to change the SHOW(..) statements back to ODT(..) before using the program in production.

## Invoke DISPLAY (/) Command

*My Oberon! What visions have I seen!  
Methought I was enamour'd of an ass.*

William Shakespeare, A Midsummer Night's Dream.

The / (single forward slash – DISPLAY Invocation) or DISP command calls upon a defined DISPLAY program and sends the generated output back to the terminal. The character - (hyphen) is a synonym for this command. Parameters must be passed to the DISPLAY as shown in the following table:

Context	Parameters
PER	Unit number list (MT only)
TAPEDB	Serial number list
TAPELABEL	Serial number list

For example, the following simple PER DISPLAY program, MT\_DISPLAY, will show several unit-specific attributes

```
TT DEFINE + DISPLAY MT_DISPLAY (PER) :  
    UNITNUMBER 4,, LABEL 35,,CREATEATE
```

If no unit type and numbers are provided when the DISPLAY is invoked, Supervisor will issue the following error message:

**MORE INPUT REQUIRED**

Assuming that units MT50,51 and 52 are valid, then if the following command is submitted to SUPERVISOR:

```
TT / PER_DISPLAY MT50,51,52
```

The following response would be displayed:

```
TT / PER_DISPLAY MT50,51,52  
  
50 META4242002/FILE000          10/11/95  
51 FLEXTAPE                     14/02/95  
52 DBSPACK95302/FILE000         04/11/95
```

If any of the above specified units were not available or did not currently have a tape loaded, Supervisor will display the warning message:

**INVALID UNIT TYPE FOR UNIT nnnn**



## WHEN Command

*The Right Hon was a tubby little chap who looked as if he had been poured into his clothes and had forgotten to say 'When!'*

P.G. Wodehouse, Very Good Jeeves.

The **WHEN** and **ONCE** commands control monitoring SITUATIONS. The difference between **WHEN** and **ONCE** is that a **WHEN** will continue running until it is deactivated. **ONCE** will stop executing the program after the first time the associated ODTSEQUENCE or DISPLAY has been performed. There are two programs specified in the **WHEN** command: a SITUATION and DISPLAY, or a SITUATION and ODTSEQUENCE. When the **WHEN** or **ONCE** is activated, SUPERVISOR displays the message:

```
SUPERVISOR <program id>+<program id>:BEGIN RUNNING (SLOT n)
```

For example, initiating the example **TPDB\_FILTER** programs with a **WHEN** will cause Supervisor to start filtering all tape notices through the **TPDB\_FILTER** SITUATION. Only tapes that satisfy the SERIALNO check in the SITUATION will then be logged into the METATAPELIB database.

```
TT WHEN TPDB_FILTER DO TPDB_FILTER
SUPERVISOR/TPDB_FILTER+TPDB_FILTER[8950]:BEGIN RUNNING (SLOT 3)
```

A **WHEN** or a **ONCE** is deactivated by leaving off the ODTSEQUENCE or DISPLAY identifier, or by using the “-” form of the syntax. The use of “-” (minus sign) is suggested when terminating a SITUATION/DISPLAY combination when a SITUATION is linked to multiple DISPLAYs. SUPERVISOR will respond immediately with the message

```
THE WHEN WILL BE DEACTIVATED
```

When the program terminates, SUPERVISOR will respond with a termination message:

```
TT WHEN TPDB_FILTER DO
SUPERVISOR/TPDB_FILTER+TPDB_FILTER[6876]:NORMAL TERMINATION
```

## TRIM-specific OPALs

Many examples of TAPEDB, TAPELABEL and PER=MT SITUATIONs are located in a file called OPALS/TPDB which is available on the Metalogic release tape. Usually, this file will be automatically loaded during the INSTALL process and it is likely that the file will have already been ENTER'ed as described in **Chapter 3: Getting started with TRIM**.

This section discusses some of the more useful OPAL programs in some detail and how they could be utilised. It is strongly recommended that you spend some time looking at the **Metalogic OPAL Reference Manual** and at least keep it at hand whilst reading this section.

Two of the most useful OPALs have already been discussed in earlier chapters. The TPDB\_AUTOPG programs discussed in **Chapter 7: TAPEMANAGER implementation** provide a mechanism to automatically purge tapes which have expired (PGOK) as long as the Metalogic TAPEMANAGER implementation is in operation.

Secondly, the TPDB\_LABEL programs discussed in **Chapter 3: Getting started with TRIM** are used to automatically produce tape labels to a simple printer attached to an ODT or remote station.

### TPDB\_DAILYREPORTS

The TPDB\_DAILYREPORTS and associated OPAL programs are currently very important to the correct operation of the TRIM **CONFIRM** authorisation process. It is very likely that this program will be replaced by top-level TP commands in the very near future but they are worth including here as some practical examples of generating reports from OPAL programs.

```
DEFINE + ODTSEQUENCE TPDB_DAILYREPORTS (TAPELABEL) :
  ODT ("TT EVAL TPDB_REPORT DO TPDB_REPORT") ;
\
```

The TPDB\_DAILYREPORTS ODTSEQUENCE consists of a single ODT command that is a Supervisor **EVAL** and is responsible for the generation of the printed tape movement reports that should be performed each day.

## TPDB\_REPORT

The TPDB\_REPORT SITUATION is very simple:

```
DEFINE + SITUATION TPDB_REPORT (TAPEDB=PENDING) :
  TRUE
\
```

By providing a subcontext of PENDING for the main TAPEDB context, the TRIM system will only examine tapes on the database which have a pending movement whose scheduled date is less than or equal to today's date.

All such pending tapes will have an associated pending location which indicates the next destination to which each tape is to be assigned. Since the METATAPELIB database maintains a DMSII set, called PENDING, just for this purpose, then access to this information is very fast. The ODTSEQUENCE is a considerably more complex:

```
TT DEFINE + ODTSEQUENCE TPDB_REPORT (TAPEDB) VERSION 1:
If #Lines = 0 Then
Begin %SET UP
  If NOT LASTEVAL Then
  Begin
    $PL:=PENDINGLOCATION;
    $L:=LOCATION;
  End;
  Call Do "TPDB_REPSETUP"; %SETS UP HEADINGS
  Call Do "TPDB_REPHEAD"
End
ELSE
If #Lines.Sum(1) > #LPP-4 Then
Begin
  PRINT("#[FF]");
  Call Do "TPDB_REPHEAD";
End;
If NOT LASTEVAL Then
Begin
  If PENDINGLOCATION NEQ $PL OR LOCATION NEQ $L Then
  Begin
    $L:=LOCATION;
    $PL:=PENDINGLOCATION;
    Call Do "TPDB_REPSETUP";
    PRINT("#[FF]");
    Call Do "TPDB_REPHEAD";
  End;
  PRINT(SERIALNO 10,TITLE 35,CREATIONDATE 9,
    TAKE (TIME (CREATIONTIME),5) 7,
    #(If EXPIRYDAY=0 Then "UNKNOWN" ELSE EXPIRYDATE) 12,
    STRING8 (REEL,4) 6,STRING8 (MIXNUMBER,5) 6,BOXNUM 8,JOBID); |
  If NOTES NEQ "" Then
  Begin
    #Lines:=#Lines+1;
    PRINT("NOTES: ",NOTES);
  End;
End
ELSE
  Call Do "TPDB_REPFOOT"
\
```

This program's primary purpose is to print a line of information for each tape that has a pending movement. However, it is also responsible for producing headings for the report (including sub-headings for switches in location/pending location combinations) and handling the paging of the report.

The block of code towards the end of the program, bounded by **IF NOT LASTEVAL**, uses the PRINT statement to report various tape attributes to a printer backup file. When the EVAL terminates, this print file will automatically be assigned a print request by the PrintS system.

A fragment from a typical **TPDB\_REPORT** run is shown below; note that the full 120-character print line available in the normal report is not shown.

		COMPANY NAME			
		TAPE MOVEMENT	TO META	EXCEPT	
SERIALNO	TAPENAME	CREATION	PURGE DATE	REEL	MIXNO.
-----	-----	-----	-----	----	-----
R80229	(META)META4242004/FILE000	13/11/95 10:18	11/02/96	1	9234
R80230	(META)META4242004/FILE000	13/11/95 10:41	11/02/96	1	9129
		COMPANY NAME			
		TAPE MOVEMENT	FROM META	EXCEPT	TO SCRATCH P
SERIALNO	TAPENAME	CREATION	PURGE DATE	REEL	MIXNO
-----	-----	-----	-----	----	-----
R70894	(META)META4242002/FILE000	15/08/95 14:31	13/11/95	1	8911
R70895	(META)META4242002/FILE000	15/08/95 14:45	13/11/95	1	8911
R70896	(META)META4242002/FILE000	15/08/95 14:53	13/11/95	1	8924

It is often effective to move common code into separate OPAL modules that are then executed by the CALL DO or CALL DISPLAY command. The string that follows this command must be a valid OPAL program or a run-time error will occur.

## TPDB\_REPSETUP

TPDB\_REPSETUP is an all-purpose program which is used by a number of different other TAPEDB Opals.

This OPAL program is executed via the CALL DO mechanism and is responsible for assisting with the setting up of page headings, determining print defaults and establishing environmental information such as company name, date and time etc.

```

TT DEFINE + ODTSEQUENCE TPDB_REPSETUP(TAPEDB) VERSION 1:
Call Do "TPDB_DEFAULTS"; %SETS UP WIDTH,LPP AND CNAME VARIABLES
$STR:="">& $CNAME;
Call Do "TPDB_CENTRESTRING";
$HEAD1:=$Str;
$STR:=DAYNAME(DAYINWEEK) & " " & TODAYSDATE;
If WHENID(SITUATION)="TPDB_REPORT" Then
    $STR:=$Str & ">TAPE MOVEMENT "&
        (If LOCATION NEQ "" Then "FROM "&$L ELSE "") &
        " TO "&$PL
ELSE If WHENID(SITUATION)="TPDB_BYLOCATION" Then
    $STR:=$Str & ">TAPES AT LOCATION "&$L
ELSE If WHENID(SITUATION)="TPDB_BYFAM" Then
    $STR:=$Str & ">TAPES IN TAPE FAMILY "&$L
ELSE If WHENID(SITUATION)="TPDB_GENS" Then
    $STR:=$Str & ">LATEST GENERATIONS OF TAPE FAMILIES"
ELSE If WHENID(SITUATION)="TPDB_BYNAME" Then
    $STR:=$Str & ">ALL TAPES BY NAME"
ELSE $STR:="">&WHENID(SITUATION);
Call Do "TPDB_CENTRESTRING"; %MANIPULATES STRING IN "STR"
$HEAD2:=$Str;
$HEAD3:=#("SERIALNO" 10,
    (If WHENID(SITUATION)="TPDB_GENS" Then #("LOCATION" 13) ELSE "")
    , "TAPENAME" 35, "CREATION" 16,
    "PURGE DATE" 12, "REEL" 6, "MIXNO" 6, "BOXNUM" 8,
    (If WHENID(SITUATION)="TPDB_GENS" Then "FAMID" ELSE "JOBID"));
$HEAD4:=#("-----" 10,
    (If WHENID(SITUATION)="TPDB_GENS" Then #("-----" 13) ELSE "") ,
    "-----" 35, "-----" 16,
    "-----" 12, "-----" 6, "-----" 6, "-----" 8, "-----");

```

As TPDB\_REPSETUP can be called by different programs, values are generated in the string variables \$HEAD1,\$HEAD2,\$HEAD3 and \$HEAD4, depending on the name of the calling program (the WHENID function is used to check this). In particular, one of the headings, the report title, is centred in the middle of a line by the TPDB\_CENTRESTRING program (not shown here).

## TPDB\_DEFAULTS

Print defaults are easily customisable by changing the TPDB\_DEFAULTS program (this is CALLED by TPDB\_REPSETUP). In particular, if the number of lines per page for your printers is not 66, changing this value in line 2 and re-defining the ODTSEQUENCE will resolve the issue.

```

DEFINE + ODTSEQUENCE TPDB_DEFAULTS(TAPEDB):
    STORE("WIDTH",120); % PAGE WIDTH IN CHARACTERS
    STORE("LPP",66); % LINES PER PAGE
    STORE("CNAME","COMPANY NAME"); % INSTALLATION COMPANY NAME

```

The programs TPDB\_REPHEAD and TPDB\_REPFOOT, respectively, are responsible for displaying heading information on each printed page and a summary line at the end of the report.

## TPDB\_REPHEAD

```

TT DEFINE + ODTSEQUENCE TPDB_REPHEAD(TAPEDB) :
  PRINT(" ");
  PRINT(" ");
  PRINT($HEAD1);           % COMPANY NAME
  PRINT($HEAD2);           % FROM PENDING LOC TO LOCATION
  PRINT(" ");
  PRINT($HEAD3);           % ITEM HEADINGS
  PRINT($HEAD4);           % UNDERLINE HEADING 3
  STORE("LINES",5);        % LINE COUNT SO FAR

```

The OPAL variables \$HEAD1, \$HEAD2 etc are used to produce the heading information. These were previously set up in the TPDB\_REPSETUP program.

## TPDB\_REPFOOT

```

TT DEFINE + ODTSEQUENCE TPDB_REPFOOT(TAPEDB) :
  PRINT(" ", MYSELF(ENTRIES), , "TAPE",
    (IF MYSELF(ENTRIES)=1 THEN "" ELSE "S") , , "REPORTED");

```

The TPDB\_REPFOOT program is only triggered at the very end of the EVAL and is responsible for appending a summary line indicating how many tapes have been reported, in total.

The MYSELF attribute, passing the parameter ENTRIES, will indicate how many time the ODTSEQUENCE has been called.

The presence of a LASTEVAL attribute in any ODTSEQUENCE requests Supervisor to call that program one extra time, once the **EVAL** has completed. In this call, none of the attributes will be valid but it does allow summary information, report totals, etc. to be appended to reports.

There is various other example TAPEDB OPALs that are used to generate standard reports from the METATAPELIB database using the **EVAL** command. For example, the TPDB\_BYLOCATION programs will produce an ordered report by the LOCATION attribute; the TPDB\_BYFAMILY will produce a report sorted by rule identity. Although useful, these routines have been superseded by extensions to the TP FIND command.

## TPDB\_PERRECORD

The TP\_PERRECORD programs can be used to assist with the build-up of the METATAPELIB database using TRIM for the first time. One common problem is how to store information about a site's current library of tapes into the TRIM system. By running a PER=MT SITUATION, linked to an ODTSEQUENCE via a **WHEN** command, Supervisor can check all tapes that are used on the system to make sure that they have an entry in the database.

```
TT WHEN TPDB_PERRECORD DO TPDB_PERRECORD
```

If the tape does not exist, the ODTSEQUENCE is triggered using a TAPERECORD statement to store that tape entry in the database.

```
TT DEFINE + SITUATION TPDB_PERRECORD (PER=MT) :  
    NOT TLOFFLINE("ACHECK") AND NOT TL(SERIALNO)
```

In the above SITUATION, the TLOFFLINE attribute is a Boolean attribute that returns TRUE if the database is unavailable (after a **TP CLOSE** for example). It is irrelevant what serial number is used for this check.

The second attribute, TL, will return TRUE if information about that tape serial number is already resident on the database. So, if the database is on-line and the tape serial number is not already in the database, then the ODTSEQUENCE will be executed.

```
TT DEFINE + ODTSEQUENCE TPDB_PERRECORD (PER) :  
    TAPERECORD;
```

The idea of this WHEN is to capture information about any tapes being used for input that are not currently on the database. This process will not trap scratched or newly created tapes because the TRIM system will already be tracking these by default.

Typically, this kind of **WHEN** should only be needed to capture tape information for a new TRIM installation and it is not usually necessary once the system has settled.

The TAPELOG will show recorded tapes in the following fashion. Note that the OWNER field will always show as UNOBTAINABLE in such cases so TRIM rules will probably not be in effect.

```
TT TP LOG

----- 14/11/95 LOG -----
10:41:41 Evt: RECORD [000013] DBSPACK95317A by UNOBTAINABLE(not SCR)
10:41:24 Err: DENY PG OF 000013 (Not PGOK)
10:40:23 Msg: -- BOT 9713 METATAPELIB UPDATER VERSION 42.420.005 --
10:37:41 ODT: -- EOT METATAPELIB UPDATER --
```

Whilst the TPDB\_PERRECORD OPAL is used, the OWNER attribute of many tapes will be set to UNOBTAINABLE. In most cases, it will be possible to change the OWNER field dynamically within the TPDB\_PERRECORD ODTSEQUENCE. This will require a little extra OPAL programming as well as operational knowledge of which tapes belong to which usercode. The approach is convenient if you wish to assist in the conversion of all your existing tapes to TRIM rules.

For example, to change the OWNER identity for tapes with the pattern "SUMLOGyymmdd" (where "yymmdd" is a numeric date) to the usercode OPS:

```
TT DEFINE + ODTSEQUENCE TPDB_PERRECORD(PER) :
  TAPERECORD;
  STORE("VOLID", DECAT(LABEL,"/",4));          %% VOLUME LABEL
  IF GETSTR("VOLID") EQW "SUMLOG#####" THEN
    ODT("TT TP ",SERIALNO," OWNER OPS");
```

The PER attribute LABEL returns the full name of the tape including the fileid (e.g. SUMLOG951023/FILE000). The DECAT function used in line 2 drops the string "/FILE000" from the label and stores the remainder into the string variable VOLID. The GETSTR function pulls back this value in line 3 and does a wildcard compare with the pattern "SUMLOG#####" (the "#" character represents any numeric character).

If a match is made, then the following ODT command will change the OWNER attribute of the tape to OPS.

This code can be easily extended to handle many usercode-name patterns if desired.



## OPAL functions and attributes

Several useful OPAL attributes and functions are available for the OPAL programmer to help automate the TRIM environment, particularly sites using DSI or UNISYS StorageTek™ Tape Robot systems. These features are used in several important OPAL scripts supplied by Metalogic, described here and in **Appendix B: Tape Robot support**.

These functions and attribute are available for use on both MASTER and SLAVE TRIM systems.

### GETSCRATCH function

The implementation of the GETSCRATCH function allows the control and allocation of scratch tapes, optionally designated by tape DENSITY and SCRATCHPOOL. This feature allows a SUPERVISOR OPAL program to allocate the FIRST eligible tape belonging to a selected subset of tapes and assigns it for "future use".

GETSCRATCH has the following syntax and expects two string parameters and an optional mnemonic, INSILO:

```
— GETSCRATCH — ( <density> , <scratchpool> [ , INSILO ] ) —
```

The <density> string should conform to one of the following known Unisys simple density values such as:

```
200, 1600, 6250, 38000, 1250, 11000, 36TRK, DDS2
QIC1000, DDS3, 128TRK, DLT3, DLT6, DLT10, DLT20, DLT35
```

The <scratchpool> parameter should be a valid SCRATCHPOOL name up to 17 characters in length: if longer, the string is truncated. The <density> parameter should be string is to allow **MX=WAITING** OPAL programs, scanning for **REQUIRES MT** waiting entries, to plug any specified density from the RSVP, directly into the GETSCRATCH attribute. Both parameters may be empty strings.

If a scratch or PGOK tape is found that matches the given <density> and <scratchpool>, GETSCRATCH will return the serial number as a string.

If no matching tape is found, GETSCRATCH will return one of the following error strings:

```

ERR:ARRAY TOO SMALL
ERR:INVALID DENSITY
ERR:DB OFFLINE
ERR:DB OPENERR
ERR:NO POOL OR DENSITY SPECIFIED
ERR:INVALID DENSITY FOR SILO
ERR:NO SERIALS

```

Once a tape has been "assigned" by the GETSCRATCH attribute, it is not "unassigned" until the tape has been used for output. During assignment, several tape attributes are changed: SLOT is set to the value 99999; SENDDATE and SENDTIME are set to the current date and time. Because these attributes are available to OPAL programs, it is possible to monitor for tapes that remain assigned over a long period of time.

GETSCRATCH is especially useful for tape pre-allocation on DSI or Unisys StorageTek™ Robot systems for output media requests. Such mount requests can take time to be satisfied and the control provided by GETSCRATCH avoids the possibility of collisions by marking these tapes "assigned".

```

TP 1

----- METATAPELIB: MT [000001] -----
Volume : A S S I G N E D (TESTNAME/FILE000)
ASSIGNED to host LX100A on 14/07/2004 at 16:12:42
  Cycle      : 0                      Reel      : 0
  Version    : 0                      Density   : BPI1250
  Owner      : FLEX
  Location   :
  Created at : 14/07/2004              Sent date  : 14/07/2004 at 16:12:37
  Savefactor : Unknown
  Pending loc : EXCEPTION              Pending date : 14/07/2004
  Expiry date is UNKNOWN
  Volume last accessed : UNKNOWN
  Created by task 03744 : DATA_ENTRY
  Owned by job no 03743 : DATA_ENTRY
  Usage count : NONE

Marked as : PGOK, ASSIGNED, NON-VOLUMED
Rule info : EXCEPTION

```

Once tape 000001 has been used by the specified task and a new tape entry has been created in METATAPELIB4, the assignment status is removed.

During assignment, several tape attributes are changed: the ASSIGNED flag in the database entry is set to TRUE, and the attributes ASSIGNDATE and ASSIGNTIME are set to the date and time of assignment. Because these attributes are available to OPAL programs, it is possible to monitor for tapes that remain assigned over a long period of time. An example OPAL is available in the OPALS/TPDB file called:

If a tape remains assigned, it cannot be used for any other tape output request. The assignment state can be removed using **UNASSIGN** command.

```
TP 1 UNASSIGN
VOL: 000001 has been UNASSIGNed
```

## MOUNT\_REQUEST example

The MOUNT\_REQUEST OPAL script is an example of handling REQUIRES MT waiting entries, such as:

```
---Job--Task-Pri---Elapsed----- 5 WAITING ENTRIES -----
8892\ 8893 50      1:55 *LIBRARY/MAINTENANCE
      LOGDUMP/FILE000 REQUIRES MT #1 (DLT3) SCRATCHPOOL=DLTPOOL1
```

An ODTSEQUENCE to automatically handle the output tape request might be:

```
TT DEFINE + ODTSEQUENCE MOUNT_REQUEST (MX) :
  IF "SCRATCHPOOL=" ISIN RSVP THEN
    $SCP:=DECAT (RSVP, "SCRATCHPOOL=", 1)
  ELSE
    $SCP:= EMPTY;
  IF " (" ISIN RSVP THEN
    $DEN:= DECAT (DECAT (RSVP, " (", 1), ") ", 4)
  ELSE
    $DEN:= EMPTY;
  IF NOT $SN:= GETSCRATCH($SN,$SCP) HDIS "ERR" THEN
    ODT (MIXNUMBER, "FA SERIALNO=", $SN, " ")
  ELSE
    DISPLAY ("GETSCRATCH ", GETSTR ("SN")) ;
```

This ODTSEQUENCE will automatically assign the serial number of the tape returned by GETSCRATCH to the waiting entry using the FA SERIALNO command.

Once an assigned tape has been processed, the assignment status is removed: the attributes SLOT, SENDDATE and SENDTIME are all set to zero.

Note that GETSCRATCH will also function in the same way from a slave TRIM system. Any assigned tape, selected on the master, will be marked in the normal way except that the requesting HOSTNAME will be that of the slave.

The GETSCRATCH attribute has been modified to allow the provision of a third optional parameter, INSILO, which enforces SUPERVISOR to check that the tape serial returned is physically present in a Unisys StorageTek™ or DSI silo.

Previously, GETSCRATCH returned the first available tape matching \*any\* user-provided criteria in the serial set. Now, due to customer requests, this behaviour has changed. The "best-fit" scratch tape will be one that matches user criteria AND has been PGOK or scratch for the longest period of time. This is determined by checking the SENDDATE attribute, as seen in a TP <serial> response, the oldest tape will then be selected.

## GETSCRATCH: INSILO modifier

The INSILO modifier confines a GETSCRATCH search to tapes that are registered as resident in a Tape Robot silo (only DSI and StorageTek™ are supported), as well as SCRATCH or PGOK.

The tapes need not be loaded on a drive.

**GETSCRATCH ("DLT35", EMPTY, INSILO) ;**

Metalogic support a small suite of programs, including TPDB\_SILOW and TPDB\_EJECT, which handle tape functionality on robot systems. The TPDB\_SILOW OPALs control the assignment of tape serial numbers, using the FA SERIALNO command, for both input and output tape requests.

Please see **Appendix B: Tape Robot support** for more detailed information on the TPDB\_SILOW programs.

## TAPENAMESERIALS attribute

The TAPENAMESERIALS system attribute returns a list of tape serial numbers, as serial number lists, which match a user-provided tape name with, optionally, tape cycle and version. The <tapename> filename should include a usercode (OWNER), volume and file identifiers.

No wild cards are permitted.

```
— TAPENAMESERIALS — ( <tapename> [ , <cycle:ver> ] ) —
```

For example:

```
TAPENAMESERIALS (" (META) DBSPACK123/FILE000" , "4:17")
```

Might return

```
("000012" , "000014")
```

This string represents all serials (in reel number order and as required by the **FA SERIALNO** command) which match the tape name provided and had a CYCLE of 4 and VERSION of 17.

Representing cycle and version as a string in this way facilitates its use in the Metalogic TPDB\_SILOW Opals because cycle and version is easily extracted in this form from a waiting entry:

```
NO FILE METATAPELIB4 (MT) #1 4:17
```

If not usercode is provided, the first tape matching the other parameters is returned regardless of the owning usercode

If TAPENAMESERIALS does not find any matching tape volumes, the string "NONE" is returned to the caller. If <tapename> is invalid, the string "BAD NAME" is returned.

## TAPEFAMILYSERIALS attribute

The TAPEFAMILYSERIALS attribute returns a list of tape serial numbers, as a serial number list, which match a given TRIM rule and generation number. The required parameters are a rule id, including any rule usercode, and optional generation number (otherwise generation 0 is assumed).

For example:

```
TAPEFAMILYSERIALS ( ' (META) FAMILYMANAGER/1 ' , 0 )
```

Might give

```
( '000008' , '000010' )
```

The serial numbers are returned in reel number order, i.e. 00008 is reel 1, 000010 is reel 2 etc. This means that job/task waiting for an input tape can be assigned all required reels by using the following example of an ODT statement:

```
ODT (MIXNUMBER, "FA SERIALNO=", TAPEFAMILYSERIALS ("X" ) , 0 )
```

## TAPEFAMILYNAME attribute

TAPENAMEFAMILY tries to match a given tape name with all the available TRIM rules that currently exist in the METATAPELIB4 database. This filename should be the full tapename, including usercode and fileid if relevant. No wild cards should be used.

For example:

```
TAPENAMEFAMILY ( " (META) DBSPACK123/FILE000" )
```

might return

```
" (META) FAMILYMANAGER/1 "
```

If no matching rule is found, an EXCEPTION rule identity will be returned.

## Chapter 7: TAPEMANAGER Implementation

*The three fundamental Rules of Robotics....One, a robot may not injure a human being, or, through inaction, allow a human being to come to harm....Two...a robot must obey the orders given it by human beings except where such orders would conflict with the First Law...three, a robot must protect its own existence as long as such protection does not conflict with the First or Second Laws.*

Isaac Asimov, "I, Robot"

The Metalogic TAPEMANAGER implementation is intended to provide operations personnel total security against the accidental or intentional purging of ineligible or unexpired tapes by unauthorised programmatic or operator purges. This applies equally well to all kinds of Unisys tape systems including both cartridge and robotic since they all ultimately rely on the capability to purge tapes.

If enabled, the TAPEMANAGER library is an optional, Metalogic-supplied library codefile that has procedure entrypoints invoked by the MCP whenever any of the following events occurs:

- A process attempts to open a tape file.
- A reel switch occurs for a tape file.
- A purge request occurs for a tape volume. Purge requests result from PG and SN system ODT commands, or from program statements that close a file with a disposition of purge.

TAPEMANAGER is currently only concerned with purge requests and does not handle requests for tape usage or reel switching.

The MCP passes various items of tape information to the TAPEMANAGER library, including the tape serial number and volume identity, allowing rigorous checking of the purge request. The TAPEMANAGER authentication process calls entrypoints in the OPALTAPELIB library to ensure that the volume has the PGOK attribute set and that the volume identity matches the entry in the database.

The entrypoints involved in this mechanism are called **CONTROL\_INTERFACE**, **CHECK\_TAPE\_ASSIGNMENT** and **CHECK\_TAPE\_PURGE**.

Since the MCP will explicitly call TAPEMANAGER for all purge requests and cannot proceed until the request has been authorised, this mechanism will also operate successfully on TRIM slave systems where the METATAPELIB database is not directly accessible. In such cases, TAPEMANAGER will automatically interrogate the master database across the network to verify any local purge requests.

## Installation and Operations interface

The Metalogic INSTALL utility will automatically load and SL the following codefile from the release tape:

```
SL TAPEMANAGER = *METALOGIC/TAPEMANAGER
```

The SYSOPS (System Options) system command must be used to set the TAPEMANAGER option. The MCP will NOT call TAPEMANAGER entrypoints unless this option is active. For example,

```
SYSOPS TAPEMANAGER +
```

### Initiating TAPEMANAGER

If the TAPEMANAGER system library has been defined and the TAPEMANAGER system option is set, the TAPEMANAGER software can be initiated in either of the following ways:

Use the ODT command SEND to pass a message to TAPEMANAGER to start handling requests:

```
SEND TAPEMANAGER START  
SEND TM START
```

The TAPEMANAGER system will be automatically initiated.



If the TAPEMANAGER option is set but the TAPEMANAGER library has not yet been invoked, running programs that access files in specific ways may enter the waiting entries with the RSVP:

**#1234 WAITING FOR TAPEMANAGER**

These waiting entries will disappear as soon as the SEND TM START command has been actioned.

### Interrogating TAPEMANAGER status

The status of the TAPEMANAGER library can be determined at any time by using another variant of the **SEND** command, for example:

**SEND TM STATUS**  
**TAPEMANAGER OPTION SET, ACTIVE, MIX NUMBER 1362**

The TAPEMANAGER library can appear in a variety of states:

INACTIVE	TAPEMANAGER library is not currently linked either because it is initialising or currently terminating.
LINKING	MCP is in the process of linking to the library, and the library has not yet frozen
INITIALIZING	MCP is linked to the library, and the CONTROL_INTERFACE procedure is currently executing the START request.
WAITING TO DELINK	CONTROL_INTERFACE procedure has exited after executing the stop request. This state will remain until the MCP has delinked from the TAPEMANAGER library, which will occur after all procedure calls that are currently in progress have completed.
GOINGAWAY	MCP has successfully delinked from the TAPEMANAGER library. The library remains in this state until it finally terminates.
ACTIVE	TAPEMANAGER library is frozen, linked, and available for use. TAPEMANAGER will be linked to the OPALTAPELIB library and capable of calling purge verification routines. This should be the status in normal running.
TERMINATING	SEND TM QUIT system command has been issued. The MCP will begin suspending processes that attempt tape actions, which require access to the TAPEMANAGER library. Although Metalogic's TAPEMANAGER only handles purge requests, all other related tape request handling will still be suspended.

## Terminating the TAPEMANAGER Library

TAPEMANAGER can be terminated in two different ways:

- By entering a DS system command
- Terminating TAPEMANAGER with a SEND Command

Using a '<mixnumber> DS' command will have no effect on the TAPEMANAGER library process unless you first use the '<mixnumber>LP-' command to "unlock" the process.

The DS command is not the preferred method of terminating TAPEMANAGER because it prevents user processes from terminating normally and also inhibits any cleanup actions that might usually be performed during termination.

The preferred method of terminating TAPEMANAGER is to use the following command:

**SEND TM QUIT**

This will cause an orderly close down of the library, deactivating any links with the OPALTAPELIB library and thus the TRIM system.

In the unlikely event that TAPEMANAGER does NOT terminate after a QUIT in a reasonable time, then there are a number of possible reasons for this problem. You can use the SEND TM STATUS command to provide additional information and you should refer to the **Unisys System Ops Guide, Chapter 15: Managing tapes** for more details.

At this point, as long as the SYSOPS TAPEMANAGER option is still active and enabled, the MCP will require access to the TAPEMANAGER library. Waiting entries may now appear:

**#1234 WAITING FOR TAPEMANAGER**

These can normally be resolved by entering **SEND TM START**. To permanently disable the TAPEMANAGER facility and therefore purge protection, the SYSOPS option must also be deactivated:

**SYSOPS TAPEMANAGER —**

### Temporarily disabling of the TAPEMANAGER library

The TAPEMANAGER library can be temporarily disabled without termination by using the SYSOPS ODT command:

**SYSOPS TAPEMANAGER -**

The MCP will resume use of the TAPEMANAGER library when you enter the 'SYSOPS TAPEMANAGER +' form of the SYSOPS command is re-entered. Please note that whilst TAPEMANAGER is disabled, programmatic tape access will be permitted and the MCP behaves as if the library was not there.

In production environments where the TAPEMANAGER implementation is in use, the disabling of the TAPEMANAGER option for long periods using this technique is not desirable and may put your tapes at risk.

## Purge protection

*Innocence always calls mutely for protection, when we would be so much wiser to guard ourselves against it: innocence is like a dumb leper who has lost his bell, wandering the world meaning no harm.*

Graham Greene, *The Quiet American*

If Metalogic's TAPEMANAGER has been SL-ed, the SYSOPS TAPEMANAGER option is active and the library has been activated by a SEND TM START, the tape purge protection facility will now be enabled. This means that the behaviour of all tape-related PG and SN operator commands is changed as follows.

Operator PG and SN requests will be passed to the CHECK\_TAPE\_PURGE verification procedure in the TAPEMANAGER library that is invoked by the MCP. Any commands that are rejected will cause one of the following system display messages:

```
<UNITNO> PURGE REQUEST REJECTED BY TAPEMANAGER
<UNITNO> SERIAL NUMBER REJECTED BY TAPEMANAGER
```

More detailed information about the reason for the rejection of specific commands can be seen from any ODT or SUPERVISOR COMS window using a variant of the **TP LOG** command.

Using the modifier 'ERR' after the command will return all error messages recorded by the TRIM system, including those related to TAPEMANAGER verifications.

```
TP LOG ERR

----- 04/07/95 LOG -----
14:17:34 Err: DENY PG OF META01 (Volumeid not matched)
14:10:38 Err: DENY PG OF META02 (Marked PGNEVER)
10:23:06 Err: RMT: DENY SN OF META88 TO META90 (Not PGOK)
08:12:09 Err: DENY SN OF MT 49 TO META61 (Dup serial on SN)
----- END OF LOG SNAPSHOT -----
```

Each error message for a PG or SN rejection is generally of the following form:

```
DENY PG OF <serial> ( <error message> )
DENY SN OF <serial> TO <serial> ( <error message> )
DENY SN OF <unit> TO <serial> ( <error message> )
```

The <error message> field contains useful information about why the command failed. Each of the categories is shown below and what action can be taken to remedy the problem, where appropriate.

Note that the **TP LOG** command is also accessible from slave systems and any errors are logged in both slave and master TAPELOGs (the master entry will be prefixed by 'RMT:' if the command originated from a slave system).

### Tape assignment by serial number

For output tape requests that have been assigned a specific output volume by serial number (by a run-time file equation or by a '**<mixno>FA SERIALNO**' command, TAPEMANAGER will automatically check that the tape serial number is scratch or PGOK even if the tape is not actually mounted. This is enforced by the use of the CHECK\_TAPE\_ASSIGNMENT entrypoint in TAPEMANAGER that is called by the MCP at the time the serial number is assigned.

If the volume is not PGOK or scratch, the serial number assignment is stripped from the waiting entry and entries are made in the TAPELOG and SUMLOG.

For example, in the following case, tape volume 000061 is not PGOK in the METATAPELIB4 database; a COPY is started explicitly requesting this volume:

```
COPY *SYSTEM/DUMPALL TO MYTESTTAPE
#RUNNING 01930
#BOT 01931 (META)WFLCODE
#BOT 01932 *LIBRARY/MAINTENANCE
#1932 TAPEMANAGER:SERIALNO ASSIGNMENT REJECTED BY TAPEMANAGER
#1932 MYTESTTAPE/FILE000 REQUIRES MT #1
```

Operator enters:

```
1932FA SERIALNO=61
```

Which cause the waiting to activate:

```
#01932 GOING
#1932 TAPEMANAGER:SERIALNO ASSIGNMENT REJECTED BY TAPEMANAGER
#1932 MYTESTTAPE/FILE000 REQUIRES MT #1
```

## TAPEMANAGER errors

All of the following messages may appear in the TAPELOG, indicating the various reasons for a TAPEMANAGER purge rejection.

### Purge Error: Not PGOK

The tape serial number being purged or SN-ed is not marked as PGOK (purge authorised) in the METATAPELIB Tape Library. Tapes must always be marked as PGOK before purge authorisation can be given by TAPEMANAGER.

### Purge Error: Dup serial on SN

An attempt was made to SN a tape to a serial number that already exists in the database. Such an SN command will only be permitted if the new serial number has been marked as DESTROYED, LOST or DAMAGED (set by the appropriate **TP** command).

### Purge Error: Volumeid not matched

The volume identifier of the tape being purged or SN-ed does not match the volume identifier held in the METATAPELIB database for that serial number.

Typically, this error message may be seen when trying to purge a tape with its serial number entry present in the database but which has been subsequently re-used on a different host. In such circumstances, the database entry can be quickly updated using the TPDB\_PERRECORD program:

```
TT DO TPDB_PERRECORD Mtnn
```

where 'nn' is the unit number currently holding the specified tape.

### Purge Error: Database not on-line

The METATAPELIB database was not on-line, most probably due to an off-line dump being taken. Since no checks can be performed, all PG and SN commands will be disallowed.

### Purge Error: MASTER unavailable

This error is only reported for 'slave' systems. It means that a network connection to the TAPELIBUPDATER process on the master system cannot be made. Since the database is unavailable, all tape PG and SN commands will be disallowed.

### Purge Error: Marked PGNEVER

The command is rejected because the serial number is marked as PGNEVER ("never purge" or permanent) in the METATAPELIB database. Permanent status can be removed by using the PGNEVER modifier with the TP <serial> command.

### Purge Error: Volume not in DB

The tape being purged or SN-ed is not in the METATAPELIB database. However, if the unit has been marked as held using the ODT command HOLD MT <unitno> ODT command, then this check is not performed:

```
HOLD MT 45  
MT45 FILE ASSIGNMENT HOLD IN EFFECT
```

The tape can be successfully SN-ed or purged and the HOLD assignment can be relaxed using:

```
HOLD - MT45  
MT45 FILE ASSIGNMENT HOLD NOT IN EFFECT
```

### Purge Error: No serials given

This error message should not normally be seen. Its appearance implies an internal logic problem in the CHECK\_TAPE\_PURGE error handling procedure and should be reported to Metalogic.

### Purge Error: Already assigned to NNNN

Tape volumes are marked as ASSIGNED when they have been allocated by the GETSCRATCH OPAL function for output to a specified user task. Any attempt to re-assign this tape to another task will return this error. The tape volume can be returned to the available scratch pool by using the **TP <serialno> UNASSIGN** command.

### Purge Error: FLEXLIB is off-line

For CATALOGING systems running with FLEX, TAPEMANAGER checks to see if the specified tape has zero backup references before allowing the purge. The Metalogic FLEXLIB library must be accessible if it is SL-ed otherwise this error will occur.

### Purge Error: Tape has backup refs

For CATALOGING systems running with FLEX, TAPEMANAGER checks to see if the specified tape has zero backup references before allowing the purge. If the requested tape has non-zero backup references the purge is rejected.

### Purge Error: B&L tape is not empty

Tape volumes created by B&L software can be protected using TAPEMANAGER by allowing calls into the B&L libraries to determine if a tape may be purged. If the B&L call rejects the request, this error is generated. Please see **Appendix D: B&L Interface** for more information.



## Successful PG and SN operations

If the PG or SN command is authorised, the command is actioned and an appropriate message will be written to the TAPELOG. Such messages may be seen as a subset of the response from a **TP LOG MSG** command.

```
TP LOG MSG

----- 05/07/95 LOG -----
10:18:59 Msg: ALLOW SN OF MT 49 TO R70875 (Volume was UL in DB)
10:04:23 Msg: ALLOW SN OF MT 49 TO R70874
09:35:52 Msg: ALLOW SN OF MT 49 TO R70872
09:09:30 Msg: ALLOW SN OF MT 49 TO R70868
08:10:59 Msg: ALLOW SN OF MT 49 TO R80206
----- 04/07/95 LOG -----
23:30:12 Msg: Volumes confirmed 5 (warnings 0),Links 0,Errors 0
17:11:59 Msg: ALLOW SN OF MT 49 TO DUNF01
17:11:17 Msg: ALLOW PG OF 000025 (Program close purge)
16:02:15 Msg: ALLOW PG OF R80203
16:01:00 Msg: VOL: R80203 purge authorised
15:03:22 Msg: ALLOW SN OF MT 49 TO R70867
14:35:06 Msg: ALLOW SN OF MT 49 TO R80204
14:32:38 Msg: Usage [IPP006] DUNFERMLINE
```

All authorised PG and SN commands will generate a message with the text “ALLOW SN OF ” or “ALLOW PG OF ” in the TAPELOG providing serial number and unit information where appropriate.

Occasionally, an information message will follow the ALLOW text indicating whether a SN command was actioned on an unlabelled tape or if a PG was issued programmatically. For example, in ALGOL:

**CLOSE (T , PURGE)**

## SN considerations

Occasionally, it may be necessary to SN a tape, with a valid serial number already present in the METATAPELIB database, to a new serial number. This is only permitted if the original serial number is marked PGOK and the new serial number does not fail with the error ‘DUP SERIAL ON SN’.

In this case, the original serial number will be automatically marked as DESTROYED and the NOTES field is changed to indicate the new serial number to which it has been SN-ed. Note also that normal tapes marked as UNLABELLED in the database must always be marked as PGOK before allowing a SN command.

## Automatic purging of expired tapes

*To every action there is always opposed an equal reaction or, the mutual actions of two bodies upon each other are equal, and directed to contrary parts.*

Sir Isaac Newton, Principia Mathematica (1687)

In addition to purge protection capabilities, TRIM is also able to provide the automatic purging of expired and write-enabled tapes, as they are loaded into the system, without operator intervention. The TAPEMANAGER software must be operating normally for this feature to be available.

If TAPEMANAGER is active, then a tape can only be purged via the PG and SN commands or programmatically if:

- The PGOK flag is ON or the tape is SCRATCH
- Tape label must match the tape name in the database (even SCRATCH)

If the volume has been created by FLEX on a CATALOGING system, the number of backup references on the tape must be zero.

This facility is controlled by using several simple OPAL programs invoked by the SUPERVISOR WHEN command. To check that these programs, called TPDB\_AUTOPG, have already been loaded into the SUPERVISOR SCHEDULE file, use the **TT DEFINE ?** Command from a COMS window:

```
TT DEF ? DEF TPDB_AUTOPG

-- SITUATIONS FOR * ----- OPAL ---- LAST USED ----- CREATED ---- VER
TPDB_AUTOPG      PER=MT      391.51    12:51,04/07/95*16:35,22/03/95    3

-- ODTSEQUENCES FOR * ----- OPAL ---- LAST USED ----- CREATED ---- VER
TPDB_AUTOPG      PER          391.51    12:51,04/07/95*16:35,22/03/95    1

-- No entries in DISPLAY dictionary subset -----
```

If the **DEFINE ?** command returns no entries in the dictionary, you can load the TRIM OPAL programs using the ENTER command:

**TT ENTER FROM OPALS/TPDB**

When the **ENTER** has completed, repeating the **DEFINE ?** Command from above should show that the programs are now present.

## TPDB\_AUTOPG Opals

From the SUPERVISOR window, enter the following command:

```
TT WHEN TPDB_AUTOPG DO TPDB_AUTOPG
```

This WHEN uses a SITUATION whose context is type PER and is triggered every time a tape unit changes status on the system. If the TAPEMANAGER options have been activated and a loaded tape is write-enabled, not in-use and not already SCRATCH, SUPERVISOR will attempt to purge the tape unconditionally. SUPERVISOR will only attempt to do this when an individual tape is seen for the first time.

Although such automatic purging may seem risky, the TPDB\_AUTOPG SITUATION has built-in safeguards to ensure that the purge request will be rejected, as expected, by the normal TAPEMANAGER checks. Note that, if the SYSOPS TAPEMANAGER option is deactivated, this automatic purging facility will be disabled.

Note also that the TPDB\_AUTOPG programs will also operate effectively from TRIM slave systems.

```
TT DEFINE + SITUATION TPDB_AUTOPG(PER=MT) VERSION 4:
% Automatically attempts to purge any recently loaded tape.
% TAPEMANAGER SYSOP must be enabled.
% 2/17/98: Now do automatic TAPERECORD If tape is not in library
%           instead of attempting PURGE
Serialno Neq Empty And
If Not TLOFFLINE(SERIALNO) And Not TL(SERIALNO) Then
  ( Store("DORCD",1) Eql Empty )
Else
  ( LABELLED And Not INUSE And
    WRITEENABLED And
    Not SCRATCH And
    Not UNLOADED And
    TLPGOK(SERIALNO) And
    Store("DORCD",0) Eql Empty )
```

The above SITUATION program, if associated by a WHEN command to an ODTSEQUENCE or DISPLAY program, will be triggered by SUPERVISOR each time that a tape changes status on the system e.g. the tape is readied onto the system, the tape label is programmatically read or the tape is rewound.

## Tape purging

For the SITUATION to return TRUE for an eligible tape, the following conditions must apply:

- The tape must be labelled and not in-use
- The tape and unit must be write-enabled
- The tape should not already be a SCRATCH volume.
- The tape unit should NOT have a status of UNLOADED or UNITHELD.
- The SYSOPS TAPEMANAGER option must be ON.
- The creation date of the tape volume must be earlier than today's date.
- Lastly, the tape should not be the most recent tape seen before on the specified unit.

The SITUATION checks that the tape has not been seen before on the same unit by storing the volume name and serial number into internal OPAL variables, using the GETSTR and PUTSTR functions. You may need to refer to the **Metalogic OPAL Reference Manual** for more details on string variables and how to manipulate them.

```
TT DEFINE + ODTSEQUENCE TPDB_AUTOPG(PER) VERSION 4:
If Get("DORCD") Eq 1 Then
    TAPERECORD
else If Not SYSOPS(TAPEMANAGER) Then
    Display("Need SYSOPS TAPEMANAGER for AUTOPG.")
Else If Getstr(#(Unitno)) = Putstr(#(Unitno),SerialNo&Title) Then
Begin
    Display("MT ",Unitno,"[",SerialNo,"]"," AUTOPG Blocked");
    Display("Enter TT DO TPDB_RETRY to try again.");
End
Else
    ODT("PG MT ",UNITNO);
```

If the SITUATION does return TRUE, then the associated TPDB\_AUTOPG ODTSEQUENCE is very simple; it will always unconditionally attempt to purge the given tape unit. At this point, the TAPEMANAGER purge protection facility will come into force and ALLOW or DENY the operation, as before.

## Chapter 8: Networking TRIM

*'But,' said Alice, 'the question is whether you can make a word mean different things.' 'Not so,' said Humpty-Dumpty, 'the question is which is to be the master. That's all.'*

Lewis Carroll, Alice's Adventures in Wonderland

For multiple Clearpath hosts connected in a BNA or TCPIP network, the TRIM software is capable of supporting a tape library system in master-slave configurations. For a multi-host network, supporting a tape library database on each system can be expensive because of the extra resources required for associated overheads in CPU, disk resource and operations support (e.g. database dumps). However, these overheads must be weighed against the practicalities of an individual host maintaining a local database. The running of TRIM on slave systems, passing and retrieving information from the master host can minimise these overheads.

If it is decided to operate TRIM as a master-slave configuration, the implementation is very simple to set up. Each slave system has its own recovery protocols invoked after a halt-load, restart or network outage, allowing recovery of queued or missed tape event notices. TRIM components on each slave will initiate dialogs, using a standard port file interface, with the master database system. A master TRIM system can support up to seven other slave hosts and, apart from the set-up of two configuration variables using the INSTALL utility, no other special requirements are needed.

Tape creations and purges are handled in exactly the same way on slave systems as the master, except that the notice is not applied locally but sent across the network. The handling of the TAPELOG and TAPEMANAGER functionality is also identical. TRIM rules will be applied to new tapes as normal though some care may be needed if similar applications are creating the same named tapes on other slaves or the master. Note that TRIM rules are maintained only on the master system and also that most TP commands are permitted from a slave except those that can modify tape information.

Note that BNA HOSTSERVICES is not necessary to run TRIM in a master-slave network. Since TRIM uses a port file interface to pass commands and information between hosts, this library is not necessary.

## Establishing a multi-host TRIM environment

During a normal installation, if the TRIM license key is valid, the Metalogic INSTALL utility will check whether the installing host should be marked as a 'master' or a 'slave' system. As with many of the controls used to configure the environment for Metalogic software, master or slave status is dictated by the settings of two MAGUS configuration variables, TL\_MASTER and TL\_DBHOST, controlled using the INSTALL utility or TP NET command.

If any host is assigned as a master or subsequently moves from slave to master, the INSTALL utility MUST have been previously run to establish the DMSII and TRIM run-time environment. If this has not been done, TRIM will abort during initialization of the OPALTAPELIB library.

Changing from a MASTER to SLAVE is very straightforward but the reverse may be more complex. This subject is discussed in more detail later in this chapter.

Once the TRIM settings have been applied, the **TP NET** command may be used to configure the network environment. By default, INSTALL will set up TRIM master and slave systems to use the BNA protocol. Using TP NET, this can be easily changed to use the TCPIP protocol.

On all systems, the Supervisor **TAPELIB** option must be set.

### Networking in a MASTER TCPIP environment

On the MASTER system (called STIRLING), to set up a BNAv2 TRIM network, the following command is needed:

```
TP NET TCPIP MASTER

TRIM Networking Configuration:
  MASTER system
  Using BNAv2 as network provider
  No slave hosts are currently connected
  METATAPELIB database is ON-LINE

Network Config changes are pending:
  Network protocol will use TCPIP (port #44444)
  System will be configured as >> MASTER <<
```

This change is now marked as pending; by default TCPIP port is set to the default of 44444 but this may be altered using the PORT modifier:

```
TP NET TCPIP PORT 55555

TRIM Networking Configuration:
  MASTER system
  Using BNAv2 as network provider
  No slave hosts are currently connected
  METATAPELIB database is ON-LINE

Network Config changes are pending:
  Network protocol will use TCPIP (port #55555)
  System will be configured as >> MASTER <<
```

To apply the pending changes:

```
TP NET APPLY

Pending Network configuration changes applied
```

TRIM will immediately apply the network changes, close all TRIM applications and restart the network the TCPIP protocol. The following messages are typical:

```
MSG

---Mix-Time----- 44 MESSAGES -----
*42581 14:33 OPALTAPELIB:METATAPELIB DATABASE ONLINE
*42581 14:33 OPALTAPELIB:TRIM MASTER SYSTEM USING TCPIP PROTOCOL (PORT
*42581 14:33 TLUPDATER:TLPORT SUBFILE #3 CLOSED
```

## Networking a SLAVE TCPIP environment

On all slave systems, the following command needs to be entered:

```
TP NET TCPIP SLAVE STIRLING

TRIM Networking Configuration:
  MASTER system
  Using BNAv2 as network provider
  No slave hosts are currently connected
  METATAPELIB database is ON-LINE

Network Config changes are pending:
  Network protocol will use TCPIP (port #44444)
  System will be configured as >> SLAVE <<
  MASTER system will be STIRLING
```

To match the TCPIP port on the STIRLING system, the same TCPIP port number should be used:

```
TP NET TCPIP PORT 55555
```

To apply the changes, the **TP NET APPLY** command must be used on both systems.

```
---Mix-Time----- 44 MESSAGES -----
* 3939 13:45 TLUPDATER:TLPORT SUBFILE #1 CLOSED
* 3939 13:45 OPALTAPELIB:METATAPELIB DATABASE REMOTE
* 3939 13:45 OPALTAPELIB:SLAVE CONNECTION TO POWEREDGE
* 3939 13:45 OPALTAPELIB:TRIM SLAVE SYSTEM USING TCPIP PROTOCOL (PORT 44444)
```

On the slave systems, a slave OPALTAPELIB library codefile will have been loaded and SL-ed by the INSTALL utility. This OPALTAPELIB codefile has no DMSII functionality and will not have been re-compiled.

```
SL OPALTAPELIB = *METALOGIC/OPALTAPELIB/SLAVE
```

To add a new slave system into an existing master-slave configuration, the TRIM software on the master host will register the new slave host as soon as SUPERVISOR has been started. No further action is needed.

## Capabilities of a slave TRIM system

As stated earlier, slave TRIM systems handle tape events and notices identically to a master system. Commands and notices are automatically sent across the network for processing on the master system. Both slave and master TAPELOGs will show tape event information.

From a slave:

```
TP LOG

----- 20/07/04 SLAVE LOG -----
23:00:21 ODT: CONFIRM
20:03:12 Evt: CREATED [DF0050] DEV_A sent POWEREDGE
16:07:13 ODT: From (IPP):CONFIRM
16:06:42 Msg: VOL: 000001 purge authorised (unchanged)
16:06:42 ODT: From (IPP):1 PGOK
```



From the master system, showing the creation of serial DF0050 from host DELL8500MCP:

```
TP LOG

----- 20/07/04 LOG -----
23:03:01 Msg: Volumes confirmed 2 (warnings 0),Links 0,Errors 0
23:03:01 Rmt: CONFIRM
23:00:24 Msg: Volumes confirmed 0 (warnings 0),Links 0,Errors 0
23:00:24 ODT: CONFIRM
20:06:37 Evt: CD IMAGE [CI0020] (FLEX)DEV/CI0020F2004202A
20:03:12 Evt: DELL8500MCP->DISK FARM [DF0050] (FLEX)DEV_A (to META1)
16:35:35 ODT: -- EOT METATAPELIB UPDATER --
16:20:42 Msg: -- BOT 44231 METATAPELIB UPDATER VERSION 50.999.026 --
```

Slave systems have complete access to the entire TP command subset; from SUPERVISOR windows on the slave. All commands are logged locally and to the master system.

Responses to **TP** commands will be generated as expected except that an additional header message is shown, indicating the responding host:

```
TP 66

***** Reply from STIRLING1 *****
----- METATAPELIB: MT [000066] -----
Volume : DBSPACK95137AM/FILE000
Cycle      : 1                      Reel      : 1
Version    : 0                      Density   : BPI1250
Owner      : FLEX
Location   : TAPE RACK              Sent date  : 17/05/2002 at 16:26:51
Creation info: 14:26:15 on 17/05/2002
Savefactor : 30
Host info  : STIRLING1 #6343
Expiry date is UNKNOWN
Volume last accessed : 14:26:15 on 17/05/2002
Created by task 4561 : *LIBRARY/MAINTENANCE
Owned by job no 4347 : MERGE
Usage count : 60 times                Last cleaned : 03/01/2000
Backup refs : 202                     Approx tape  : 223.53 feet

Marked as : VOLUMED
Rule info  : FAMILYMANAGER
Notes     : MERGE 95130 101448
```

Similarly, **TP** modification commands display a similar response

```
TP 44 PGOK-

***** Reply from STIRLING1 *****
VOL: 000044 purge no longer authorised
```

## Switching slave and master systems

*What we call 'progress' is the exchange of one nuisance for another nuisance.*

Havelock Ellis, Impressions and Comments (1914)

Occasionally, it may be necessary to change a TRIM master-slave configuration. Some careful planning and preparation is required if a master and slave system need to be exchanged; it is straightforward to switch a master->slave but slave->master requires much more effort.

This chapter discusses the procedures and precautions that should be taken.

When preparing the new master system, to help minimise incompatibilities, it is advisable to retain the same pack names that were used to hold the database on the old master.

Try to schedule the conversion for a quiet period when tape activity on both systems is minimal.

### Prerequisites

On the MASTER and all SLAVES, the following must be set:

<b>TAPE LIBRARIAN usercode</b>	<b>(i.e. TT USE USER ... FOR TAPELIB)</b>
<b>SUPERVISOR usercode</b>	<b>(i.e. TT USE USER ... FOR SUPERVISOR)</b>
<b>TT SO + TAPELIB</b>	

### Changing SLAVE to MASTER

Initial preparations depend on whether the existing METATAPELIB4 database is being migrated or re-initialized. If migration is required then a DMSII dump should be performed along with a precautionary TRIM backup:

**TP DUMP**

**METATAPELIB4 dumped under (TAPELIB) on DEV**

Transfer the files generated by TP DUMP to the slave system to be designated as MASTER or make the DMSII backups available.

First, change the basic TRIM configuration to MASTER:

**TP NET TCPIP MASTER**

**TP NET APPLY**

If no METATAPELIB4 database has been previously installed on this system, the TAPELIBUPDATER process will appear with a NO FILE:

**6514 09:52 OPALTAPELIB:NO FILE (TAPELIB)METATAPELIB4/CONTROL ON DEV**

This waiting entry should be actioned using the NF command:

**6514 NF**

```
MSG
-----Mix-Time----- 44 MESSAGES -----
6532 09:54 TLUPDATER:TAPELIBUPDATER VERSION 50.00.05
6532 09:54 OPALTAPELIB:OPENERORR : METATAPELIB4 : CAN'T OPEN CONTROL FILE DUE TO
MISSING ROWS
6532 09:54 OPALTAPELIB:DB OPEN ERROR 11, TYPE 49, STRUCTURE 0
6532 09:53 OPALTAPELIB:NO FILE (TAPELIB)METATAPELIB4/CONTROL ON DEV (DK)
6514 09:53 OPALTAPELIB:OPERATOR DSED @ (21540000)
6514 09:52 OPALTAPELIB:NO FILE (TAPELIB)METATAPELIB4/CONTROL ON DEV (DK)
6514 09:52 OPALTAPELIB:TRIM MASTER SYSTEM USING TCPIP PROTOCOL (PORT 44444)
```

At this point, any TP command or tape notice will generate the NO FILE condition.

From CANDE, log on to a privileged usercode and run the Metalogic INSTALL utility:

**U META/INSTALL CONFIG**

Select the TRIM menu and check the value of TL\_CFFAM. This is the family assigned to the location of the METATAPELIB4 CONTROL file and is used by TRIM to determine where to find the flat file created by the TP DUMP command on the MASTER system. If this variable is incorrect or missing, it may be changed at this point.

Copy the METATAPELIB/DUMP/= directory, under the current usercode, to the family specified by TL\_CFFAM so that it may be used by the following INSTALL run.

Run the Metalogic INSTALL utility again:

**U META/INSTALL NEWDMS RESTORE**

The NEWDMS run first checks that the TRIM configuration is correct; if not INSTALL shows various screens to allow the user to configure the environment. It is also possible to configure the CONTROL file family at this point.

Once configuration has been confirmed, INSTALL will perform the following:

- Load and recompile the METATAPELIB4 DASDL
- Recompile the tailored DMSII software and the Metalogic OPALTAPELIB library
- Re-initialize the METATAPELIB4 database using DMUTILITY
- Bring down OPALTAPELIB, TAPEMANAGER and SUPERVISOR
- Restart SUPERVISOR and the TRIM system
- Load tape data from the METATAPELIB/DUMP directory on the CONTROL file family

TRIM should now respond with responses to TP interrogations and a TP VER should show that the system is a MASTER system.

Problems will occur if the METATAPELIB/DUMP files have not been loaded to the CONTROL file or other DMSII configuration items are not set up correctly. However, this mechanism permits a NEWDMS RESTORE to be run many times until the set up is correct. Note that INSTALL will also look on DISK if

Alternatively, once the database has been reinitialised, it may be reloaded from the DMSII backups using DMUTILITY.

## Changing MASTER to SLAVE

To change the MASTER status of a TRIM system to SLAVE and setting the host to STIRLING, depending on the network protocol being used:

**TP NET TCPIP SLAVE STIRLING**

Or

**TP NET BNA SLAVE STIRLING**

Then

**TP NET APPLY**

This applies the network changes permanently. If any problems occur, enter the command:

**TT DELINK OPALTAPELIB**

This terminates the TAPELIBUPDATER process and the OPALTAPELIB library; any TP command or tape activity will restart both processes.

No further action is required.

## Restart and recovery on slave systems

Generally, recovery is much simpler on TRIM slave systems since there are no database considerations, though it is obviously important that no locally generated tape events are missed.

The slave TAPELIBUPDATER task uses a port file interface to pass notices to the master system. If this link is lost or the I/O fails, TAPELIBUPDATER will close and re-open the port file and retry the I/O.

If the BNA or TCPIP network link to the master is unavailable and a queued notice or command is pending, the slave TAPELIBUPDATER process will show the following message:

**NO FILE TLPORT**

TAPELIBUPDATER will not respond to any TP input until this NO FILE condition has been resolved i.e. the network link has been restored. If TAPELIBUPDATER terminates abnormally whilst processing a tape notice, the task will automatically restart and attempt to re-send the notice.

If TAPELIBUPDATER abnormally terminates a second time, the following ACCEPT will be displayed:

**\*\* FILE TLPORT PROBLEM - CANNOT SEND TP INFO TO <HOST> \*\***

At this point, operator intervention is required to acknowledge the problem, a programdump will be taken and the tape notice will be discarded. These are very exceptional circumstances and Metalogic should be contacted in the event that this occurs.

Should Supervisor be unavailable on a slave system for any length of time, the LOGREADER utility will be invoked during Supervisor initialisation to capture any missed tape events. It should be noted that tape usage notices are NOT recorded from slave systems under any circumstances.

If you believe that data may have been lost from either a slave or master TRIM system, a restart of SUPERVISOR will always attempt to recover unprocessed tape notices by running the LOGREADER utility.

## OPAL attributes usage from SLAVEs

Supervisor attributes for the TL system, TAPELABEL and TAPEDB contexts are now available for use from TRIM slave systems. This provides sites with the ability to access tape library information from OPAL even though the METATAPELIB database is not available on the host.

To achieve this, Supervisor passes the request via a port file to the TAPELIBUPDATER task on the master system. Because the network link may not be available or become delayed, OPAL programs that use these attributes may appear with a status of "(WAIT TLREC)" in the response to a **WHEN** interrogation command.

For example, the TPDB\_AUTOPG Opals, discussed in Chapter 7, can be used to provide automatic purge protection on systems that do not have TAPEMANAGER capabilities.

```
TT WHEN ?

----- SUPERVISOR WHEN STATUS (LIMIT = 40, ACTIVE = 6) -----
W 000 1027 WHEN TPDB_AUTOPG DO    TPDB_AUTOPG RUNNING          (WAIT TLREC)
001                                TIMES:CPU=00:00:02,IO=00:00:02,ET=63:33:58
      52 evals, 6 ODTs entries, 10 Restarts
```

Usually, in a case like this, optimisation of OPAL programs using a mixture of TL attributes and non-TL attributes would be of concern on performance grounds especially if network access is required. In particular, it is difficult for the OPAL compiler to optimise TL attributes since they can take a parameter, which is often an attribute itself, such as the SERIALNO attribute from the PER context. This problem, on a slave TRIM system means that a time-consuming port file I/O is required to retrieve tape library information each time the optimisation is broken.

To avoid this problem, Supervisor maintains a temporary "cache" once information about a particular tape has been retrieved from across the network. Any subsequent attempts to get tape information from the master will check the cache before attempting another network access.

The use of TL attributes inside an OPAL program generally reference the same serial number so such a caching mechanism can be very effective. Note that this cached tape data is only valid for 2 seconds, after which time the cache will be invalidated. The cache will also be discarded if the serial number has been the subject of a TAPELABEL event (i.e. a purge or new tape create).

Other attributes such as TAPENAMESERIALS, TAPENAMEFAMILY, GETSCRATCH and functions such as TAPEDB and GETSCRATCH are fully supported.

It should be noted that TAPELABEL and TAPEDB attributes are available for use within event-based WHENs, as normal, on any slave system. Accessing tape information using these contexts from stand-alone DO's or DISPLAY's is permitted but is not currently available using the **EVAL** command.



---

## Appendix A: User Interfaces

*I just come and talk to the plants, really — very important to talk to them, they respond I find.*

Prince Charles, television interview, 21 Sept. 1986

This appendix discusses some of the various ways of communicating with Supervisor and the TRIM system. By far the most effective mechanism is the use of COMS windows because of its ease of use and the extra security features easily configurable with COMS Utility.

The REMOTESPO mechanism is very dated now but it is useful to retain some capability for emergencies. It is likely that Supervisor REMOTESPOs will eventually be de-implemented at some future date. The TTINTERFACE library interface, discussed later, is very flexible because of the excellent paging functionality of MARC and it is very likely that this product will be considerably enhanced in the near future.

It is important to provide user access to TRIM and Supervisor at an early stage because this is the easiest way to become familiar with the package. In particular, it is recommended that the COMS interface is available for the two Getting Started chapters as this is the most convenient to use.

## COMS windows

In earlier releases of Supervisor, the REMOTESPO mechanism was the preferred interface for a user to communicate with Supervisor. This has now been basically superseded by the COMS-window implementation, allowing users access from an MCS window.

Declaring an MCS window called SUPERVISOR requires the following small batch file:

```
CREATE WINDOW SUPERVISOR
    MAX_USERS                = 0
    ,MAX_DIALOGS              = 1
    ,MAX_TRANCODE_SIZE        = 0
    ,REMOTE_FILE              = N
    ,MCS                      = Y
    , TRUNCATED_RESULTS       = Y
    , TITLE                   = METALOGIC/SUPERVISOR
    ,NOTIFY_OPEN              = N
    ,NOTIFY_ON                = Y
    ,ON_TEXT                  = "TT EH"
```

Note that it is not necessary to set NOTIFY\_OPEN to 'Y' since Supervisor will always automatically simulate a **TT WS** command when a new window session has been established. If NOTIFY\_ON is set to 'Y', then **"TT EH"** is useful to see the last response that was generated the last time the user was on a Supervisor dialog for that station.

The above batch file may be compiled into the COMS CFILE using the LOAD command, or the SUPERVISOR window declaration can be interactively created from the Utility window. Once the window has been created, a Supervisor dialog can be readily established, if COMS and Supervisor security controls permit:

```
?ON SUPERVISOR
```

In the current release of Supervisor, both COMS windows and REMOTESPO mechanisms are supported up to a combined total of 16 stations. In a standard Supervisor installation, this is typically split into a maximum of 15 concurrent Windows sessions and 1 REMOTESPO, reflecting the non-preferred status of the REMOTESPO mechanism.

It is possible to configure the maximum number of Supervisor COMS Windows by using the configuration variable SUP\_MAXWINDOWS. The value of

SUP\_MAXWINDOWS indicates how many concurrent Supervisor-COMS windows are permitted at any time; it may be assigned a maximum value of 16 and a minimum of 0. Note that its value will also assign the maximum number of permitted REMOTESPOs made available. SUP\_MAXWINDOWS may be changed by the INSTALL utility using the CONFIG option.

In this case, setting SUP\_MAXWINDOWS to 12 means that 4 REMOTESPO sessions will be available at any time. The default value for SUP\_MAXWINDOWS is 15.

With reference to security, one of the advantages of using a COMS window interface is that it is possible to control access to SUPERVISOR windows using standard COMS features (e.g. the use of STATIONLISTs). However, Supervisor provides a second level of security, which is applied after the station, has passed COMS security validation. The effectiveness of these secondary security checks is controlled through the use of a second configuration variable called SUP\_WINDOWSECURITY.

SUP\_WINDOWSECURITY currently has two settings: MINIMUM and MAXIMUM; if the variable does not exist then the default is MAXIMUM. A value of MAXIMUM enforces any users attempting to open Supervisor COMS dialog to have a usercode, which is PU or SYSTEMUSER. Note that SUPERUSER capable stations, logged-on under '\*' will also be granted access. If SUP\_WINDOWSECURITY is MINIMUM then Supervisor does not perform any security checking; all security is controlled entirely by standard COMS mechanisms.

The **TT USE** command reports the current setting of security level and can change it using the **TT USE WINDOWSECURITY <level>** command.

**TT USE WINDOWSECURITY MAXIMUM**

Changing the level of Window Security will not affect current Window sessions but will be used to validate any new log on attempts. If SUPERVISOR rejects the log-on, the following message will be briefly displayed at the originating COMS station:

**SUPERVISOR: USERCODE does not have sufficient SECURITY clearance**

The station will then be returned to the normal MARC dialog.

There are several commands available to terminate a Supervisor COMS dialog:

**?CLOSE**

**BYE**

**QUIT**

Please note that in the event of any problems with a Supervisor COMS window, issuing a? CLOSE followed by a re-issue of? ON SUPERVISOR (assuming this is your chosen window name) should clear any fault.

Previously, the COMS window implementation within Supervisor relied heavily on code originally designed to support the old REMOTESPO implementation. This code has now been rationalised to separate handling of windows stations from that of REMOTESPOs. Supervisor will invoke a separate task to support stations using each facility.

The WINDOWS task will terminate when the last Supervisor-COMS dialog has been terminated; similarly, the REMOTESPO task will also terminate when the last REMOTESPO disappears.

```
AA MCS METALOGIC/SUPERVISOR
```

```
---Mix-Pri--CPU Time----- 5 ACTIVE ENTRIES (ALL) MCS=METALOGIC/SUPERVISOR
 7438  50      8:06 JOB *METALOGIC/SUPERVISOR/WAITWATCHER ON DISK
* 7872  85      :00 (SUPERVISOR) (SUPERVISOR) SUPERVISOR/REMOTESPO
* 7847  50      :00 (SUPERVISOR) SUPERVISOR/GRINDER
 7688  50      :01 (SUPERVISOR) (SUPERVISOR) SUPERVISOR/WINDOWS
 7427  84      :15 (TAPELIB) (SUPERVISOR) METALOGIC/SUPERVISOR/TAPELIBUPDATER
```

The **TT WINDOW** command has also been implemented to separate the control functions. This command is similar to **TT REMOTESPO** except that there is no **TT WINDOW +** facility. Numbers of in-use window and REMOTESPO sessions are reported in the responses to **TT WINDOW ?** and **TT REMOTESPO ?** commands. See the **Metalogic Supervisor Reference** manual for more details of this command.

## MARC directive

Metalogic also supply an optional MARC Directive library allowing an alternative interface into TRIM and Supervisor. Directives are very advantageous because they allow access to the excellent response paging facilities that are built into the MARC software. Both **TT** and **TP** commands to Supervisor can be routed through the Directive library and normal responses are returned, but with the ability to scroll backwards and forwards through the output. Directives are defined by using a prefix that informs MARC to route text messages to links to the library so it is very convenient to use 'TT' and 'TP' as these prefixes.

For example, a typical response to the **TP SCR** command is shown below. Note the '+' character in the Action field allowing the response to be scrolled.

OUTPUT - MARC COMMAND OUTPUT		11:11:12
Action:<+		>
HOMe GO RETurn COMnd STORE + -		(Press SPCFY for Help)
Response returned at 11:11:08		
----- FIND: PGOK (1250) -----	Location -----	Created ----- Expire
SHIP45:META4040108/FILE000	SCRATCH POOL 15:17	16/06/2002 14/09/19
SHIP46:META4040108/FILE000	SCRATCH POOL 14:14	20/06/2002 18/09/19
000036:DBSPACK95283A/FILE000	SCRATCH POOL 23:11	10/10/2002 31/10/19
----- FIND: PGOK (1600) -----	Location -----	Created ----- Expire
BLLOG7:CMIGTAPE/FILE000	SCRATCH POOL 00:00	20/09/1993 19/12/19
BOB055:META91303B/FILE000	SCRATCH POOL 00:00	30/10/1991 28/01/19
BOB064:TESTPACK/FILE001	SCRATCH POOL 00:00	27/01/1993 27/04/19
BRISTB:METAOECD/FILE000	SCRATCH POOL 00:00	09/02/1985 10/05/19
IFN471:RMVTAPE/FILE000	SCRATCH POOL 12:02	14/06/2002 12/09/19
KEYS1 :KEYSTAPE/FILE000	SCRATCH POOL 00:00	02/10/1989 31/12/19
MDUMP :MEMDUMPA12/FILE000	SCRATCH POOL 00:00	25/09/1990 24/12/19
METAL :BADENTERSCHED/FILE000	SCRATCH POOL 00:00	22/11/1993 20/02/19
METARN:METAPELIB	1:1 SCRATCH POOL 09:19	24/10/1994 21/11/19
META31:META PRIVLIB/FILE000	SCRATCH POOL 00:00	18/02/1984 18/05/19
TP SCR		

The INSTALL utility will automatically load the MARC Directive library from the Metalogic release tape during installation of either full Supervisor or TRIM. The library codefile is called:

**METALOGIC/SUPERVISOR/TTINTERFACE**

The 'TP' or 'TT' directive can be enabled by entering the command:

```
DIRECTIVE TP = METALOGIC/SUPERVISOR/TTINTERFACE  
DIRECTIVE TT = METALOGIC/SUPERVISOR/TTINTERFACE
```

Using the above command would permit access to Supervisor and TRIM commands to all logged-on users, so some security constraints should be considered here. The DIRECTIVE command has modifiers which restrict such access and are signified by the presence of a ":" following the filename:

```
DIRECTIVE TP = METALOGIC/SUPERVISOR/TTINTERFACE: CONTROL
```

The current list of security modifiers is SYSTEM, PRIVILEGED, CONTROL and COMMAND. The above example would restrict the TP command to only those users with COMS CONTROL capabilities. Please refer to the Unisys MARC Operations Guide for a more detailed description of this facility.

Metalogic are intending considerable development of this interface in the near future, enhancing both security and command handling capabilities.

## PRINTLABEL utility

This program is a small utility designed to allow printing of tape labels to a remote datacom printer instead of an ODT printer. The codefile is available on the Metalogic release tape and is called:

**\*METALOGIC/SUPERVISOR/PRINTLABEL**

The PRINTLABEL utility permits a remote datacom printer to be available to receive Supervisor-generated labels because of the capability of a station, through COMS Utility modifications, to be auto-logged on to a default window after any halt-load or COMS restart.

To use this utility, you will need a printer station declared to the SYSTEM/COMS MCS and a WINDOW and PROGRAM, both called PRINTLABEL, defined by COMS Utility. This small batch file is also on the release tape :

**\*OPALS/SUPERVISOR/PRINTLABEL/COMS**

Please note in the CREATE station code that you will need to change the HOSTNAME from STIRLING1 and the DEFAULT USERCODE from TAPELIB to a valid HOSTNAME and USERCODE applicable to your site.

An example batch file (suitable for the Utility LOAD command) is listed here.

```
CREATE PROGRAM PRINTLABEL
  REMOTE_FILE          = Y          , REMOTE_USERS          = 0
  , TITLE              = METALOGIC/SUPERVISOR/PRINTLABEL
  , USERCODE           = ". "      , FAMILY               = ". "
  , CHARGECODE         = ". "
  , PRIORITY           = 50
  , SECURITY_CATEGORY_LIST = ALL      , DATABASE           = NONE
  , MIN_COPIES         = 0          , MAX_COPIES          = 1
  , INITIATION_TIME_LIMIT = 00:00
  , TERMINATION_TIME_LIMIT = 00:00 , INPUT_QUEUE_MEMORY_SIZE = 0
  , INSTALLATION_DATA   = NONE
;
```



```

CREATE WINDOW PRINTLABEL
    MAX_USERS                = 1      ,MAX_DIALOGS                = 1
    ,MAX_TRANCODE_SIZE       = 0      ,DIALOG_PRIORITY         = 1
    ,REMOTE_FILE             = Y
    , REMOTE_PROGRAM         = PRINTLABEL
    ,MCS                     = N
    ,PROTECTION              = NONE
    ,NOTIFY_OPEN             = N      ,NOTIFY_ON                 = N
;
CREATE STATION PRINTLABEL
;
    DEFAULT_WINDOW           = PRINTLABEL
    ,HOSTNAME                = STIRLING1
    ,DEVICE_TYPE             = DEFAULTDEVICE
    ,DEFAULT_USERCODE        = TAPELIB
    ,SECURITY_CATEGORY_LIST  = ALL
    ,CONTROL                 = Y      ,SUPER_USER                = Y
    ,SYSTEM_USER             = Y      ,PRIVILEGED_USER           = Y
    ,CONTINUOUS_LOGON        = N      ,TRANCODE_POSITION         = 1
    ,TIMEOUT_INTERVAL        = 0:00
    ,CLOSE_ACTION            = 1
    ,CLOSE_WINDOW            = PRINTLABEL
    ,INSTALLATION_DATA       = NONE
;

```

For SUPERVISOR to pass tape label notices to this program you must have a TAPEDB SITUATION and ODTSEQUENCE running linked via a **WHEN** command. An example SITUATION and two ODTSEQUENCES are provided in the OPALS/TPDB file, loaded from the release tape during the Metalogic software installation.

These OPALs can be linked together using the **WHEN** command:

```
TT WHEN TPDB_LABEL DO TPDB_LABEL
```

Alternatively, if your system is using a tape cartridge system, then the label format will need to slightly different so you should use the following **WHEN**:

```
WHEN TPDB_LABEL DO TPDB_CARTLABEL
```

All of the above programs are easy to customise and can be changed to suit site requirements.

## RECORD[47]

The ODTSEQUENCES use the RECORD[47] statement to communicate to the PRINTLABEL utility via port file. It is important, if you use this mechanism, that you have no other ODTSEQUENCES, which might already be using RECORD [47]. If you do already use index 47 in RECORD statements or wish to use a different number for any other reason, then it is possible to modify the PRINTLABEL code file, allowing any number from 5 to 47 to be used as the new index. You need to do the following:

```
COPY *METALOGIC/SUPERVISOR/PRINTLABEL AS METALOGIC/LOCAL/PRINTLABEL  
FROM <FAMILY>(PACK) TO <FAMILY>(PACK)
```

```
MODIFY *METALOGIC/LOCAL/PRINTLABEL;TASKVALUE=22
```

Where 22 would be the new RECORD index to be used in the TPDB\_LABEL ODTSEQUENCE that will need to be recompiled.

```
RECORD [22] (/,  
    "METALOGIC TAPECONTROL" ,/,  
    . . . . .  
    //// );
```

Remember to update the definition of the PROGRAM PRINTLABEL in the COMS CFILE to use the new METALOGIC/LOCAL/PRINTLABEL codefile.

## Appendix B: Tape Robot support

*The three fundamental Rules of Robotics....One, a robot may not injure a human being, or, through inaction, allow a human being to come to harm....Two...a robot must obey the orders given it by human beings except where such orders would conflict with the First Law...three, a robot must protect its own existence as long as such protection does not conflict with the First or Second Laws.*

Isaac Asimov, I, Robot (1950) "Runaround"

Specifically, in this discussion, the support of robotic tape systems by the TRIM package will refer only to the StorageTek™ tape silos, which are becoming increasingly available on A-Series machines. The TRIM rules mechanisms work just as effectively for such tape library systems but specific handling processes, such as ejections of tapes being moved to off-site locations from the silo, require special actions.

The TRIM software will link directly to StorageTek™ supplied software, known as the "CSC library", allowing messages from the silo hardware system to be passed directly into Supervisor. As well as receiving event-related information in real-time from the CSC library, this interface allows commands to be passed from Supervisor to the silo system to perform tape operation functions e.g. mounting or dismounting tapes.

A very small subset of OPAL attributes, within the TAPEDB context, supports both the StorageTek™ and DSI robots. TRIM supports the DSI TM command and a simple TK command is available for Unisys StorageTek™ systems.

There are many variants of the hardware available to customers but these TRIM interfaces provide additional functionality by communicating only with the software components of both systems.

It should be emphasized that the basic TRIM package already supports a variety of Cartridge Tape Library (CTL) systems, **without specific enhancements**, because TRIM receives tape scratch, creation and usage notices directly from the MCP as with conventional tape subsystems. This means that, as to the operating system itself, such CTL systems appear to the TRIM software in the same way as any other tape subsystem.

## Robot Tape Silos and TRIM

TRIM now supports all currently known tape densities though the retention of this information in the METATAPELIB database needs some explanation. The DENSITY field of a tape entry in the database is numeric and stores two subsets of values. The first set includes the values 1250, 6250, 38000 and 11000 which are actual density settings still used on many old and new Unisys systems.

The second subset includes the integer value of the density attribute for the tape. For example, a value of 8 for DENSITY means 36TRK, 9 is FMTDDS2, 10 is QIC1000, 11 is FMTDDS3, 12 is FMT128TRK, 13 is FMTDLT3, 14 is FMTDLT6, 15 is FMTDLT10, 16 is FMTDLT20 and 17 is FMTDLT35. It should be noted that this is only of importance to the OPAL DENSITY attribute in the TAPEDB context.

Attributes in the PER and TAPEDB contexts have been updated to reflect these changes and a new attribute DENSITYID returns the DENSITY mnemonic e.g. BPI1250, FMTDLT35, FMT128TRK etc. The normal responses to TP interrogation and TP FIND SCR commands will always show the correct density.

With the arrival of large capacity tape drives which can hold very large databases on a single reel, it is now likely for database audit dumps to be initiated whilst an on-line dump is still running.

In earlier TRIM releases, if rules were being used to "link" the audits with the master database dump, this would cause the audit to become linked to a **previous** database dump. This was because the TRIM system only received the tape creation notice once the on-line dump was complete. Although this problem relies on the customer using "linked" rules, this is a very common of handling database dumping and Metalogic decided to address this important issue in the following way.

Instead of just tracking tape creation at the time the label notice was produced, TRIM will now detect specific LOG VOLUME events for new tapes, which have just been opened for output. A "raw" tape entry is written into the TRIM database, which will be assigned a creation date and time for the tape reflecting the **actual open time**. Although termed "raw", rules are applied automatically applied and the tape entry is virtually complete except that job and task name attributes are marked as "TAPE\_IN\_USE".

When the MCP generates a tape creation notice, the job and task attributes are updated in the TRIM database entry but the creation date and time is NOT changed. The event entries in the TAPELOG will appear as:

```
01:39:15 Evt: CREATED [000026] DEV99021A by FLEX
01:30:18 Evt: NEW VOL [000026] DEV99021A by FLEX
```

In the above example, the response for a TP 26 command would show the true creation time to be 01:30:18 whereas earlier TRIM systems would have shown a time of 01:39:15. This important change protects the linking mechanism by allowing potential new linkages to the correct master tape.

Where an application creating the tape is aborted before completion, the "raw" entry will remain in the database unless the caller does a programmatic CLOSE with PURGE. An example of this is a normal COPY JOB, which was terminated before any files were copied to the tape; here, MCP automatically purges the tape as part of its clean up.

## Functionality

The following TRIM activities apply to tape robot systems:

- Tape creations/scratches are captured in real-time
- Auto-purging and purge protection using the Metalogic TAPEMANAGER library
- OPAL scripts used to generate database reports will work without recompilation or modification
- LOGREADER utility will check for and process any "missed" cartridge creations in the event of a restart
- TRIM rules will operate as before with the usual expiration and movement controls
- Daily reports and TP CONFIRM phases operate as normal
- Full TP interrogate/modify command functionality

## Unisys StorageTek™ robot support

TRIM expects the CSC library software to be installed and the TAPESERVER MCP interface should be active (SYSOPS TAPESERVER ON and SEND TS START). DSI support is enabled through the following SUPERVISOR command:

**TT USE SILO UNISYS**

This command changes an internal configuration variable, TL\_SILOKIND, and causes the TAPEMANAGER library and TRIM to be advised of the change. No restart of the TRIM software is required.

The following messages will be seen:

MSG

```
-----Mix-Time----- 44 MESSAGES -----  
6671 13:08 TAPEMANAGER:ALL TAPEMANAGER REQUESTS WILL BE HANDLED LOCALLY  
6671 13:08 TAPEMANAGER:APPLYING SILO SUPPORT CHANGES
```

In the case of StorageTek™ tape subsystems, currently available from Unisys, TRIM interfaces only with the Unisys CSC Library 2.1 (version 43.015) or greater. This CSC software, written by the University of Washington, emulates the Unisys TAPESERVER interface which is driven through a MCP interface; this is discussed in more detail later.

The software currently supports many variants of the StorageTek™ systems (e.g. Nearline, TimeberWolf, Powderhorn) using various drive technologies such as DLT4000, DLT7000.

It should be noted but TRIM is no longer compatible with sites using the older CSC software (version 2 or earlier) that used StorageTek™'s own proprietary interface and is totally different from the TAPESERVER implementation.

If the UNISYS operational mode is active, then SUPERVISOR automatically links to the TAPESERVER entrypoint in the MCPSUPPORT SL function. The TK command subset is available here, allowing interrogation of cartridge details and the ad-hoc changing of some individual attributes in the silo.

As with DSI, a separate task is invoked to handle the library linkage, TK commands and CARTINSILO requests. As with the DSI implementation, TAPEDB attribute support is available which to retrieve cartridge status information from the robot hardware.

## TK command

The CSC software package from Unisys is not a complete implementation of the Unisys RoboHost software so there is limited command functionality.

The SUPERVISOR **TK** command allows interrogation of an individual serial number to see if it is currently registered in a tape silo and also the ability to physically eject a tape from the silo.

```
— TK — <serial no> —————|  
                               |  
                               | EJECT |  
                               |
```

```
TK META01  
  
---- Cartridge status ----  
Cartridge META01 is REGISTERED  
SCRATCH flag is OFF
```

The EJECT modifier requests the CSC software to initiate an eject of the specified tape from the robot.

```
TK META01 EJECT  
  
Cartridge META01 eject requested
```

As with the DSI implementation, all TK commands and responses to Supervisor windows are written to the TAPELOG and are retrievable through the TP LOG command.

## Supported OPAL attributes

The only special attribute available is the **CARTINSILO** attribute in the TAPEDB context.

This attribute allows an OPAL script to determine if a cartridge is currently registered in a silo; this fact can be used for controlling automatic off-site tape movement and tape input/output handling. It is possible that in the future, the CSCLIB software may be enhanced to support more TAPESERVER entrypoints. Since TRIM already implements many of these calls, there are numerous cartridge attributes that already exist and may become useful.

For example, to select all tapes currently resident in the METATAPELIB4 database AND registered in the tape robot:

```
DEFINE + SITUATION INSILO (TAPEDB) :  
    CARTINSILO
```

```
EVAL INSILO  
SUPERVISOR/INLINE_TAPEDB_2:RETURNED 2 TAPES META01,META02
```



## DSI robot support

With DSI systems, as far as is known, TRIM will interface directly to their TapeManager software, irrespective of software version.

The DSI TM command interface is supported for use from OPAL scripts and SUPERVISOR-COMS windows.

With reference to the DSI software, some previous knowledge of how their software functions is required to understand the different modes of operation.

There are four possible settings for the **USE SILO** command:

### USE SILO DSI

This setting provides enhancements for the DSI Cartridge Tape Library. By default, all tape assignments are passed through to the DSI CSCLIB library by the Metalogic TAPEMANAGER, for processing by the DSI TAPEMANAGER library. A TM STATUS will show the operational mode as ACTIVE.

In all the DSI settings it is assumed that Metalogic's own TAPEMANAGER library is SL-ed to the TAPEMANAGER function; this is necessary to support auto-purging and purge protection facilities provided by TRIM.

### USE SILO DSI\_SERIALNO

This setting is similar to DSI, except that tape assignment requests are only passed to the DSI library if a tape serial number has already been assigned in the request.

If a tape has both a SERIALNO and SCRATCHPOOL assignment, the SCRATCHPOOL setting is automatically removed.

## USE SILO DSI\_NOTM

This setting allows the DSI TAPEMANAGER to still run in ACTIVE mode but any tape assignment requests will NOT be passed to the DSI software by Metalogic's TAPEMANAGER.

In all the above DSI cases, the Metalogic TAPEMANAGER will only pass through control and tape assignment requests to the DSI TAPEMANAGER. If no DSI mode is active, Metalogic's TAPEMANAGER will not forward any requests to the DSI software.

Tape purge requests are NEVER passed to DSI software regardless of the operating mode.

## TM command interface

The DSI TM command interface is supported from COMS windows, the ODT and OPAL scripts (must be prefixed by TT). Please refer to the DSI Tape Library software reference manuals for more information.

When a TM command is used or a cartridge attribute referenced for the first time, SUPERVISOR invokes a subtask called DSIHANDLER, which handles linkage to the DSI libraries to satisfy the request. Any reference to the CART attributes within an OPAL script, as described above, causes the calling Supervisor slot to appear with a status of WAIT SILO in the response to a WHEN ? command. This status is only temporary, as soon as DSIHANDLER has processed the request, the WHEN will continue as normal.

A new variant of the DELINK command allows the graceful delinking of Supervisor from the DSI system libraries, CSCLIB and TAPELIBRARYSUPP:

### **TT DELINK DSI**

Further OPAL attributes are also planned in the near future which will allow TRIM or Supervisor to track the status of DSI hardware e.g. whether a door is open or closed, specific slots are occupied.

## Supported OPAL attributes

Several additional attributes are available when DSI support is enabled. These include:

**CARTINSILO**  
**CARTSLOT**  
**CARTLIBRARYID**

These attributes allow the OPAL programmer access to useful information about tapes in one or more tape robots.

It is strongly recommended that TRIM rules are used to control tape expiration and movement - in particular to assist with off-site storage if required. By using a default location of say, DSI\_ROBOT, to signify that a cartridge should remain in the library any OPAL script handling the export can automatically generate an export by checking the current location of the tape from the database.

Although the DSI software can handle the automatic mounting of scratch tapes from the library to a drive by detecting the waiting entry, it may be better for Supervisor to provide this facility where the TRIM database is being used. An example, OPAL, TPDB\_SILOW, shows how the GETSCRATCH function can be used for waiting entries requiring output tapes with a SCRATCHPOOL assignment.

To ensure that only tapes from the DSI robot are loaded, all the tapes used by the robot would need to be SN-ed or PG-ed with a SCRATCHPOOL assigned.

By necessity, requirements will vary from site to site but these example scripts are a good basis for local customisation.

## Example OPALs for Robot Automation

Metalogic provide several useful example OPALS that can be used to automate some of the day-to-day running of both the DSI and StorageTek™ tape robots. The OPALs discussed here may be found in the released file:

### OPALS/TPDB

This file is automatically loaded by the Metalogic INSTALL utility during a software installation.

### TPDB\_SILOW Opals

The implementation of the GETSCRATCH attribute allows the control and allocation of scratch tapes, optionally designated by density and scratchpool. These facilities allow a SUPERVISOR OPAL program to allocate the FIRST eligible tape belonging to a selected subset of tapes, assigning it for "future use".

TPDB\_SILOW consists of a SITUATION and ODTSEQUENCE, linked via a WHEN. The SITUATION, shown below, is triggered on waiting entries requesting both input and output tape requests

```
TT DEFINE + SITUATION TPDB_SILOW(MX=W) :
(RSVP Incl " REQUIRES " And Decat(RSVP,"REQUIRES ",1) Incl "MT " And
RSVP Incl {" SCRATCHPOOL=DLT35"} And      % Site changeable
Not RSVP Incl "]" #")          % Not got a SERIALNO already
Or
(RSVP Hdis "NO FILE " And
(RSVP Incl "(MT)" And Not RSVP Incl "FIND ON ") And
Not RSVP Incl "]" #")
```

The SITUATION only selects tape requests that do NOT already have a serial number associated with the request; this is because the FA SERIALNO command issued by TPDB\_SILOW will cause the task to go waiting again, until the tape has been mounted, and we do not want the waiting entry to be triggered again.

The ODTSEQUENCE is shown below:

```

TT DEFINE + ODTSEQUENCE TPDB SILOW(MX):
% For SILO sites using StorageTek™ or DSI libraries
% This ODTs is triggered for NO FILE(1) and REQUIRES MT(2) conditions
% (1)... Uses TAPENAMESERIALS to get best serialno match
% (2)... Uses GETSCRATCH, passing waiting entry SCRATCHPOOL and
%         DENSITY extracted from the RSVP (If present)
% Doing the FA SERIALNO triggers the StorageTek software/hardware
% to handle the MOUNTs automatically.
% Modified 11/11/1999 for INSILO protection
%         15/02/2000 for more flexible input tape handling
%         03/06/2000 Cycle version handling NO FILE X(MT) #1 12:44
%         14/03/2002 Handling for QUICKCOPY tapes with multiple audits
If RSVP Hdis "NO FILE " Then
Begin
$User:=If USER=EMPTY Then EMPTY Else #("(" ,USER,"");
$Tape:=Head(Decat(RSVP,"NO FILE ",1),Not " ");
If RSVP Eqw "=#:#=" Then
$Gen:=Trim(Decat(Decat(RSVP,"#",1)," ",1))
Else
$Gen:=EMPTY;
$SN:=TAPENAMESERIALS($User&$Tape,$Gen);
If $SN Eq1 {"NONE","BADNAME","INVALID CYCLE"} Then
$SN:=TAPENAMESERIALS($Tape,$Gen);
%% Last chance. Check for QCAUDIT which have used QUICKCOPY APPENDs
%% Validate first with usercode and then, If necessary, without
If $SN Eq1 {"NONE","BADNAME","INVALID CYCLE"} Then
If $Tape Incl "/QCAUDIT" Then
Begin
If $Fam:=TAPEFAMILYNAME($User&$Tape) Eq1 "EXCEPTION" And
$Fam:=TAPEFAMILYNAME($Tape) Eq1 "EXCEPTION" Then
$Sn:="NONE"
Else
$Sn:=TAPEFAMILYSERIALS($Fam);
End;
If $SN Neq {"NONE","BADNAME","INVALID CYCLE"} Then
ODT(MIXNO,"FA SERIALNO=", $SN)
Else
DISPLAY("Tape match failed due to ", $SN, " error");
End
ELSE
Begin
$Pool:=Head(Decat(RSVP,"SCRATCHPOOL=",1),Alpha);
If RSVP Incl "(" And RSVP Incl ")" Then % Got a density
$Den:=Decat(Decat(RSVP,""),4),"(",1);
$SN:=GETSCRATCH($Den,$Pool,INSILO);
If $SN Hdis "ERR:" Then
DISPLAY("NO AVAILABLE SCRATCH FOR ",MIXNO,, $SN)
ELSE
Odt(MIXNO,"FA SERIALNO=","",$SN,"");
End;

```

The TPDB\_SILOW Opals are linked via a WHEN command:

```
WHEN TPDB_SILOW DO TPDB_SILOW
```

This WHEN is specifically intended to automatically handle requests for scratch tapes in silo systems by automatically allocating and assigning the next available tape matching density and scratchpool.

This OPAL detects waiting entries for both input and output tapes. For output tape requests that have the appropriate scratch pool, the ODTSEQUENCE will automatically FA the serial number of a suitable scratch/PGOK tape changing the status of the waiting entry.

TPDB\_SILOW uses the TAPENAMESERIALS attribute to obtain the serial number of the latest generation matching the name provided, optionally with usercode, if it is present. If the request includes a CYCLE and VERSION, then the TAPENAMESERIALS call is extended to include the genealogy.

For output requests, the GETSCRATCH function is used, passing the required tape DENSITY and SCRATCHPOOL if present.

In both cases, TPDB\_SILOW will automatically issue FA SERIALNO commands to the waiting entry, if valid tape serial numbers have been obtained, which will then initiate tape mounts by the CSC or DSI software. Whilst the robot software is loading the requested tape(s), these tapes cannot be used by other tasks unless they are subsequently unassigned.

## TPDB\_EJECT Opals

These OPALS depend on TRIM rules providing offsite movement for locations OUTSIDE of the tape robots. The SITUATION can identify such tapes by the tape density, PENDING LOCATION or SCRATCHPOOL.

In the Metalogic example below, the SITUATION only considers tapes with a DENSITY of DLT35 and PENDING LOCATION of VAULT, EDINBURGH or GLASGOW. The tapes must be REPORTED (i.e. processed by the TPDB\_DAILYREPORTS phase) and currently in the silo.

```
TT DEFINE + SITUATION TPDB_EJECT(TAPEDB=PENDING) :
  % This OPAL checks for tapes moving to an off-site location and
  % will issue TK <serial> EJECT commands where needed.
  % 1. It MUST be run between the TPDB_DAILYREPORTS and CONFIRM phases
  %    since pending locations are being checked.
  % 2. Density lists can be provided to optimise the search
  % 3. Insert the list of PENDING LOCATIONS which are offsite
  REPORTED And
  DENSITYID = FMTDLT35 AND
  PENDINGLOCATION Hdis {"VAULT","EDINBURGH","GLASGOW"} And
  CARTINSILO
```

The TPDB\_EJECT ODTSEQUENCE is simple:

```
TT DEFINE + ODTSEQUENCE TPDB_EJECT(TAPEDB):  
  If Not LASTEVAL And SUM ("COUNT", 1) > 0 Then  
    Odt ("TT TK ", SERIALNO, " EJECT")  
  Else  
    If #Count > 0 Then  
      DISPLAY ("Cartridges queued for EJECT: ", #Count)  
    Else  
      DISPLAY ("No cartridges EJECTed for off-site movement");
```

In the above, the TK.. EJECT command is used for each serial number present in the silo; both robots handle the queuing of multiple tapes for ejection. This obviously supports the StorageTek™ robot system but the command can be easily changed for DSI

```
ODT("TT TM ", SERIALNO, " EJECT")
```

The OPALs should be run by:

```
TT EVAL TPDB_EJECT DO TPDB_EJECT
```

Then, to confirm the movements off-site:

```
TP CONFIRM
```

## Appendix C: METATAPELIB DASDL

This listing corresponds to the source DASDL symbolic used by the INSTALL utility to create or update the METATAPELIB database.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 00001000
OPTIONS (AUDIT,STATISTICS,KEYCOMPARE,ADDRESSCHECK); 00002000
PARAMETERS (OVERLAYGOAL=5,ALLOWEDCORE=30000,SYNCPPOINT=1 TRANSACTIONS, 00008000
            CONTROLPOINT=1 SYNCPPOINTS ); 00011000
DEFAULTS (REBLOCK=TRUE, 00013000
PACK=DISK 00013001
,CHECKSUM=TRUE 00013101
,BUFFERS = 0 + 0 PER RANDOM USER OR 2 PER SERIAL USER 00016000
,ALPHA (INITIALVALUE BLANKS),BOOLEAN (INITIALVALUE FALSE), 00016100
NUMBER (INITIALVALUE 0)); 00016200
AUDIT TRAIL ATTRIBUTES (AREAS=100,AREASIZE=50 BLOCKS, 00028000
                        BLOCKSIZE=900 WORDS ,CHECKSUM=TRUE, 00029000
PACK=SPARE141 00029001
                        COPY TO TAPE 2 TIMES AND REMOVE); 00030000
00034000
CONTROL FILE ( 00034500
USERCODE=*,PACK=DISK 00034501
); 00034540
ACCESSROUTINES =*SYSTEM/ACCESSROUTINES ON 00034560
DISK 00034561
; 00034600
DMSUPPORT = 00034620
*DMSUPPORT/METATAPELIB4 ON DISK 00034621
; 00034660
RECONSTRUCT= 00034800
*RECONSTRUCT/METATAPELIB4 ON DISK; 00034801
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 00035000
%%% 00035500
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 00036000
%%% M E T A T A P E L I B %%% 00037000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 00038000
00039000
%***** 00040000
RDS RESTART DATA SET 00041005
%***** 00043000
00043200
AREAS=6 00043400
00043600
(DUMMYDEC ALPHA (6) ); 00044000
% 00062000
%----- E N D O F R E S T A R T A R E A S D A T A S E T ----- 00063000
% 00064000
% 00065000
%***** 00066000
MEDIA DATASET 00067000
%***** 00072000
00073000
POPULATION=50000,REBLOCKFACTOR=5,BLOCKSIZE=7 RECORDS 00073002
00073004
( 00073005
00073500
SERIALNO "SERIAL NUMBER" ALPHA (6); 00074000
00074500
CYCLE "CYCLE FILE ATTRIBUTE" FIELD (20); 00074600
SITENO "SITE SYSTEM SERIAL NUMBER" FIELD (20); 00074900
SPARE "SPARE FIELD" FIELD (8); 00075200
00075400
SENDDATE "DATE TAPE SENT OUT IN YYYYDDD" FIELD (24); 00075500

```



RETURNDATE	"DATE TAPE WILL BE BACK YYYYDDD"	FIELD(24);	00075800
REELNO	"REEL NUMBER FILE ATTRIBUTE"	FIELD(20);	00076100
ACCESSTIME	"TIME IN HHMMSSDD (DD=HUNDREDTHS)"	FIELD(28);	00076400
			00076700
			00077000
SENDTIME	"TIME TAPE SENT OUT IN HHMMSSDD"	FIELD(28);	00077300
ACCESSCOUNT	"NUMBER OF ACCESSES SINCE LAST CLEAN"	FIELD(20);	00077600
			00077900
TASKNO	"TASK NUMBER OF CREATING TASK"	FIELD(24);	00080000
JOBNO	"JOB NUMBER OF CREATING JOB"	FIELD(24);	00081000
			00081200
VERSION	"VERSION FILE ATTRIBUTE"	FIELD(8);	00082700
UNITNO	"UNIT NUMBER WHERE LAST WRITTEN"	FIELD(16);	00083000
CLEANDATE	"USER PROVIDED DATE OF LAST/NEXT CLEANING"	FIELD(24);	00083200
			00083300
SAVEFACTOR	"THE SAVEFACTOR FILE ATTRIBUTE"	FIELD(12);	00083500
RULE-INX	"INDEX OF RULE APPLYING"	NUMBER(2);	00083600
ASSIGNED-TIME	"TIME TAPE WAS ASSIGNED"	FIELD(28);	00083700
			00083800
STATES FIELD	(DESTROYED "VOLUME LIBRARY DESTROYED VOLUME";		00084200
	SCRATCH;		00084500
	VOLUMED "PRESENT IN VOLUME LIBRARY";		00084800
	CATALOGFILE "CATERGOGED ON THE TAPE FAMILY";		00085100
	PGNEVER "RETAIN THIS VOLUME FOREVER";		00085400
	PGAUTHORISED "PURGE HAS BEEN AUTHORISED";		00085700
	LOST;		00086000
	DAMAGED "DAMAGED OR SUSPECT";		00086300
	UNLABELLED "VOLUME IS AN UNLABELLED TAPE";		00086600
	REPORTED "PENDING ACTION HAS BEEN REPORTED";		00086900
	MANUAL "NOT TO BE PROCESSED BY RULES";		00087500
	SILOCART "HELD IN AN AUTOMATED LIBRARY";		00087800
	GENEXP "EXPIRED BY GEN BUT DAYS TO RUN";		00088100
	BOOL1 "SPARE";		00088400
	BOOL2 "SPARE";		00088700
	BOOL3 "SPARE";		00089000
	BOOL4 "SPARE";		00089300
	BOOL5 "SPARE";		00089600
	BOOL6 "SPARE";		00089650
	BOOL7 "SPARE"; %20 BITS		00089700
	);		00090200
			00090500
ACCESSDATE	"DATE IN YYYYDDD"	FIELD(24);	00090550
COMPRESSION	"EDRC OR HS COMPRESSION"	FIELD(4);	00090600
			00090620
PENDING-SLOT	"SLOT WHERE TAPE WILL BE STORED"	FIELD(20);	00090640
SPARE-1		FIELD(4);	00090645
SPARE-3		FIELD(8);	00090650
SPARE-4		FIELD(16);	00090660
			00090670
SPARE-2		FIELD(8);	00090720
SPARE-5		FIELD(16);	00090760
%Following items are numbers instead of fields to prevent null keys			00090763
%items.			00090764
ASSIGNED-DATE	"DATE TAPE WAS ASSIGNED"	NUMBER(7);	00090765
EXPIRY	"DATE TO EXPIRE - SET BY USER"	NUMBER(7);	00090766
SLOT	"SLOT NUMBER WHERE TAPE IS STORED"	NUMBER(6);	00090767
CREATIONTIME	"TIME IN HHMMSSDD (DD=HUNDREDTHS)"	NUMBER(8);	00090768
DENSITY	"800,1600,6250,38000,11000"	NUMBER(5);	00090769
CREATIONDATE	"DATE IN YYYYDDD"	NUMBER(7);	00090770
PENDING-DATE	"DATE OF PENDING LOCATION"	NUMBER(7);	00090772
ALPHA-1	"SPARE ALPHA FIELD"	ALPHA(17);	00090780
VOLUMEID	"FILE TITLE OF TAPE"	ALPHA(17);	00090800
FILEID	"SECOND LEVEL OF TAPE TITLE"	ALPHA(17);	00091100
LOCATION	"USER PROVIDED LOCATION OF MEDIA"	ALPHA(12);	00091150
PENDING-LOCATION	"PENDING LOCATION"	ALPHA(12);	00091200
NEXT-LOCATION	"LOCATION AFTER PENDING"	ALPHA(12);	00091250
BOXNUM	"BOX ID FOR BACKUP FACILITY"	ALPHA(6);	00091400
	% CAN GIVE RACK NUMBER, OR BACKUP SITE INFO		00091700
OWNER	"NAME OF CREATING USER"	ALPHA(17);	00092000

[illegible]

ID	"IDENTIFIER OF RULE"	ALPHA(17) ;	00156300
USER-PATTERN	"PATTERN TO MATCH USERCODE"	ALPHA(17) ;	00156500
VOL-PATTERN	"PATTERN TO MATCH VOLUME ID"	ALPHA(17) ;	00156600
FIL-PATTERN	"PATTERN TO MATCH FILE ID"	ALPHA(17) ;	00156700
HOST-PATTERN	"PATTERN TO MATCH HOST NAME"	ALPHA(17) ;	00156750
USER-EXAMPLE	"EXAMPLE OF USER TO MATCH PATTERN"	ALPHA(17) ;	00156800
VOL-EXAMPLE	"EXAMPLE OF VOLUME ID TO MATCH PATTERN"	ALPHA(17) ;	00156900
FIL-EXAMPLE	"EXAMPLE OF FILE ID TO MATCH PATTERN"	ALPHA(17) ;	00157000
HOST-EXAMPLE	"HOST NAME TO MATCH"	ALPHA(17) ;	00157200
LINK-USER	"USER OF TAPE SET THIS IS LINKED TO"	ALPHA(17) ;	00162100
LINK-ID	"ID OF TAPE SET THIS IS LINKED TO"	ALPHA(17) ;	00162200
);			00174000
RUL SET OF RULES KEY(USER, ID, INX) ;			00174100
SCRATCHTAPE SUBSET OF MEDIA WHERE (PGAUTHORISED OR SCRATCH)			00174200
KEY (SCRATCH-POOL, DENSITY, SERIALNO, ASSIGNED-DATE) DUPLICATES ;			00174300
REPORTS SUBSET OF MEDIA WHERE REPORTED BIT VECTOR ;			00174400
PENDING SUBSET OF MEDIA WHERE PENDING-DATE > 0			00174500
KEY (PENDING-LOCATION, LOCATION, PENDING-DATE, SLOT, SERIALNO)			00174600
DUPLICATES ;			00174700
BYNAME SUBSET OF MEDIA WHERE NOT SCRATCH AND NOT PGMAUTHORISED			00174720
KEY (OWNER, VOLUMEID, CREATIONDATE DESCENDING,			00174800
CREATIONTIME DESCENDING, FILEID, SERIALNO)			00174850
DUPLICATES FIRST ;			00174900
BYFAM SUBSET OF MEDIA WHERE NOT SCRATCH AND NOT PGMAUTHORISED			00175000
KEY (RULE-USER, RULE-ID,			00175050
CREATIONDATE DESCENDING, CREATIONTIME DESCENDING,			00175100
RULE-INX, SERIALNO)			00175150
DUPLICATES FIRST ;			00175200
BYLOCATION SET OF MEDIA KEY (LOCATION, SLOT, SERIALNO) DUPLICATES FIRST ;			00175300
BYTS SET OF MEDIA KEY (CREATIONDATE DESCENDING, CREATIONTIME DESCENDING,			00175320
SERIALNO)			00175330
DUPLICATES ;			00175340
LINKS SUBSET OF MEDIA WHERE LINK NEQ " "			00175400
KEY (LINK) DUPLICATES ;			00175500
POOL SET OF MEDIA KEY (SCRATCH-POOL, DENSITY, EXPIRY, SERIALNO) DUPLICATES ;			00175600

## Changes to DASDL

From time to time, changes are made to the METATAPELIB4 DASDL by Metalogic usually to change data set structure or set specifications. Because of the complexity of supporting DMSII reorganizations on different MCP and DMSII levels, updates to METATAPELIB4 are automatically handled by the Metalogic INSTALL utility.

When a structure change is detected, INSTALL initiates a DUMP of the TRIM database to flat files, installs the new release, updates the DASDL, regenerates the DMSII tailored software, reinitialises the databases and performs a reload.

Using the INSTALL utility, it is possible to change many run-time settings in the DASDL such as file locations, usercodes and audit specifications. Please refer to the **Metalogic INSTALL Reference manual** for more information on these topics.

# Index

- ?**
- ?ON SUPERVISOR command
    - getting started 27
- @**
- @ modifier
    - tape history 120
- <**
- <box> modifier
    - CONFIRM command 169
  - <density> modifier
    - FIND SCRATCH command 194
  - <keyword> modifier
    - HELP command 199
  - <location> modifier
    - CONFIRM command 169
  - <serialno> command
    - getting started 34
  - <unitno> modifier
    - use with TP CLEANED 123
  - <update action >
    - TP commands 120
  - <wildcard location>
    - with FIND LOCATION command 182
  - <wildcard name>
    - with FIND NAME command 184
- 2**
- 24 line limit 229
- 9**
- 999999 DAYS
    - permanent retention 64
    - permanent retention 85
- A**
- abbreviations of command keyword 104
  - ACCESSCODE field
    - holds previous volume id 161
  - ACCESSCOUNT attribute
    - use with FIND SCR TBC modifier
      - TP command 197
  - ACTIVE
    - TAPEMANAGER status 262
  - ADM events
    - EVENT PRINTLABEL 218
    - getting started 44
  - ALL modifier
    - for LOG command 204
  - Already assigned to NNNN
    - TAPEMANAGER log message 273
  - APPLY command 113, 135, 163
    - change of JOBNO 134
    - change of OWNER 143
    - changing CREATIONDATE 122
    - changing FILEID 132
    - changing JOBID 133
    - PGOK restrictions 91
  - ASSIGN command 164
  - ASSIGNDATE attribute
    - GETSCRATCH function 255
  - ASSIGNED attribute
    - GETSCRATCH function 255
  - assigned state
    - GETSCRATCH 254
  - assigned tapes 164
  - ASSIGNTIME attribute
    - GETSCRATCH function 255
  - attributes
    - HELP Command 222
    - printing 221
    - SYSTEM 221
  - audit tapes
    - linking to database dumps 82
  - authorisation phase 76–81
  - automatic purging 278
    - using OPAL programs 279
- B**
- B&L tape is not empty
    - TAPEMANAGER log message 274
  - back slash character 229
  - BNA networking 206
    - and TRIM 279–93
  - BNA networks 4
    - capabilities of slave TRIM systems 284
    - establishing a multi-host TRIM environment 281
    - NO FILE TLPORT error message 290
    - remote access of OPAL attributes 292
    - restart and recovery on TRIM slaves 290
    - TRIM in a multi-host environment 22
  - BOX modifier
    - TP command 106, 121
  - BOXNUM attribute 121
  - BYFAMILY set 221
  - BYLOCATION set 221
  - BYNAME set 221
  - BYTS set 221
- C**
- CALL DO mechanism 246

cancelling tape movements 145  
 CANDE  
   editing long OPALs 229  
 CATALOGING option 4  
   DESTROYED modifier  
     TP command 128  
   TRIM concepts 17  
   VERSION command 212  
 CDATE modifier  
   TP command 106, 122  
 changing MASTER to SLAVE  
   networking 289  
 changing OPAL programs  
   getting started 46  
 changing SLAVE to MASTER 285  
 CHECK\_TAPE\_ASSIGNMENT entripoint  
   TAPEMANAGER implementation 269  
 CHECK\_TAPE\_PURGE entripoint 267  
 checking presence of OPAL program 225  
 CLEANDATE attribute 123  
 CLEANED modifier  
   TP command 106, 123, 124  
 CLOSE command 113, 166  
 CLOSE modifier  
   LC command 202  
 closing METATAPELIB database  
   CLOSE command 166  
 communicating with SUPERVISOR  
   getting started 23  
 communicating with TRIM 103  
 COMS utility  
   getting started 25  
   setting up PRINTLABEL utility 301  
   setting up Supervisor windows 295  
 COMS windows 294–98  
   changing logon security 297  
   closing window dialog 297  
   DISPLAYs 218  
   getting started 27  
   logon rejected message 297  
   max stations allowed 295  
   processing of WINDOWS task 298  
   security considerations 296  
 CON modifier  
   for LOG command 204  
 condition to look for  
   SITUATION 214  
 configuration variables  
   SUP\_MAXWINDOWS 296  
   SUP\_WINDOWSECURITY 296  
   TL\_MASTER 283  
   TL\_MAXUSES 198  
   TL\_TAPELOG 213  
 CONFIRM command 110, 113, 125, 135,  
   164–69  
   errors and warnings 170  
   getting started 33  
   handling of PGNEVER tapes 146, 171  
   pending location warnings after PLOC use 149  
   reporting phase 162

  retrieving log messages 203  
   used with APPLY 163  
 contexts 234  
   description of 42  
   contracted synonym 105  
 CONTROLLER 237  
 creating or modifying an OPAL program 229  
 creating rules  
   TRIM utility 59  
 creating tapes  
   managing rules 55  
 creation date  
   changing with CDATE 122  
 CREATIONDATE attribute 122  
 CTIME modifier  
   TP command 106

## D

daily reports  
   and CONFIRM 168  
 DAMAGED attribute 123, 124  
 DAMAGED modifier  
   TP command 107, 124  
 DASDL changes  
   updates to METATAPELIB database 13  
 DASDL symbolic 112, 315–20  
 DASDL symbolic for METATAPELIB 317  
 DAT tape drives 10  
 database dumps  
   example TRIM rule 82  
 database not on-line  
   TAPEMANAGER log message 272  
 dates  
   US format 122, 130, 145, 152, 153, 212  
 DAYS retention 64  
 DB UPDATED' 67  
 DEFER modifier  
   TP command 107, 125, 151  
 DEFINE Command 223, 229  
 DEFINERESP option 226  
 defining rules  
   using job and task names for unlabelled tapes 54  
 definition of rules  
   managing rules 52  
 DELETE modifier  
   getting started 49  
   TP command 107, 126  
 deleting tapes  
   DELETE modifier  
     TP command 126  
 deleting tapes from METATAPELIB  
   getting started 49  
 DENSITY  
   GETSCRATCH function 252  
 DESTROYED attribute 128  
 DESTROYED modifier  
   TP command 107, 128  
 difference between WHEN and ONCE 241  
 DISPLAY

DEFINE Command 214  
 editing in CANDE 229  
 EVALUATE Command 232  
 invoking 239  
 listing dictionary 223  
 OPAL dictionary 218  
 parameters 239  
 stopping 241  
 WHEN Command 241  
 DISPLAY (/) Command  
   getting started with OPAL 239  
 DISPLAY Invocation 239  
 DISPLAYs  
   getting started with OPAL 218  
 DMSII dump 12  
 DO Command  
   getting started with OPAL 235  
   stopping 236  
   TEST 236, 237  
 DSI tape robot 254  
 DUMP command 113, 167, 172  
 dumping METATAPELIB database 11  
 Dup serial on SN  
   TAPEMANAGER log message 271

## E

ENTER command  
   OPALS/TPDB programs 277  
 ERR modifier  
   for LOG command 204  
 EVAL command  
   searching METATAPELIB database 221  
 EVALUATE command  
   debugging SITUATIONS 234  
   getting started with OPAL 232  
 EVALUATE Command 234  
   getting started with OPAL 232  
 EVT modifier  
   for LOG command 204  
 EXCEPTION rules  
   checking for usercoded rule 55  
   getting started 34  
   managing rules 77, 88  
 EXP command  
   synonym for FIND SCRATCH EXP 198  
 EXPIRED command 113  
 expired tapes 9, 162  
   automatic tape purging 276  
 EXPIRY attribute 129  
   use with FIND SCRATCH EXP 197  
 expiry dates  
   SAVEFACTOR attribute 55  
 EXPIRY modifier  
   TP command 107, 129  
 externally created OPALS 229

## F

FA SERIALNO command 256

FAMILYMANAGER utility 138  
 FAMILYMANAGER Utility 124  
 faster FIND searches  
   speeding up searches 186  
 FILEID  
   rule creation 61  
 fileid example 53  
 FILEID modifier  
   TP command 107, 132  
 fileid pattern 53  
 FIND command 172–98  
   200 entries limit 173  
   list of modifiers 173  
 FIND DATE command 113, 176–78  
   TODAY modifier  
     TP command 177  
   use of search filters 177  
   YESTERDAY modifier  
     TP command 177  
 FIND EXPIRED command 113  
 FIND FAMILY command 113, 179–80, 181  
   RULEINDEX attribute 69  
   testing and changing rules 91  
 FIND LINKED command 114, 181  
   similarity to FIND FAMILY 180  
 FIND LOCATION command 114, 182  
 FIND NAME command 114, 184–87  
   getting started 40  
 FIND PENDING command 114, 188  
 FIND POLL command 190  
 FIND RULES command 114, 190–93  
 FIND SCRATCH command 114, 194–98  
   getting started 39  
 FIND searches  
   speeding up searches 183  
 finding EXCEPTION tapes  
   managing rules 93  
 finding tapes by date 176  
 FLEX 137  
 FLEX Inquiry 124  
 FLEX package 141, 146, 147, 154, 212  
 FLEXLIB is off-line  
   TAPEMANAGER log message 273  
 FOR modifier  
   use with TP DELETE 49, 126  
 forward slash character 239  
 free form format 229

## G

generation retention 52  
 generation-based rules  
   effect of SR modifier  
     TP command 155  
 generations  
   definition of 64  
   getting started 37  
   using JOBNO to count 134  
 generations and locations  
   managing rules 74–75

GETSCRATCH function 252  
 TP ASSIGN 164  
 UNASSIGN modifier 159  
 GOINGAWAY  
 TAPEMANAGER status 262

## H

HELP command 114, 116, 199  
 communicating with TRIM 103  
 TRIM only 199  
 HELP Command  
 attributes 222  
 HOSTNAME  
 rule creation 61  
 HOSTNAME modifier  
 TP command 107

## I

INACTIVE  
 TAPEMANAGER status 262  
 indentation 229  
 information retrieval 214  
 INITIALIZING  
 TAPEMANAGER status 262  
 INSILO parameter  
 GETSCRATCH function 252, 256  
 installing TRIM 12  
 invoking a DISPLAY 239

## J

JOBID attribute 133  
 JOBID modifier  
 TP command 108, 133  
 JOBNO attribute  
 counting generations 134  
 JOBNO modifier  
 TP command 108, 134  
 JOBNUMBER attribute 140

## L

LC command 114, 202  
 LGR modifier  
 for LOG command 204  
 library of OPAL programs 214  
 LIBRARY/MAINTENANCE tapes 61  
 linked rules  
 FIND LINKED command 181  
 managing rules 82–83  
 linked tapes  
 in FIND FAMILY output 180  
 LINKING  
 TAPEMANAGER status 262  
 linking delays  
 managing rules 87  
 location itinerary

definition 66  
 LOCATION modifier  
 changes to Volume Library 142  
 TP command 108  
 TP command \r 135  
 location movements  
 examples of TRIM rules 70  
 LOG  
 TP command 117  
 LOG command 114  
 all modifiers **Error! Not a valid bookmark in entry on page 203**  
 getting started 37  
 using ERR for TAPEMANAGER messages 267  
 LOG CON command  
 CONFIRM errors and warnings 170  
 logging  
 CONFIRM errors and warnings 170  
 LC command 202  
 LOGREADER utility 4  
 how to run stand-alone 94  
 on slave systems 290  
 TRIM concepts 17  
 using FIND NAME for bad usercodes 186  
 LOST attribute  
 reset of 138  
 LOST modifier  
 TP command 108  
 TP command 138

## M

MAGUS library  
 TRIM concepts 16  
 manuals  
 Commonly used Attributes used in Supervisor 112  
 Metalogic INSTALL Reference 5, 12  
 Metalogic OPAL Language Reference 6  
 Metalogic OPAL Reference 279  
 Metalogic Supervisor Reference 1  
 Metalogic SUPERVISOR Reference 20, 49  
 MARC directive 20  
 overview 299  
 security considerations 300  
 TTINTERFACE codefile 299  
 Marked PGNEVER  
 TAPEMANAGER log message 272  
 master spo 25  
 MASTER unavailable  
 TAPEMANAGER log message 272  
 master-slave TRIM 206, 280, 281–84  
 MAXIMUM setting  
 USE WINDOWSECURITY 296  
 MCP notices 3  
 MERGETAPE utility 137  
 and PGNEVER tapes 146  
 NOTES attribute 141  
 MERGETAPE Utility 124  
 METATAPELIB DASDL 315–20  
 METATAPELIB DASDL symbolic 112

METATAPELIB database 3  
     maintaining tape history 17  
     TRIM concepts 17  
 METATAPELIB DMSII sets 221  
 METATAPELIB/DUMP/MEDIA  
     DUMP command 172  
 MINIMUM setting  
     USE WINDOWSECURITY 296  
 minus sign 236, 241  
 MIXID modifier  
     TP command 108, 139  
 MIXNO attribute 140  
 MIXNO modifier  
     TP command 109, 140  
 MIXNUMBER attribute 140  
 modifying rules  
     managing rules 71  
 modifying tape information 116  
 MOUNT\_REQUEST 255  
 MSG modifier  
     for LOG command 204  
 multi-host TRIM environment 279–93  
 multiple authorisations  
     managing rules 87

## N

NET command 114  
     all modifiers 206  
 network configuration 207  
 networking  
     and TRIM 280  
 NEVER modifier  
     TP command 136  
     use with CDATE 122  
     use with EXPIRY 129  
     use with PEND 145  
     use with RETURNDATE 152  
     use with SENDDATE 153  
 NEVER retention 66  
 new tape rules  
     TRIM concepts 15  
 NO FILE TLPORT message 290  
 no serials given  
     TAPEMANAGER log message 273  
 No serials given  
     TAPEMANAGER log message 273  
 Not PGOK  
     TAPEMANAGER log message 271  
 NOTE modifier  
     TP command 109  
     TP command 141  
 NOTES attribute  
     changes by MERGETAPE 141  
     effect of SR modifier  
     TP command 155

## O

ODT commands

PV MT 137  
 ODT modifier  
     for LOG command 204  
 ODTSequence  
     DEFINE Command 214  
     DO Command 235  
     editing in CANDE 229  
     EVALUATE Command 232  
     listing dictionary 223  
     ONCE Command 241  
     OPAL dictionary 217  
     parameters 235  
     results 235  
     stopping 236  
     TEST Command 237  
     unconditional execution 235  
     WHEN Command 241  
     without a controlling SITUATION 235  
 ODTSEQUENCE ALREADY RUNNING 235  
 ODTSequences  
     getting started with OPAL 217  
 OFFSITE  
     Volume Library location 19  
 OFFSITE location 137  
 offsite locations 52, 142  
 OFFSITE modifier  
     TP command 142  
 offsite tapes 137  
 ON SUPERVISOR command 295  
 ONCE Command  
     Getting started with OPAL 241  
 on-line help for TRIM 199  
 OPAL attributes 138  
     ACCESSCOUNT 197  
     BOXNUM 121  
     CLEANDATE 123  
     CREATIONDATE 122  
     DAMAGED 123, 124  
     DESTROYED 128  
     EXPIRY 129  
     getting started 42  
     JOBID 133  
     optimisation issues on slave systems 292  
     PENDINGDATE 129, 135, 145, 169  
     PENDINGLOCATION 149, 169  
     PGOK 129, 147  
     REPORTED 125, 151  
     RULEID 155  
     RULEOWNER 155  
     SENDDATE 135, 153  
     SENDTIME 135, 153  
     SYSNO 158  
     TASKID 139  
     VOLID 160  
 OPAL contexts 219  
     attributes 219  
     EVALUATE command 234  
     EVALUATE Command 232  
     PER 221  
     SYSTEM 221



OPAL Contexts  
     getting started with OPAL 219  
 OPAL dictionary  
     DISPLAYs 218  
     getting started with OPAL 215  
     ODTSequences 217  
     SITUATIONs 216  
 OPAL DISPLAY programs  
     getting started 44  
 OPAL identifiers 214  
 OPAL language  
     SHOW statement 231, 238  
 OPAL program  
     checking for presence 225  
     deleting 229  
     dictionaries 214  
     dictionary listing 223  
     TPDB\_FINDER example 227  
     TRIM concepts 15  
 OPAL programming language 4  
 OPAL programs  
     contexts 219  
     getting started 41  
     getting started with OPAL 227  
 OPALS/TPDB 243  
 OPALTAPELIB library  
     overview 18  
 OPEN command 114, 208  
 opening METATAPELIB database 208  
 operator interfaces 4  
 optimisation of OPAL attributes 292  
 options available within commands 103  
 OWNER modifier  
     TP command 109, 143

## P

parameters  
     to DISPLAYs 239  
 pattern  
     listing OPAL dictionary 225  
 pattern recognition 214  
 PDATE modifier  
     TP command 135  
 PEND modifier  
     TP command 109, 145  
 pending locations  
     changing with PLOC modifier  
     TP command 149  
 pending movement  
     TPDB\_REPORT 244  
 PENDING set 221  
 PENDING subcontext 244  
     mark ing tapes as REPORTED 151  
 PENDINGDATE attribute 145, 168  
     effect of EXPIRY modifier  
     TP command 129  
     effect of LOCATION change 135  
     use by FIND PENDING 188  
 PENDINGLOCATION attribute 149, 168

    use by FIND PENDING 188  
 PER context 41, 219  
     unit number list 235  
 PER Context  
     unit number list 239  
 permanent tapes 146  
     managing rules 85  
     purge protection 272  
 PG and SN commands  
     successful tape purges 274  
 PG and SN error messages  
     from TAPEMANAGER 271  
 PG command  
     TAPEMANAGER protection 260  
 PGNEVER attribute 64  
 PGNEVER modifier  
     TP command 109, 146  
 PGNEVER tapes  
     purge protection 272  
 PGOK attribute 129, 260  
     and FIND SCRATCH command 114  
     purge authorisation 147, 271  
 PGOK authorisation 8  
 PGOK modifier  
     backup references check 147  
     inhibited by PGNEVER 146  
     TP command 109  
     TP command 147  
 PLOC modifier  
     conflict with CONFIRM 170  
     TP command 110, 149  
 plus sign 226, 227, 229  
 POOL attribute  
     use with FIND SCRATCH POOL 197  
 POOL modifier  
     TP command 110  
 PORT modifier  
     NET command 206  
 previous volume id 161  
 PRINT modifier  
     getting started 40  
 printing tape labels  
     getting started 44  
     PRINTLABEL utility 301  
 printing TP output 40  
 PRINTLABEL utility  
     COMS batch file example 301  
     for remote printers 301  
     modifying the codefile 303  
 purge authorisation 147  
     auto-purging 9  
 purge protection 260, 267–75  
     on non-TAPEMANAGER systems 292  
 PURGE REQUEST REJECTED BY  
     TAPEMANAGER 267

## Q

question mark 226, 229

**R**

railroad diagrams 103  
 rebuild reinitialized database 172  
 rejected tape PG or SN 268  
 remote access of OPAL attributes 292  
 REMOTESPO command  
   getting started 27  
 RemoteSpos  
   getting started 27  
 report formatting 214  
 REPORTED attribute 125, 151  
   changed by PEND modifier  
     TP command 145  
   reset by PLOC modifier  
     TP command 149  
   used by CONFIRM 168  
 REPORTED modifier  
   TP command 110, 151  
 reporting and rule handling  
   list of 113  
 restarting Supervisor  
   TT QUIT SUPERVISOR 13  
 retention by generation 64  
 retention by SAVE and NEVER 66  
 retention criteria  
   generations and locations 74–75  
 retention information  
   managing rules 64  
 retention times  
   maximum values 75  
 RETURN modifier  
   TP command 110, 152  
 robotic tape systems *See* silo support  
 rule commands 162  
 rule creation  
   managing rules 59–67  
 rule family 7  
   managing rules 52  
 rule identification 58  
 rule identifier 50, 53  
 rule identity  
   TRIM utility Home Menu screen 58  
 rule index 53  
   TRIM utility Home Menu screen 58  
 rule interrogation  
   TRIM utility 68  
 rule maintenance  
   TRIM utility 56–71  
 rule matching 8, 55  
   using wildcards 63  
 rule modification  
   managing rules 71  
 rule searching  
   managing rules 88  
 rule user 53  
 rule usercode  
   TRIM utility Home Menu screen 58  
 RULEID attribute 155  
 RULEINDEX attribute 155

effect on rule families 69  
 rules  
   creation of 59–67  
   definition 7  
   definition of 52  
   EXCEPTION rules 88  
   features of 3  
   interrogation of 68, 192  
   linking 3, 82–83  
   management of 49–56  
   practical examples 72–73  
   unlabelled tapes 54  
   wildcard patterns 63  
 rules testing  
   managing rules 90–93  
 RULEUSER attribute 155  
 running the TRIM utility 56  
 run-time options 210

**S**

SAVE retention 66  
 SCHEDULE 13  
   OPAL dictionary 214  
 SCR command  
   synonym for FIND SCRATCH 198  
 SCRATCH command 115  
   getting started 39  
 SCRATCH POOL location 129  
   changed by APPLY 163  
   CONFIRM and PGOK 169  
 scratch pools  
   in the ideal TRIM environment 9  
 SCRATCH set 221  
 SCRATCHPOOL attribute  
   GETSCRATCH function 252  
   use by FIND POOL 190  
 security  
   log-on security 296  
   protecting rules 51  
   Tape Librarian usercode 59  
 semicolons  
   ODTSequence statements 217  
 SEND modifier  
   TP command 110, 153  
 SENDDATE attribute 153, 254  
   changed by LOCATION modifier  
     TP command 135  
   changed by OFFSITE 142  
 SENDTIME attribute 153, 254  
   change by LOCATION modifier  
     TP command 135  
   change by OFFSITE 142  
 SER command 115  
   synonym for FIND NAME187  
 serial number commands 118  
 serial number list  
   TAPEDB Context parameter 235, 239  
   TAPELABEL Context parameter 235, 239

- 
- SERIAL NUMBER REJECTED BY
    - TAPEMANAGER 267
  - serial numbers
    - uniqueness 9
  - SHOW command 115, 209
  - SHOW statement
    - OPAL language 231
  - silo support
    - TRIM interface 303–15
  - SITE location 137
  - SITE modifier
    - TP command 154
  - SITUATION
    - contexts 234
    - DEFINE Command 214
    - editing in CANDE 229
    - EVALUATE Command 232
    - getting started with OPAL 216, 227
    - listing dictionary 223
    - OPAL dictionary 216, 225
    - results returned 234
    - stopping 241
    - WHEN Command 241
  - SITUATION linked to multiple DISPLAYs 241
  - slave TRIM system
    - capabilities 283
  - slave TRIM systems 22, 285–89
    - restart and recovery 290
    - VERSION command 212
  - SM <mixno> ODT command
    - getting started 25
  - SN command
    - TAPEMANAGER protection 260
  - SN considerations 274
  - SN-ing tapes
    - marking tapes as DESTROYED 275
    - unlabelled tape handling 275
  - SR command
    - rule modify 71
  - SR modifier
    - changing CREATIONDATE 122
    - changing FILEID 132
    - changing JOBID 133
    - changing OWNER 143
    - changing VOLID 160
    - getting started 35
    - testing rules 90
    - TP command 110, 155
  - starting TAPEMANAGER 261
  - StorageTek™ *See* silo support
  - StorageTek™ tape robot 254
  - strategy
    - building a rule family 98
    - building a tape library 94
    - caring for TAPELOG 100
    - deimplementation of old systems 100
    - EXCEPTION rules 99
    - management of rules 93
    - modifying rules 100
    - new rules and EXCEPTION tapes 96
    - regular CONFIRMs 101
    - reporting and CONFIRM 99
    - review tape expiration 96
    - using rules linking 100
  - successful PG and SN operations 274
  - SUMLOG
    - tape notices 17
    - use during TRIM recovery 19
  - SUP\_MAXWINDOWS configuration variable 296
  - SUP\_WINDOWSECURITY configuration variable 296
  - SUPERVISOR
    - program overview 14
    - restarting 47
    - TRIM concepts 14
  - SUPERVISOR commands
    - synonyms 103
  - Supervisor COMS windows 294–98
  - switching systems
    - NET command 207
  - switching TRIM slave systems 285–89
  - synonyms
    - for commands 103
  - synonyms for commands 104
  - SYSNO attribute 158
  - SYSNO modifier
    - TP command 111, 158
  - SYSOPS command
    - TAPEMANAGER option 261
  - SYSOPS option
    - temporary disabling of 266
  - SYSOPS TAPEMANAGER 278
  - system command
    - MSG 237
  - system commands
    - ODTSequences 217
    - TEST command 237
  - system console 218
- T**
- tape assignment by serial number 269
  - tape creation
    - managing rules 55
  - tape drives
    - non-Unisys 10
  - tape file identifier 7
  - tape has backup refs
    - TAPEMANAGER log message 273
  - tape history
    - getting started 37
    - interrogation of 120
  - tape interrogation
    - displaying information 118
  - tape label
    - printing
      - getting started 44
  - tape labels
    - printing 3
    - printing from a WHEN 44

- 
- Supervisor restarts 14
  - TAPE LIBRARIAN usercod 172
  - Tape Librarian usercode 4
    - deleting tapes 126
    - from VERSION command 212
    - getting started 48
    - rule security 59
  - tape modification commands
    - list of 106
  - tape movements
    - authorisation phase 76–81
    - getting started 36
  - tape notices
    - TRIM concepts 14
  - tape owner 7
  - tape purging
    - criteria 279
  - tape robot systems
    - assigned tapes 165
  - tape serial number list
    - TAPENAMESERIALS 258
  - tape volume identifier 7
  - TAPEDB context 41, 219
    - serial number list 235
  - TAPEDB Context
    - serial number list 239
  - TAPEDB events
    - getting started 30
  - TAPEFAMILYSERIALS attribute 259
  - TAPELABEL context 41, 219
    - serial number list 235
  - TAPELABEL Context
    - serial number list 239
  - TAPELABEL events
    - getting started 28
  - TAPELIB option
    - getting started 31
    - setting after installation 13
    - setting on slave systems 283
    - TRIM concepts 15
  - TAPELIBUPDATER process
    - failure on slave systems 290
    - TRIM concepts 15
  - TAPELOG file 114
    - closed by LC command 202
    - CONFIRM errors 170
    - creating a new TAPELOG 202
    - displaying name from VERSION command 212
    - LOG command 114, **Error! Not a valid bookmark in entry on page 203**
    - logging of TP commands 116
    - overview 19
  - TAPEMANAGER 9
    - chacking volume ids 161
    - implementation 260
  - TAPEMANAGER implementation 3
    - and PGOK flag 147
    - DENY messages in TAPELOG 268
    - in the ideal TRIM environment 9
    - initiating TAPEMANAGER 261
    - installation and operations interface 261
    - interrogating status 262
    - purge protection 267–75
    - purge request rejected 267
    - SEND TM command 261
    - SEND TM QUIT command 264
    - SEND TM STATUS command 262
    - SN-ing tapes 275
    - successful PG and SN operations 274
    - temporary disabling of 266
    - terminating the TAPEMANAGER library 264
    - TPDB\_AUTOPG programs 276
    - WAITING FOR TAPEMANAGER message 264
  - TAPEMANAGER library
    - DS-ing the library stack 264
    - program overview 18
  - TAPEMANAGER option
    - SYSOPS command 261
  - TAPENAMEFAMILY attribute 259
  - TAPENAMESERIALS attribute
    - use by TPDB\_SILOW 258
  - TAPERECORD statement 250
  - TASKID attribute 139
  - TBC command 115
    - synonym for FIND SCRATCH TBC 198
  - TCPIP networking 206
    - and TRIM 279–93
  - TERM command 126
  - TERM FIRSTLINE
    - TT responses
      - getting started 24
  - terminate an ODTSequence 236
  - TERMINATING
    - TAPEMANAGER status 262
  - TEST (DO) command
    - getting started with OPAL 237
  - TEST command
    - getting started with OPAL 237
  - TEST FUNCTION message
    - getting started 23
  - testing and changing rules
    - managing rules 90–93
  - testing rules
    - CONFIRM and SR 92
    - using DEFER modifier
      - TP command 125
  - THE EVALUATE WILL BE DONE message 234
  - THE WHEN WILL BE DEACTIVATED 241
  - time retention 52
  - TL\_MASTER configuration variable 283
  - TL\_MAXUSES configuration variable
    - used for tape cleaning 198
  - TL\_TAPELOG configuration variable
    - TAPELOG file name 213
  - TLOFFLINE attribute 250
  - TLPORT error handling 290
  - TODAY command 115
    - getting started 33
  - TODAY modifier
    - FIND DATE command 177

- 
- TP <serial> command
    - TP commands 118
  - TP APPLY
    - rule modify 71
  - TP ASSIGN
    - list of assigned tapes 159
  - TP commands
    - <serial> 118
    - <serialno> modifier
      - TP command 34
    - APPLY 113, 122, 132, 135, 143, 163
    - ASSIGN 164
    - BOX modifier
      - TP command 106, 121
    - CDATE modifier
      - TP command 106, 122
    - CLEANED modifier
      - TP command 106, 123, 124
    - CLOSE 113, 166
    - CONFIRM 33, 113, 125, 135, 146, 164–69
    - CTIME modifier
      - TP command 106
    - DAMAGED modifier
      - TP command 107, 124
    - DEFER modifier
      - TP command 107, 125, 151
    - DELETE modifier
      - TP command 107, 126
    - DESTROYED modifier 107
      - TP command 128
    - DUMP 113, 172
    - EXPIRED 113
    - EXPIRY modifier
      - TP command 129
    - FILEID modifier 107
      - TP command 132
    - FIND DATE 113, 176–78
    - FIND EXPIRED 113
    - FIND FAMILY 113, 179–80
    - FIND LINKED 114, 181
    - FIND LOCATION 114, 182
    - FIND NAME 114, 184–87
    - FIND NAME 40
    - FIND PENDING 114, 188
    - FIND POOL 190
    - FIND RULE 114
    - FIND RULES 190–93
    - FIND SCRATCH 39, 114, 194–98
      - getting started 32
    - HELP 103, 114, 116, 199
    - HOSTNAME modifier
      - TP command 107
    - JOBID modifier
      - TP command 108
    - JOBNO modifier
      - TP command 108, 134
    - LC 114, 202
    - LOCATION modifier
      - TP command 108
      - TP command \r 135
    - LOG 37, 114, 203
    - LOST modifier
      - TP command 108, 138
    - MIXID modifier
      - TP command 108, 139
    - MIXNO modifier
      - TP command 109, 140
    - NET 114, 206
    - NEVER modifier
      - TP command 136
    - NOTE modifier
      - TP command 109, 141
    - OFFSITE modifier
      - TP command 142
    - OPEN 114, 208
    - OWNER modifier
      - TP command 109, 143
    - PDATE modifier
      - TP command 136
    - PEND modifier
      - TP command 109, 145
    - PGNEVER modifier
      - TP command 109, 146
    - PGOK modifier
      - TP command 109, 146, 147
    - PLOC modifier
      - TP command 110, 149
    - POOL modifier
      - TP command 110
    - REPORTED modifier
      - TP command 110, 151
    - RETURN modifier
      - TP command 110, 152
    - SCRATCH 39, 115. *See* FIND SCRATCH
    - SEND modifier
      - TP command 110, 153
    - SER command 115
    - SHOW 115, 209
    - SITE modifier
      - TP command 154
    - SR modifier
      - TP command 35, 110, 122, 132, 133, 143, 155
    - SYSNO modifier
      - TP command 111, 158
    - TAPELIBUPDATER process 15
    - TBC 115
    - TODAY 33
    - UNASSIGN modifier
      - TP command 111, 159
    - VER 32
    - VERSION 115, 210
    - VOLID modifier
      - TP command 111, 160
    - TP directive
      - MARC interface 300
    - TP DUMP 167
    - TP VER 207
    - TP\_PERMT programs
      - remote access of OPAL attributes 292
    - TP\_PERRECORD 250

- 
- TP\_PERRECORD programs  
     building a tape library 94  
 TPDB\_AUTOPG 243, 278  
 TPDB\_AUTOPG programs  
     auto purging 276  
 TPDB\_BYFAMILY 248  
 TPDB\_BYLOCATION 248  
 TPDB\_DAILYREPORTS 243  
 TPDB\_DAILYREPORTS programs  
     authorisation phase 76  
 TPDB\_DEFAULTS 247  
 TPDB\_EJECT 256  
 TPDB\_FINDER program  
     getting started with OPAL 227  
 TPDB\_LABEL 243  
 TPDB\_LABEL programs  
     getting started 44  
     using the PRINTLABEL utility 302  
 TPDB\_REPFOOT 247  
 TPDB\_REPHEAD 247  
 TPDB\_REPORT 244  
 TPDB\_REPORT programs  
     use with CONFIRM 168  
 TPDB\_REPSETUP 246  
 TPDB\_SILOW 256  
 TRIM  
     and full Supervisor 2  
     introduction 1  
 TRIM recovery  
     DMSII considerations 21  
     LOGREADER utility 21  
     missed tape notices 21  
 TRIM slave systems *See* slave TRIM systems  
 TRIM software  
     and BNA networks 22  
     files and codefiles 14  
 TRIM utility 56–71  
     adding or creating rules 59–67  
     Home Menu screen 57  
     interrogating rules 68  
     modifying rules 71  
     RULES set and dataset 50  
     TRIM concepts 18  
 TT directive  
     MARC interface 300  
 TT DO TPDB\_DAILYREPORTS 168  
 TT prefix  
     from ODTs and COMS 20  
     getting started 23  
 TT SO 13  
 TTINTERFACE library  
     user interfaces 20
- U**
- UL\_rules 54  
 UL\_rules  
     managing rules 84  
 UNASSIGN command 255  
 UNASSIGN modifier  
     TP command 111, 159  
 underscore  
     in identifiers 214  
 unique name  
     for OPAL programs 214  
 unit number list  
     PER Context parameter 235, 239  
 UNKNOWN usercode  
     correcting wrongly usercoded tapes 187  
 unlabelled tape rules  
     managing rules 84, 85  
 unlabelled tapes  
     defining rules for 54  
     rule matching 55  
     task and job patterns 54  
     use of JOBID attribute 133  
     use of TASKID attribute 139  
 UNOBTAINABLE  
     tape owner 251  
 updating rules  
     implications for rule families 71  
 USDATES option 122, 130, 145, 152, 153, 212  
     date display formats 31  
 USE command  
     getting started 25  
     USE WINDOWSECURITY 297  
 USE DESTINATION  
     print daily reports 76  
 USE WINDOWSECURITY 297  
 user interfaces 20, 294, 293–303  
     TTINTERFACE library 20  
 usercode  
     rule owner 50  
 USERCODE  
     rule creation 61  
 usercode example 53  
 usercode pattern 53  
 usercoded rules  
     EXCEPTION rules 88  
     searching criteria 88
- V**
- VERSION command 115, 210  
     getting started 32  
 version information 210  
 VOLID  
     rule creation 61  
 VOLID attribute 160  
 VOLID modifier  
     TP command 111, 160  
 Voluemid not matched  
     TAPEMANAGER log message 271  
 VOLUME DESTROYED command 128  
 volume identifier 61  
 Volume Library  
     and TRIM software 19  
     OFFSITE flag 137  
 volume not in DB  
     TAPEMANAGER log message 272

volumeid example 53  
volumeid pattern 53

## W

WAIT TLREC  
    WHEN status 292  
**WAITING FOR TAPEMANAGER** 262  
WAITING TO DELINK  
    TAPEMANAGER status 262  
WFL VOLUME command  
    OFFSITE and ONSITE 137  
WHEN command  
    using OPAL programs 219  
WHEN Command  
    getting started with OPAL 241  
    Getting started with OPAL 241  
wildcard

    patterns for building rules 63  
wildcard characters  
    definitions 175  
wildcard patterns  
    with FIND command 173  
wildcards  
    listing OPAL dictionary 225  
WINDOWS command 298  
WS command  
    getting started 25

## Y

YEARS retention 64  
YESTERDAY modifier  
    FIND DATE command 177