Total Tape Solution User Guide Contents

Other Docs

Known Problems

Total Tape Solution User Guide

Other Docs

Known Problems

Overview

This chapter contains an overview of the ProMAX tape system and its infrastructure. It describes tape system components, their features, and overall functionality.

About this Document

This *Overview* provides an overview of TTS and its infrastructure. ProMAX users should read this overview, *Using the Tape System*, and *Using the ShowCat Catalog*.

The *QuickStart Guide* describes minimal configuration required for tape functionality equivalent to previous releases.

The *Usage* section describes:

- *Using the Tape System* explains aspects of using the tape system within ProMAX.
- *Using the Showcat Catalog* describes the interface, making queries, and gives example of the user interface for the catalog system.
- *Using the OpCon Operator Console* is written for ProMAX tape operators in a data center or ProMAX users who mount their own tapes.

The *TopCat Guide* provides installation and setup instructions as well as instructions for use.

The *PMX Guide* provides installation and setup instructions as well as instructions for use. It also includes *Using TapeDiag Tape Diagnostics* which describes the tapediag program, an optional command-line utility for testing tape performance, labeling blank tapes, and displaying tape labels.

The *Reference Section* contains a *Glossary*, *About ProClient* (ProClient is not needed by most users), *Tape library Robotics*, and detailed information about *TTS files*.

Troubleshooting provides help for trouble with the tape system.

The *ProMAX Release Notes* contain a Known Problems section describing possible workarounds and potential upgrade enhancements.

About TTS

The **Total Tape Solution**, or **TTS**, is a comprehensive and scalable solution for managing tape datasets and tape devices with ProMAX.

TTS behaves very much like previous versions of the ProMAX tape system. However, TTS differs from the previous tape system by offering:

- enhanced capabilities
- more configurable flexibility
- modularity of design

ProMAX now passes information as *out-of-band data* rather than *in-band*, from the job packet file through the master executive and on to the tape I/O process.

Before you begin the TTS installation process, we suggest that you:

- Build a team. Installing TTS can involve three different ProMAX personnel: the system administrator, tape operator or computer operator, and a user.
- Take inventory of your equipment. You need to know about your local and remote tape devices, tape catalogs, and pooled device management.

TTS Components

TTS is a collection of the following five components made up of binary executables, shell scripts, configuration files, and, Java libraries:

- **ProClient Tape Management System**. This component can be used alone in a simple installation to communicate with local and/or remote tape devices.
- **TopCat Tape Catalog Media Manager.** This component is used in a complex installation involving a tape catalog and pooled device management. It is the tape catalog manager.
- **ShowCat Catalog Graphical User Interface**. This component is the user interface for TopCat.

- **PMX Device Management System** This component is also used in a complex installation involving a tape catalog and pooled device management. It is the tape catalog manager. It handles device allocation issues for tape jobs.
- **OpCon Graphical Operator Console** This component is the user interface for the PMX device management system.

Each component and its features are described below.

ProClient Tape Management System

ProClient is the tape management system. It is the shell script executed by ProMAX and in turn executes a *proclient* binary executable.

The ***proclient*** binary is the actual tape I/O client that handles all the reading and writing of data. By the time it is executed, it knows about the physical device name, device type, label type, and other operating parameters for the I/O process. It communicates with the executive through a standardized tape command protocol via a client-server socket interface. If the tape device is physically attached to the host where the ProMAX job (and executive) is running, data I/O is done via fast UNIX sockets; if the tape device is remotely attached, INET sockets are used.

All tape configuration settings and dataset information are sent to ProClient to handle communication with tape devices.

Full Label Tape Processing

ProClient provides full runtime interpretation of Standard (SL), ANSI or ASCII (AL) or Non (NL) tape labels. Label tape processing provides added security by minimizing the risk of overwriting or reading a wrong tape. Because the tape system physically stores attributes such as volume serial number, tape owner, dataset name, and expiration date the tape system can interpret this information.

Configurable Tape System Runtime Behavior

In TTS, all tape I/O is handled by ProClient regardless of the job’s device requirement. Many runtime properties can be configured by associating a property with a device in the device.tts or PMX database. Some of the runtime properties include:

- reading labels; how and what information to bypass
- running with or without the TopCat catalog
- using a mount message prior to tape I/O

Full Tape Catalog Integration

The ProClient script and the ***proclient*** executable are integrated with the optional new ProMAX Tape Catalog, **TopCat**. During runtime, they communicate with the tape catalog to allocate volumes and open/close datasets.

Improved Tape I/O Performance

Block size is now configured to take advantage of different tape drives. This improves tape I/O performance.

Improved Robotic Tape Library Support

TTS fully integrates the new ProMAX tape system, the new ProMAX tape catalog, *and* robotic tape libraries to automate tape retrieval. Currently, TTS supports ***StorageTek robotic tape libraries***.

TopCat Tape Catalog media manager

TopCat is the tape catalog media manager. It is comprised of a topcat binary executable.

Full Catalog Data Management

TopCat supports full catalog management of native ProMAX data, various field data, and archival datasets. You can import other catalogs (including old tcat catalogs) into TopCat. You can also have multiple secondary tape catalogs to help you logically catalog tapes by data type, such as workgroup or project.

Standard ProMAX Tape Catalog Interface

TopCat can be executed from other programs, shell scripts, or graphical UIs because it:

- accepts command line options
- has a command line processor

- outputs information to standard output.

Improved Tracking Of Dataset and Runtimes

TopCat tracks *85* attributes for each dataset. It tracks volume and dataset information, date attributes, and runtime information that is written to tape. TopCat also stores global survey information

Extensive Catalog Edit/Query/Report Functions

The ***topcat*** executable accepts a wide range of command-line options. These interactive options are used to edit and modify catalog contents; customize displayed information based on a catalog attribute. Other command-line options are also available.

ShowCat Catalog Query Graphical User Interface

ShowCat is the graphical user interface to the TopCat catalog program. It runs as a ***showcat*** shell script; starting a Java graphical user interface. ShowCat runs TopCat to query, edit, and display catalog contents.

Interactive Catalog Query/Display

ShowCat allows you to interactively query and display TopCat catalog entries. You can display information based on a catalog attribute.

With TTS administrator privileges, you can use the ShowCat UI to interactively edit catalog fields to update or correct values.

Extensive Catalog Query/Report Functions

ShowCat makes use of topcat’s extensive querying functions. You can construct complex queries based on Boolean notation, ranges of values, string patterns, or date comparisons. Catalog attributes matching criteria is displayed for the tape volumes. Commonly used search queries can be stored. Since all query/edit/display functions performed by ShowCat are actually executed by topcat, these same search queries can also be incorporated into shell scripts for command line execution.

PMX Resource Manager

The PMX Resource Manager is a collection of six binary executables: ***pmxDev***, ***pmxInit***, ***pmxRls***, ***pmxRsv***, ***pmxSet***, ***pmxMov***. These executables handle device allocation issues for tape jobs.

Simplified Device Selection

The PMX Resource Manager allows you to define pools of tape devices, organized by device type. With device pools, any device in the pool is used. You just request a device pool based on your device type. However, if you choose you can mount on a specific drive.

Drives are selected based on availability and you are provided a mount message telling you what drive is being used. Tape drives can also be pooled together based on workgroup or location.

Device pools also work in conjunction with **TopCat** tape catalog *media* pools. Media pools allow you to group tapes together based on media/device type. Currently, there can exist a *many to one* mapping between a device pool and a media pool but not a *one to many*. In other words, the same media pool can have many device pools associated with it but a single device pool cannot be associated with more than one media pool.

Resource Overallocation Prevention

The PMX Resource Manager queues and releases tape jobs as they are submitted. This prevents overallocation and potential deadlock situations. The number of available tape drives control the number and type of tape jobs released for execution. A priority system gives queued tape jobs higher priority over others.

OpCon Operator Console Graphical User Interface

OpCon is the graphical interface to the PMX Resource Manager. It is used to display information about device pooling and allocation. Like the ShowCat interface, it runs as a *opcon* shell script. This script starts up a Java graphical user interface.

Centralized Dispatching/Monitoring Of Tape Mount Requests

OpCon gives a centralized view of all network tape devices in use by ProMAX. This is especially useful for large data centers. Device allocation, mount queue management, and tape drive monitoring functions are all available from this one interface; you can cancel a pending tape request, kill an active tape request, or change queue priority ratings. OpCon tracks job status information such as, how much data has been read/written and if a job was completed, cancelled, or killed.

OpCon is optional. The old **Tape Help** mount request window is still used if you do not use OpCon.

Centralized Device Allocation Monitoring

OpCon centrally displays all information about device pooling and allocation. It provides the following information:

- which tape devices are grouped into which device pools
- which drives are currently used/available
- which tape request jobs are active/queued

You can also run the **PMX Resource Manager** from the command line to receive the same information.

With OpCon, you can toggle a device **offline** to reserve it for other jobs or if it is temporarily unavailable, for example, due to maintenance.

TapeDiag Tape Diagnostics Utility

TapeDiag is a command line utility program. It contains a ***tapediag*** binary executable with a set of shell script wrappers. It is run independent of ProMAX.

Tapes Prelabeled and Tape Problems Diagnosed

TapeDiag allows you to prelabel tapes, diagnose tape problems, and determine performance boundaries. Depending on how ProClient is configured, a tape output step expects a labeled tape for comparing label information with the volume information it is expecting. You can prelabel a tape as either AL or SL format.

You can also perform fixed/variable block I/O tests to determine the block size your tape drives successfully support. It also performs multi-file or multi-volume read/write verification tests, tape file positioning tests, and I/O performance tests.

About TTS Infrastructure

When you execute a job, a new superExec program executes a ***$PROMAX_HOME/port/bin/reserveDevice*** script. This script invokes the PMX Resource Manager.

Device Pools

The PMX Resource Manager takes the device pool(s) that you have selected and allocates specific tape devices in those device pool(s) based on availability. If a tape device is available, it is reserved exclusively for that one processing step using the device reservation program **pmxRsv.** If all the tape devices are in use, the tape request is queued, preventing the ProMAX job from proceeding until necessary drive(s) in the selected device pool become available.

Runtime Directory

Every time you execute a flow, a Runtime Directory is created in the ProMAX flow directory. It has a UNIX process ID (PID) number for a name, typically three to six digits, for example: ***$PROMAX_DATA_HOME/area_name/line_name/flow_name/47238/***. The Runtime Directory contains the out-of-band data files. For example, ***catEnv*** files to set environment variables used by the client shell script.

When the requested tape devices are available, the superExec calls the ProMAX executive, ***$PROMAX_HOME/sys/exe/exec.exe***. Depending upon the flow, superExec can all more than one executive.

Shell Scripts

The executive runs a ***$PROMAX_HOME/sys/exe/ProClient*** shell script for each tape I/O step in the flow. The ProClient shell script is run on the host where the requested tape device is physically attached. The script sets some configurable environment variables that affect the tape system's runtime behavior and executes the ***$PROMAX_HOME/sys/exe/proclient*** binary executable.

proclient Executable

While the ***proclient*** executable is running, it communicates with the OpCon Operator Console by sending it Java messages.

- ***proclient*** issues tape mount requests by running the script, ***$PROMAX_HOME/port/bin/TAPE_REQUEST***.

  TAPE_REQUEST takes arguments passed by ***proclient***, and passes a mount request message to OpCon.

  TAPE_REQUEST has a *blocking* action. This means that after it passes a tape mount request, ***proclient*** waits and does not continue until the tape mount is fulfilled.

- ***proclient*** updates job status information by running the script, ***$PROMAX_HOME/port/bin/TAPE_NOTIFICATION***.

  TAPE_NOTIFICATION updates information such as how much data was read/written, and whether a tape process completed, was cancelled, or had an error.

  TAPE_NOTIFICATION has *non-blocking* action. This means that ***proclient*** continues to run while it sends out tape notification messages.

Job Cleanup

While the tape job is running, the TopCat tape catalog is updated as each VSN is written. Upon job completion, the dataset is closed. The superExec again invokes the PMX Resource Manager, this time to release the drives, using the PMX Resource Manager’s device release program, **pmxRls**.

Other Docs Known Problems

QuickStart Guide

In this chapter we will explain the steps for setting up and using basic tape configurations, along with description of common usage scenarios.

Basic Configuration

This section describes the basic requirements for configuring TTS and making tape devices accessable to ProMAX. We describe the required settings and recommend options. To select the most appropriate settings for your site, read the remainder of the TTS documentation.

Defining and Initializing Tape Devices for Linux

The following two steps are required for Linux operating systems:

1. Build the /etc/stinit.def file to define tape drive and their characteristics to the system.

   In the example below, the manufacturer and model number are obtained from the **dmesg** command. The absence of /sbin/stinit shows that this package is not installed on the systems you are looking at. This package is not installed by default when loading the Red Hat 7.2 software. You can determine which package a file belongs to by using the follow **rpm** command:

   ```
   rpm --query -f (filename)

   rpm --query -f /sbin/stinit

   > mt-st-0.6-1
   ```

   Once the mt-st-0.6-1 package is installed, you can find stinit.def examples and documentation in the /usr/share/doc/mt-st-0.6-1 directory. You can also type **info stinit** to get this information.

2. Run the **/sbin/stinit** command to initialize the appropriate /dev/nst? tape devices.

These steps initialize the appropriate tape devices:

```
su to root

cd to /etc

type stinit

test device by typing, for example, `mt -f /dev/nst0 status`
```

Setting up the device.tts file

All tape devices accessed by ProMAX jobs must be described in the **device.tts** file. This text file sets the behavioral characteristics of tape devices. It is similar to the **device.txt** file used in earlier versions of ProMAX.

New ProMAX installations

You should copy $PROMAX_HOME/etc/device.tts_example to a file named device.tts and edit the file as described in *Required entries for device.tts.*

Upgraded ProMAX Installations

Installations running previous versions of ProMAX should convert their existing device.txt file(s) to device.tts. Executing the following script, prompts for the locations of your old device.txt file and your new device.tts file:

```
$PROMAX_HOME/port/bin/convert_device.txt2tts
```

Once you verify the resultant file, you should have identical device behavior as before. However, there are a number of new capabilities provided for your use. These are described in ***ProClient Command Arguments***.

TTS allows use of personal copies of the device.tts file, as we did for device.txt. As administrator, you can take this opportunity to clean-up and synchronize these files with the files found in the users' home directories.

Required entries in device.tts

Each device used by ProMAX must be defined in the device.tts file; all descriptors for a device must be contained on the same line (wrap-around is allowed, but not carriage return). The following is the first device stanza in the example file:

```
stacker_8mm  DEVICE="thor:/dev/rmt/4b"   DPOOL="8mm"
CLIENT="ProClient" DEVTYPE="STACKER" DEVCAP="10"
```

logical device name must be the first entry, and should describe the device. All subsequent entries associated with the logical device are in the form KEYWORD="argument", with the argument contained in quotation marks. The keywords can occur in any order after the logical device name.

DEVICE="*hostname*:/dev/*???*" is the hostname of the device and the UNIX tape device name.

DPOOL="*mediatype*" is the type of media used on this device. We suggest values such as 8mm, dlt, or 3590.

CLIENT="*ProClient*" entry must appear and set to ProClient.

DEVTYPE="*devicetype*" This entry should say SIMPLE or STACKER. If you intend for ProMAX to use robotic devices, you should use the device management capability. You will need to contact ProMAX Support for further information.

Recommended Entries In device.tts

CONFIG="FMNT" If a valid tape is found in the drive, FMNT causes the job to begin using that tape without issuing a mount message. FMNT is can be set for stacker devices and used for simple devices. Be aware that if a tape is not available in the device with FMNT set, it can take several minutes before a mount message appears.

Numerous additional entries are allowed for CONFIG and are described in the ***ProClient Command Arguments*** document.

Note: If you converted your device.txt to device.tts, you may see an entry LMNT. LMNT causes a tape mount message to be issued after all specified input tapes have been read. This prevents the job from terminating even though all processing is complete. This means the job retains an active position in job queues until you cancel the tape mount message. We recommend that you specify the correct number of input volumes in the job menu, and do not set LMNT.

RWAIT="nn" This sets the time in seconds to wait before attempting to access a device after an automated tape change, such as on a stacker or robot. Set the time large enough to allow the next tape to get to *load point*.

MAXBLK="nnnn" This sets the maximum block size to write or read on a device. We recommend that you run the utility **TapeDiag** to find the maximum supported block size for each device. On Linux, find maximum block size using "dmesg" command and use that number for MAXBLK. Data transfer rates are faster for larger block sizes.

TMHOST="*terminal:0.0*" This is the location to display mount messages issued for this device. If this is not supplied, mount message are sent to the location of the User Interface where the job was submitted.

Note: If using a remote tape drive, the default mount message only works if you specify the DISPLAY environmental variable to **hostname:0.0**. Mount messages are sent to the location where the job was submitted, if this variable is not set.

REMHOME="/a/b/c" This is only needed for devices on remote systems, that is a different system than the one where the ProMAX job is executing. Refer to the next section if this applies to your installation.

Please read ***ProClient Command Arguments*** document for additional options for controlling behavior of your tape devices.

Note: It is acceptable and often very useful to have several logical device names for the same physical device. The device descriptions below indicate the same physical device, but the second one has a CONFIG entry that indicates that the drive should not rewind the tape.

```
dlt_normal DEVICE="thor:/dev/rmt/4b" DPOOL="dlt" CLIENT="ProClient"
DEVTYPE="SIMPLE"
```

```
dlt_no_rewind DEVICE="thor:/dev/rmt/4b" DPOOL="dlt" CLIENT="ProClient"
DEVTYPE="SIMPLE" CONFIG="NREW"
```

Setting up the Remote Device Access

This section applies to installations requiring access to tape devices across a network. Skip this section if all of your tape devices are local to the ProMAX server.

Overview of requirements

- We provide an executable script to put required files on each remote tape host that does not have a complete ProMAX installation accessable.
- Ensure device.tts includes appropriate entries for all desired devices.
- All users needing access to devices on remote systems should have **rsh** access to those remote systems. If **rsh** is unacceptable due to security issues at your site, run $PROMAX_HOME/sys/bin/aar_promax on each remote tape host.

- ProMAX data primary storage must be accessible via identical paths from all hosts serving tape drives.

Installing Required Files

The remote tape host must have access to a specific directory of files. This directory is identified in the device.tts file by the keyword REMHOME. The directory indicated by REMHOME must be available by one of the following three circumstances:

- If the remote tape host has its own local ProMAX installation, set REMHOME to the location of that ProMAX installation directory.
- If the remote tape host is the same brand as the ProMAX host machine, that is both Sun or both IBM or both SGI machines, and the PROMAX_HOME directory is mounted such that it can be accessed from the remote tape host, set REMHOME to that directory.
- If the remote tape host is a different make than the ProMAX host machine, and does not have ProMAX installed, follow these steps:

1. Create a directory accessable to the remote tape host for installing the remote tape directory. Set the environment variable REMOTE_TAPE_HOME to this directory name.
2. Copy or ftp the file ***$PROMAX_HOME/port/misc/remote_tape/RemoteTapeInstall*** from the ProMAX host machine to REMOTE_TAPE_HOME.
3. From the directory $PROMAX_HOME/port/misc/remote_tape, copy or ftp the appropriate file for the operating system of the remote tape host to that machine:

   rs6000_remote_tar.Z

   sgimips4_remote_tar.Z

   solaris_remote_tar.Z

   linux_remote_tar.Z

   Note: Do not uncompress the file.
4. Execute the script **RemoteTapeInstall** copied in step 1. You may have to change permissions on the file to allow

execution. Nearly all installations should select the first option when prompted.

5. Edit the ***device.tts*** file to ensure the value for REMHOME is correct for each remote tape device.

If rsh privileges are unacceptable

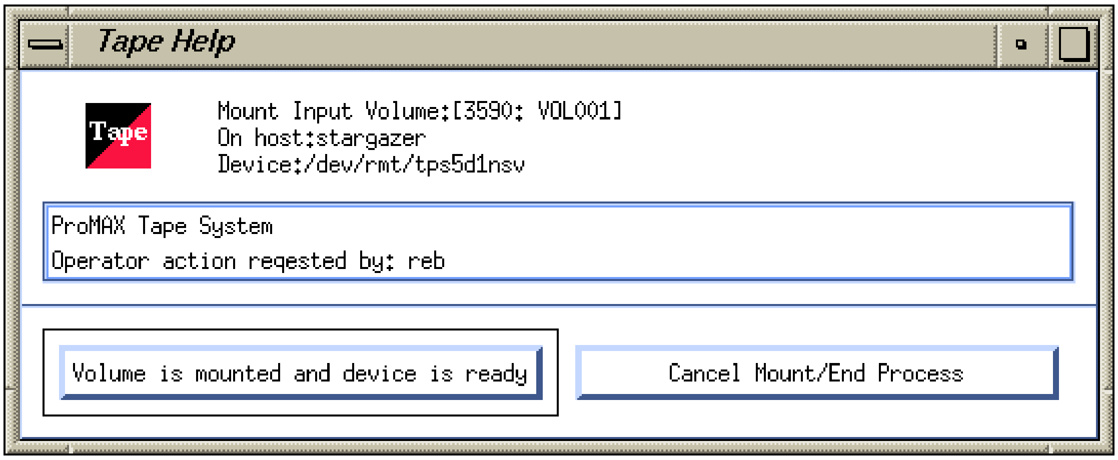
If **rsh** privileges pose an unacceptable security issue, ProMAX can use **aar_promax** for remote tape device access. In this case, all remote execution done via ProMAX will use aar_promax. The fundamental requirements for this are:

- aar_promax must be running on all machines involved in ProMAX, including each remote tape host.
- for remote, partial installs, using aar_promax, edit the $REMOTE_TAPE_HOME/etc/config_file and change the path to the log file from /release/1998.6/ProMAX to match $REMOTE_TAPE_HOME.
- in $PROMAX_HOME/etc/config_file, you must activate the following stanza: **remote_exec_type: aar**.
- start aar_promax, found in REMHOME/sys/bin by the UNIX super-daemon called **inetd**. Refer to ***Enabling the Account/Remote Execution Server*** in the *ProMAX System Administration* guide for information on starting aar_promax automatically during system start-up.

Alternatively, ProMAX can be configured to use Secure Shell (**ssh**) for remote execution. See ***Setting Up Parallel Processing*** in the *ProMAX System Administration* guide. **aar_promax** may still be used for ProMAX accounting.

Tape Mount Messages

When a tape needs to be mounted on a device, the following message window appears:


Mount Input Volume:[3590: VOL001]
On host:stargazer
Device:/dev/rmt/tps5d1nsv

ProMAX Tape System
Operator action reqested by: reb

This message appears on the terminal specified by the keyword **TMHOST** in the ***device.tts*** file. If this keyword is not present, the mount message appears on the terminal from which the tape job was submitted. You can set TMHOST to any terminal. Common choices would be the terminal nearest the device, the terminal where the user submitted the job (this is the default), or the terminal of your designated tape operator.

When a mount message appears, allow the tape to reach *load point*, then click **Volume is mounted and device is ready**. To cancel the mount, click **Cancel Mount/End Process**. Cancelling the mount cancels the job and all its child processes.

Configuration with a Tape Catalog

This configuration is for clients with local and remote drives with tape catalog, but no device management.

Configuration requirements

- Complete Scenario 1 configuration
- TopCat catalog directory and files visible to all ProMAX compute servers
- Environment variable PROMAX_TOPCAT_HOME pointing to TopCat directory (if not $PROMAX_HOME/TopCat)

Configuring the TopCat Tape Catalog

Refer to Configure the ***Using the TopCat Tape Catalog*** for information on defining a tape catalog. Refer to the ***Using ShowCat*** for usage of the ShowCat java catalog interface.

Usage

This chapter gives details for using:

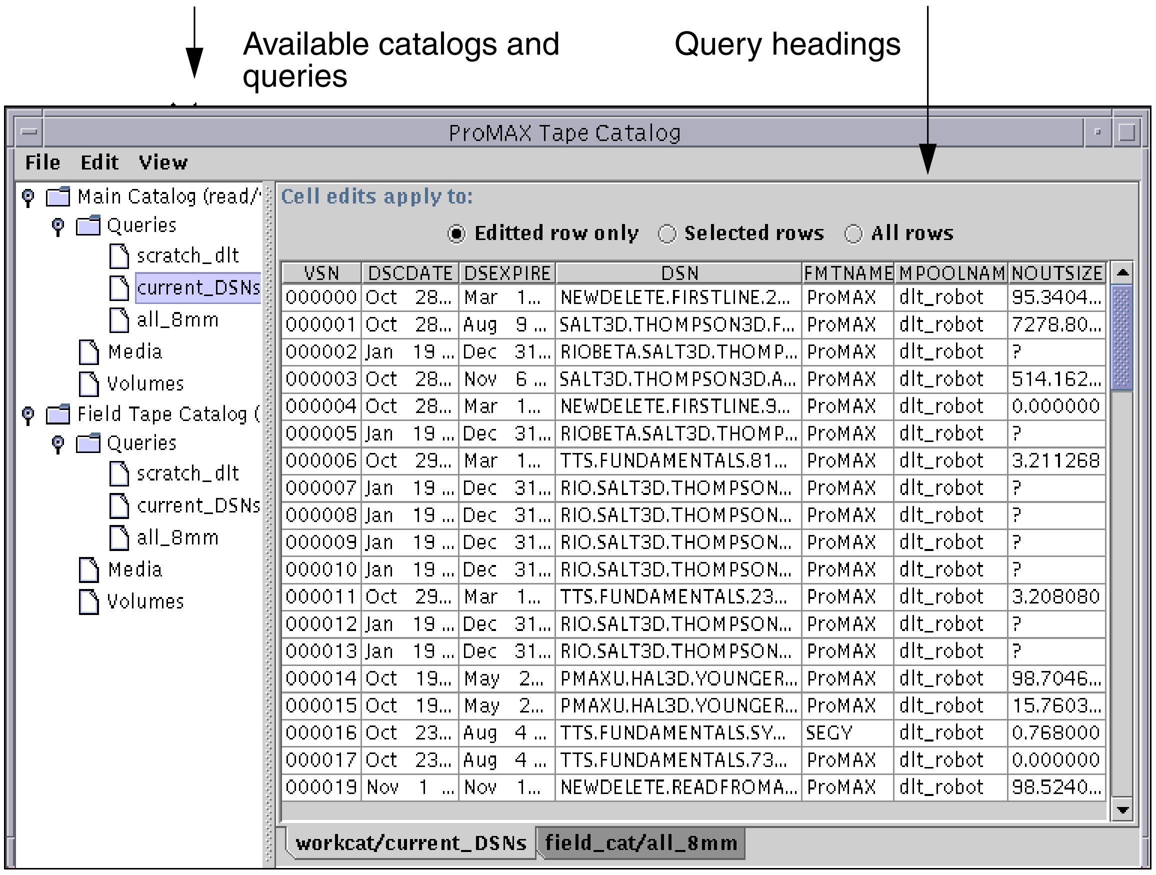
Using the ShowCat Catalog

The ShowCat Tape Catalog is the main interface for querying and viewing TopCat Tape Catalog information. It is used to directly edit Catalog fields. With this tool, you can make corrections or insert additional information, such as the tape catalog's COMMENT field.

Prerequisites to running the ShowCat script $PROMAX_HOME/port/bin/showcat are:

- environment variables DISPLAY, PROMAX_HOME are correctly set in your environment.
- if the TopCat Catalog does not reside under the ProMAX installation at $PROMAX_HOME/TopCat, set PROMAX_TOPCAT_HOME to point to its correct location.

ShowCat User Interface

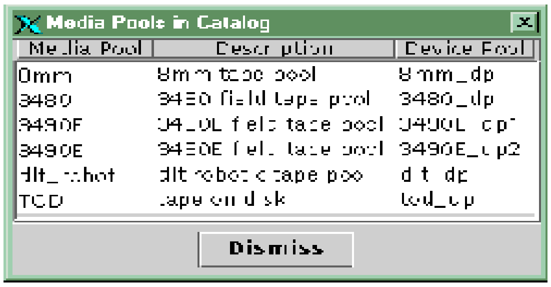


Cell edits apply to:

(● Editted row only) ○ Selected rows ○ All rows

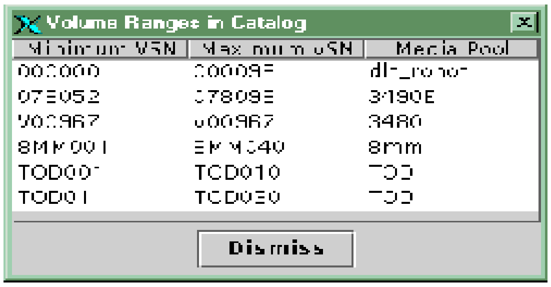
VSN	DSCDATE	DSEXPIRE	DSN	FMTNAME	MPOOLNAM	NOUTSIZE
000000	Oct 28...	Mar 1...	NEWDELETE.FIRSTLINE.2...	ProMAX	dlt_robot	95.3404...
000001	Oct 28...	Aug 9 ...	SALT3D.THOMPSON3D.F...	ProMAX	dlt_robot	7278.80...
000002	Jan 19 ...	Dec 31...	RIOBETA.SALT3D.THOMP...	ProMAX	dlt_robot	?
000003	Oct 28...	Nov 6 ...	SALT3D.THOMPSON3D.A...	ProMAX	dlt_robot	514.162...
000004	Oct 28...	Mar 1...	NEWDELETE.FIRSTLINE.9...	ProMAX	dlt_robot	0.000000
000005	Jan 19 ...	Dec 31...	RIOBETA.SALT3D.THOMP...	ProMAX	dlt_robot	?
000006	Oct 29...	Mar 1...	TTS.FUNDAMENTALS.81...	ProMAX	dlt_robot	3.211268
000007	Jan 19 ...	Dec 31...	RIO.SALT3D.THOMPSON...	ProMAX	dlt_robot	?
000008	Jan 19 ...	Dec 31...	RIO.SALT3D.THOMPSON...	ProMAX	dlt_robot	?
000009	Jan 19 ...	Dec 31...	RIO.SALT3D.THOMPSON...	ProMAX	dlt_robot	?
000010	Jan 19 ...	Dec 31...	RIO.SALT3D.THOMPSON...	ProMAX	dlt_robot	?
000011	Oct 29...	Mar 1...	TTS.FUNDAMENTALS.23...	ProMAX	dlt_robot	3.208080
000012	Jan 19 ...	Dec 31...	RIO.SALT3D.THOMPSON...	ProMAX	dlt_robot	?
000013	Jan 19 ...	Dec 31...	RIO.SALT3D.THOMPSON...	ProMAX	dlt_robot	?
000014	Oct 19...	May 2...	PMAXU.HAL3D.YOUNGER...	ProMAX	dlt_robot	98.7046...
000015	Oct 19...	May 2...	PMAXU.HAL3D.YOUNGER...	ProMAX	dlt_robot	15.7603...
000016	Oct 23...	Aug 4 ...	TTS.FUNDAMENTALS.SY...	SEGY	dlt_robot	0.768000
000017	Oct 23...	Aug 4 ...	TTS.FUNDAMENTALS.73...	ProMAX	dlt_robot	0.000000
000019	Nov 1 ...	Nov 1...	NEWDELETE.READFROMA...	ProMAX	dlt_robot	98.5240...

Lists appear in compressed format when the UI is first opened. Pointer-sensitive bubble help appears to help navigation through the options. Click the toggle button next to a catalog name to expand its information list and then next to the Query heading to view the list of saved queries. To view summary information about media types and volume ranges for a particular catalog, click **MB3** on **Media** and then **Volumes**. The follwoing results of these information queries appear as separate windows and are equivalent to executing topcat CATINFO[INFMPOOL] or topcat CATINFO[INFSCAN] from a UNIX shell:


Media Pools in Catalog

Media Pool	Description	Device Pool
8mm	8mm tape pool	8mm_dp
3480	3480 field tape pool	3480_dp
3490F	3490F field tape pool	3490F_dp
3490E	3490E field tape pool	3490E_dp2
dlt_robot	dlt robotic tape pool	dlt_dp
TCD	tape on disk	tcd_dp

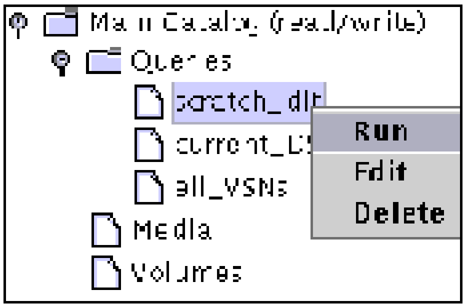
Dismiss


Volume Ranges in Catalog

Minimum VSN	Maximum VSN	Media Pool
000000	00009F	dlt_robot
073052	078093	3490E
V00987	V00987	3480
8MM001	8MM040	8mm
TCD001	TCD010	TCD
TCD011	TCD030	TCD

Dismiss

Activating a Query



To activate a query, click **MB3** on the query and select to **Run**, **Edit**, or **Delete** that query. The results of running a query appear as a tab in the right pane. The tab title bar shows the catalog name and that of the query run against it. You can redefine the relative size of left and right panes by dragging the dotted vertical bar. Resize columns in the right pane by placing your cursor over column boundaries in the header. Reorder columns by dragging headers. Sort on a column in ascending order by clicking on its column heading. Click again to sort in descending order. You can also select a cell and choose **View** ➛ **Sort by selected column** from the menu bar. Sorts and column redefinitions are used only for the current session.

Menu Bar

FIle

- **Exit** from the ShowCat catalog.

Edit

- **Undo** the previous edit. In order to be able to directly edit Catalog fields from the ShowCat UI, your user ID must be included in the "CATADMIN:" list of the catalog's catadmin file. If you keep clicking **Undo**, ShowCat recursively undoes each of the edits you have made.
- **Redo** the last undo.

View

- **Reread Catalog** re-executes the catalog query. Column sizing, order and sorting are retained. A query is also re-executed if its tab is selected after being hidden behind another query tab.
- **Sort by Selected Column** when a cell is highlighted. This can also be done by clicking on the column header.

Above the Query headings are the following options:

Cell edits apply to:

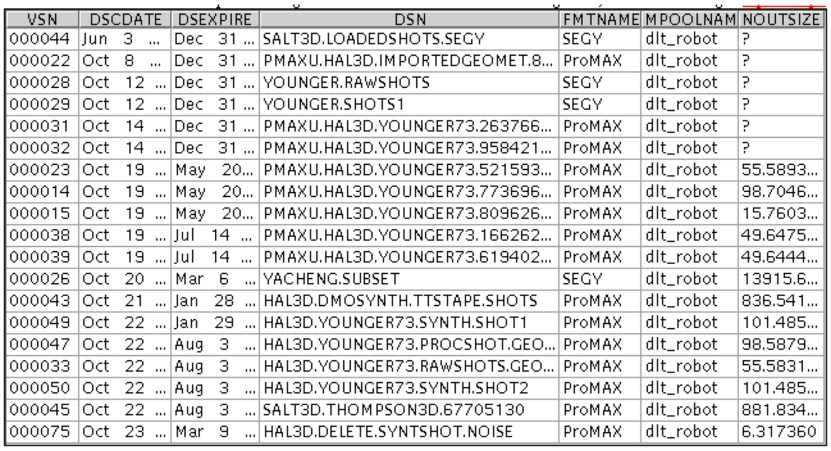
- **Editted row only** Any changes made to a TopCat field only applies to that VSN.
- **Selected rows** Any changes made to a TopCat field only applies to the highlighted range of VSNs.
- **All rows** Any changes made to a TopCat field applies to all VSNs. If you use this option to apply an edit to all VSNs, set it back to "Editted row only" just in case in a later edit accidentally gets applied to all VSNs unintentionally.

In this same area, you can click **MB3** to select from:

- **Refresh** is equivalent to View ➛ Reread catalog.
- **Delete** removes active query results pane.

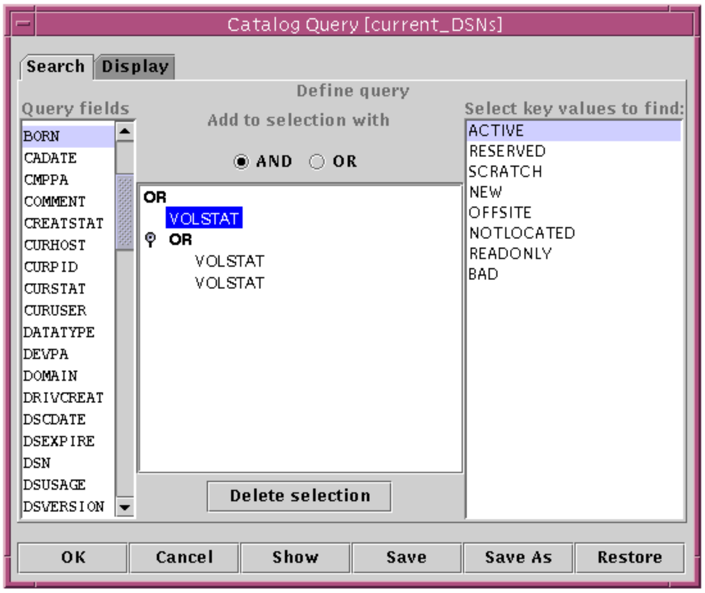
Query Definition

The query results shown below include 7 of the 85 TopCat catalog fields. For information on all the catalog fields, refer to the listing of ***TopCat Tape Catalog Attributes***.



VSN	DSCDATE	DSEXPIRE	DSN	FMTNAME	MPOOLNAM	NOUTSIZE
000044	Jun 3 ...	Dec 31 ...	SALT3D.LOADEDSHOTS.SEGY	SEGY	dlt_robot	?
000022	Oct 8 ...	Dec 31 ...	PMAXU.HAL3D.IMPORTEDGEOMET.8...	ProMAX	dlt_robot	?
000028	Oct 12 ...	Dec 31 ...	YOUNGER.RAWSHOTS	SEGY	dlt_robot	?
000029	Oct 12 ...	Dec 31 ...	YOUNGER.SHOTS1	SEGY	dlt_robot	?
000031	Oct 14 ...	Dec 31 ...	PMAXU.HAL3D.YOUNGER73.263766...	ProMAX	dlt_robot	?
000032	Oct 14 ...	Dec 31 ...	PMAXU.HAL3D.YOUNGER73.958421...	ProMAX	dlt_robot	?
000023	Oct 19 ...	May 20...	PMAXU.HAL3D.YOUNGER73.521593...	ProMAX	dlt_robot	55.5893...
000014	Oct 19 ...	May 20...	PMAXU.HAL3D.YOUNGER73.773696...	ProMAX	dlt_robot	98.7046...
000015	Oct 19 ...	May 20...	PMAXU.HAL3D.YOUNGER73.809626...	ProMAX	dlt_robot	15.7603...
000038	Oct 19 ...	Jul 14 ...	PMAXU.HAL3D.YOUNGER73.166262...	ProMAX	dlt_robot	49.6475...
000039	Oct 19 ...	Jul 14 ...	PMAXU.HAL3D.YOUNGER73.619402...	ProMAX	dlt_robot	49.6444...
000026	Oct 20 ...	Mar 6 ...	YACHENG.SUBSET	SEGY	dlt_robot	13915.6...
000043	Oct 21 ...	Jan 28 ...	HAL3D.DMOSYNTH.TTSTAPE.SHOTS	ProMAX	dlt_robot	836.541...
000049	Oct 22 ...	Jan 29 ...	HAL3D.YOUNGER73.SYNTH.SHOT1	ProMAX	dlt_robot	101.485...
000047	Oct 22 ...	Aug 3 ...	HAL3D.YOUNGER73.PROCSHOT.GEO...	ProMAX	dlt_robot	98.5879...
000033	Oct 22 ...	Aug 3 ...	HAL3D.YOUNGER73.RAWSHOTS.GEO...	ProMAX	dlt_robot	55.5831...
000050	Oct 22 ...	Aug 3 ...	HAL3D.YOUNGER73.SYNTH.SHOT2	ProMAX	dlt_robot	101.485...
000045	Oct 22 ...	Aug 3 ...	SALT3D.THOMPSON3D.67705130	ProMAX	dlt_robot	881.834...
000075	Oct 23 ...	Mar 9 ...	HAL3D.DELETE.SYNTSHOT.NOISE	ProMAX	dlt_robot	6.317360

Query selection criteria are defined using the **Query Edit** window:



Add to selection with: [✓ AND] OR

Query fields: BORN, CADATE, CMPPA, COMMENT, CREATSTAT, CURHOST, CURPID, CURSTAT, CURUSER, DATATYPE, DEVPA, DOMAIN, DRIVCREAT, DSCDATE, DSEXPIRE, DSN, DSUSAGE, DSVERSION

```
OR
  VOLSTAT
  OR
    VOLSTAT
    VOLSTAT
```

Select key values to find: ACTIVE, RESERVED, SCRATCH, NEW, OFFSITE, NOTLOCATED, READONLY, BAD

To make a selection:

1. Double click to select a Catalog field from the left column
2. Choose a value for that field from the right column.

Depending on the field selected, the right pane can contain a selection list or blanks for specifying a numeric range, string or wildcard match.

Add additional criteria by:

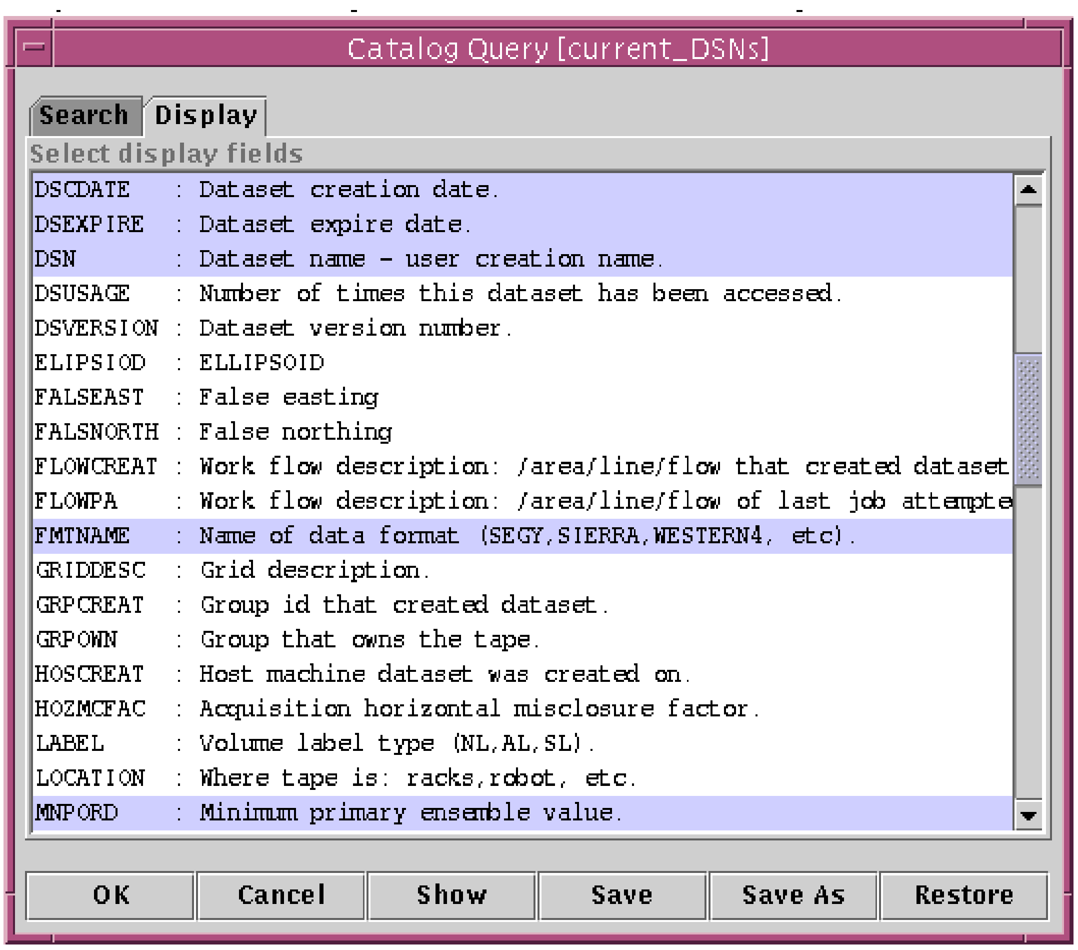
3. Selecting AND or OR logic at the top of the center pane.
4. Double click the next search field.

This query selects all VSNs for which VOLSTAT is ACTIVE, RESERVED or READONLY.

The VOLSTAT volume status field is one of the most useful TopCat fields, with the following value definitions:

ACTIVE	Volume is in use and contains non-expired data.
RESERVED	Volume is reserved, usually by a running job.
SCRATCH	Volume is scratch and available for output.
NEW	Volume is newly initialized (via topcat POPULATE) and available for output.
OFFSITE	Volume is offsite. This value may be manually set by the administrator
NOTLOCATED	Volume is not found. With robotic tape libraries, TopCat will set this if a volume is not located by the robot
READONLY	Volume is read-only, and cannot be written to. TopCat will also set this if it encounters a write-protected volume
BAD	Volume is bad or unuseable. TopCat may set this if an I/O error is encountered

Fields to display for the search results are selected from the **Query Display** tab. All fields available for display are listed alphabetically. Select blocks of fields using **MB1** ➛**Shift MB1** and discrete entries using **Ctrl MB1**.



```
DSCDATE   : Dataset creation date.
DSEXPIRE  : Dataset expire date.
DSN       : Dataset name - user creation name.
DSUSAGE   : Number of times this dataset has been accessed.
DSVERSION : Dataset version number.
ELIPSIOD  : ELLIPSOID
FALSEAST  : False easting
FALSNORTH : False northing
FLOWCREAT : Work flow description: /area/line/flow that created dataset
FLOWPA    : Work flow description: /area/line/flow of last job attempte
FMTNAME   : Name of data format (SEGY,SIERRA,WESTERN4, etc).
GRIDDESC  : Grid description.
GRPCREAT  : Group id that created dataset.
GRPOWN    : Group that owns the tape.
HOSCREAT  : Host machine dataset was created on.
HOZMCFAC  : Acquisition horizontal misclosure factor.
LABEL     : Volume label type (NL,AL,SL).
LOCATION  : Where tape is: racks,robot, etc.
MNPORD    : Minimum primary ensemble value.
```

Menu Bar

OK

Execute the query and exit the **Query Editor** without saving the query. A new, unsaved query appears as **/unknown** on the results tab. If you edit an existing query without saving it, **OK** dismisses the Query Edit window without re-executing the query. Save your edits to re-execute.

Cancel

Exit the Query Editor without saving or executing the query.

Show

Display the query as it is executed by topcat. This information is mainly useful if you want to put together shell scripts executing topcat from the command line. You can use your mouse to copy-and-paste the string from the **Query Command** window into a file. See TopCat User Commands: ***QRY***.

Save

Save the query under its current name. This dismises the **Query Edit** window. To execute the updated query, select it from the left pane list of queries and choose **Run** from **MB3** pop-up. If a results view from this query already exists in the right pane, use the **MB3**➛Delete option to remove the old version from memory and then **Run** from the query list.

Save As

Save the query to a name specified in the edit window. Each saved query is a file in your $HOME/.savedQueries/ directory.

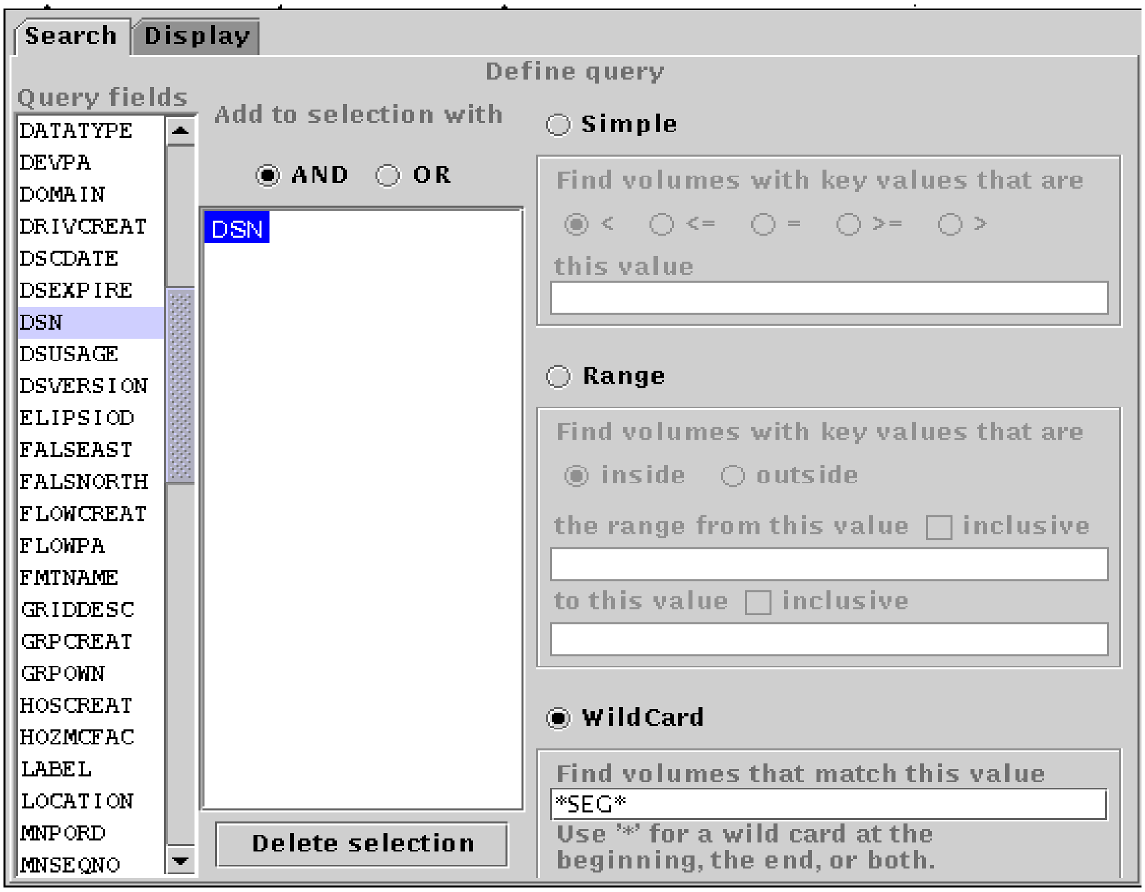
The name you enter into the **Type name for saved query** prompt (including blanks and punctuation marks) appears under Query drop list in the ShowCat UI.

Restore

Load a query, selected from the window, into the editor.

Examples

The following query performs a search on any DSN which has the pattern **SEG** in the DSN dataset name; the ’ *’ asterisk is a wildcard character.



Add to selection with: [✓ AND] OR

DSN

Simple: [✓ <] <= = >= >

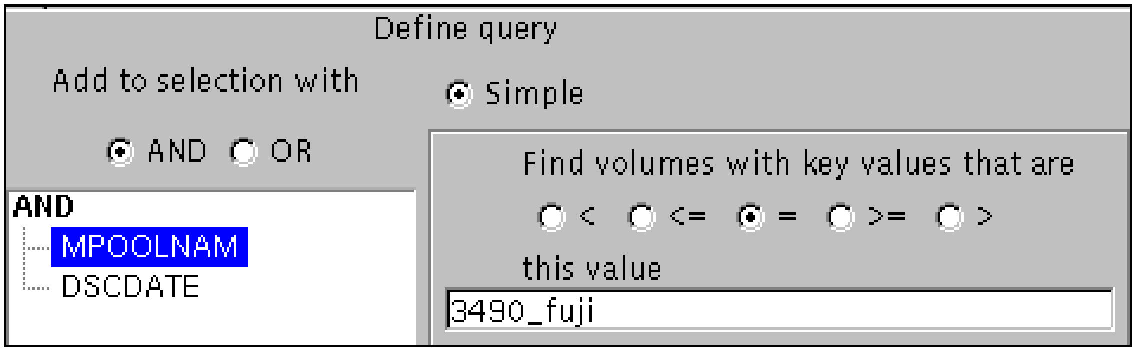
Range: [✓ inside] outside; inclusive ☐; inclusive ☐

[✓ WildCard]

Find volumes that match this value: *SEG*

To list all VSNs, without any subsetting or querying, construct a **Simple** query to show all VSNs **>=** to **0** or use a WildCard match on "*".

The following example is one of a compound query based on two attributes. This query says, "Show me all VSNs whose media pool (MPOOLNAM) is **3490_fuji** *AND* whose dataset's DSCDATE (dataset creation date) were created later than **May 12, 1998**."

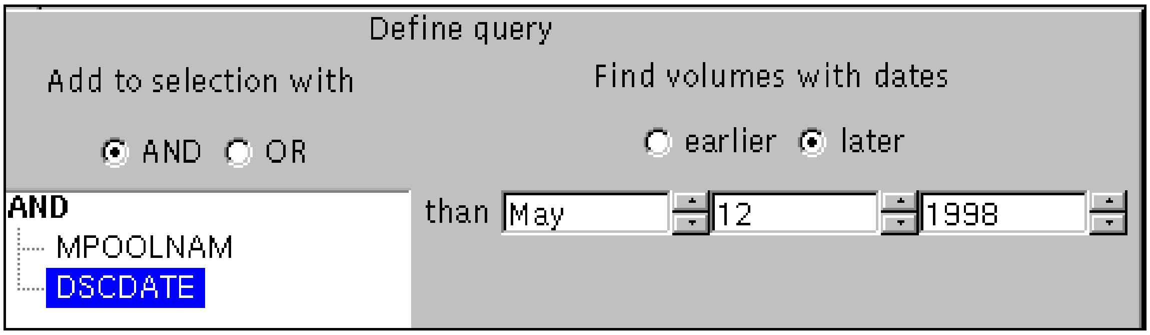


[✓ AND] OR

AND
- MPOOLNAM
- DSCDATE

Find volumes with key values that are < <= [✓ =] >= >

this value: 3490_fuji



[✓ AND] OR

AND
- MPOOLNAM
- DSCDATE

Find volumes with dates earlier [✓ later]

than May 12 1998

Some of the most useful TopCat fields to display are:

VSN	BORN	CREATSTAT	CURSTAT	DATATYPE
DSCDATE	DSN	FLOWPA	FMTNAME	HOSCREAT
FLOWCREAT	LABEL	MPOOLNAM	NXTVOL	PRSTAT
UNAMCREAT	PRVVOL	USERPA	VOLSEQ	VOLSTAT

Using the OpCon Operator Console

OpCon Operator Console provides a single interface for the dispatching and monitoring of tape mount requests and monitoring of device activity. It also monitors device allocation and lets the operator disable/enable devices and device pools.

Note: You must contact ProMAX Support in Denver for information required to enable the use of PMX device management.

Cancelling Jobs Submitted from Batch Queues

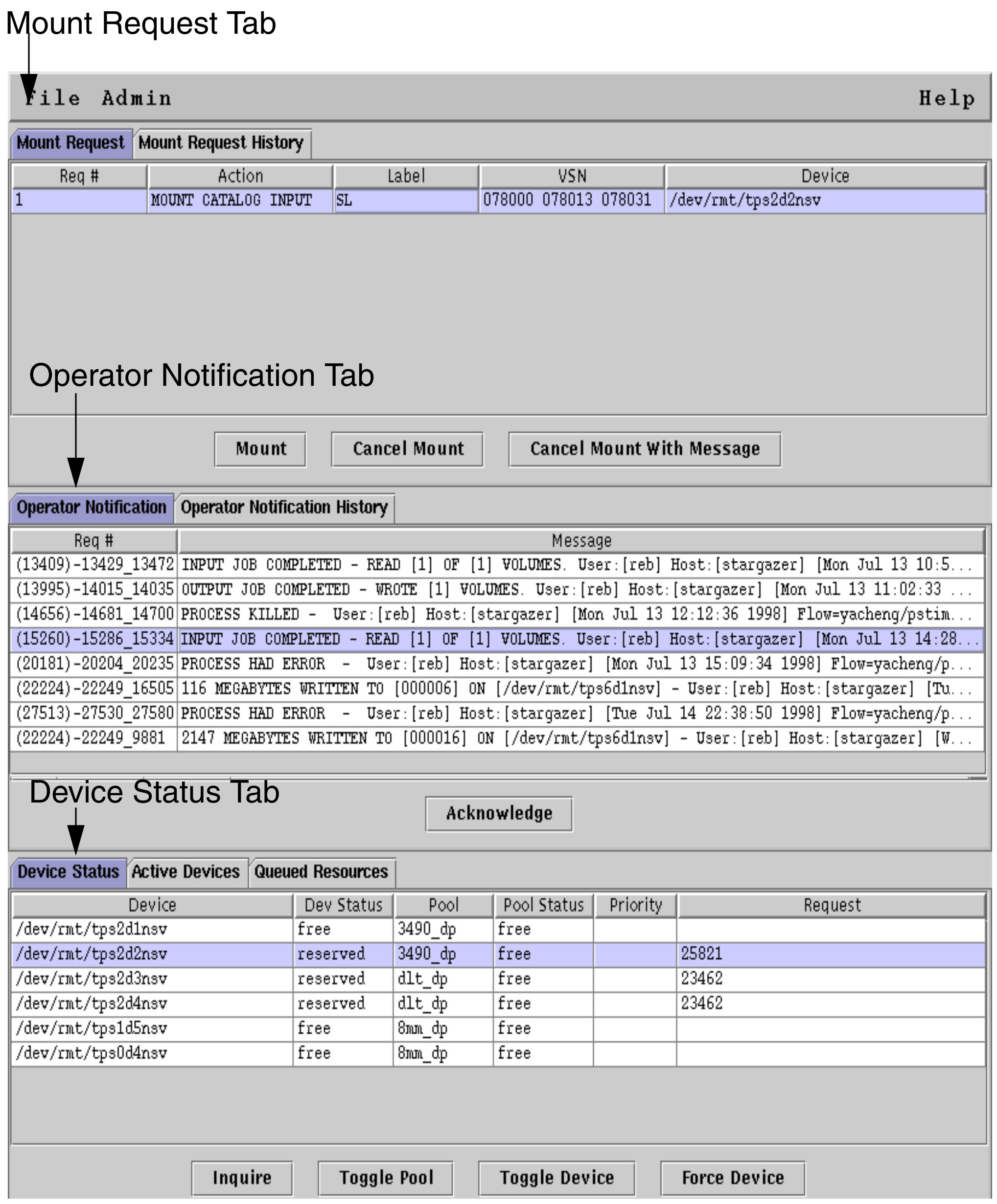
If you are using batch job queues (lp or NQE), the following steps allow you to cancel jobs from the operator console:

1. Edit file $PROMAX_HOME/port/bin/PromaxOpConFind.
2. Locate the line export QUETYPE="NONE".
3. Change "NONE" to "LPD" or "NQS".
4. Save the file.

OpCon Panes

This section discusses the following OpCon Panes: Mount Request, Operator Notification, Device Status, Mount Request History, Operator Notification History, Active Devices, and Queued Resources.

The following graphic is the main OpCon user interface. You can resize the width of all columns in the display by dragging the edges of the header labels.



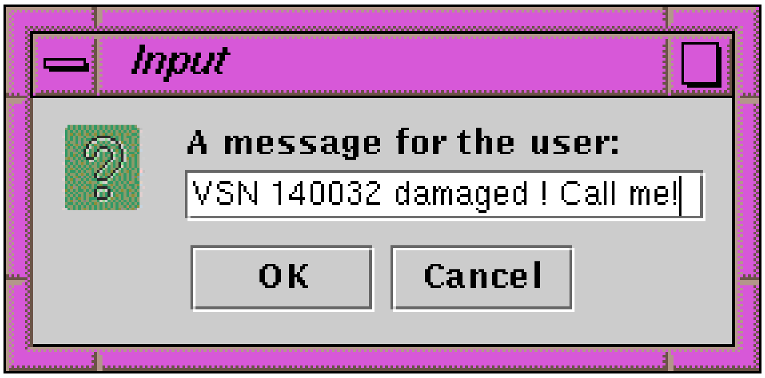
Req #	Action	Label	VSN	Device
1	MOUNT CATALOG INPUT	SL	078000 078013 078031	/dev/rmt/tps2d2nsv

Req #	Message
(13409)-13429_13472	INPUT JOB COMPLETED - READ [1] OF [1] VOLUMES. User:[reb] Host:[stargazer] [Mon Jul 13 10:5...
(13995)-14015_14035	OUTPUT JOB COMPLETED - WROTE [1] VOLUMES. User:[reb] Host:[stargazer] [Mon Jul 13 11:02:33 ...
(14656)-14681_14700	PROCESS KILLED - User:[reb] Host:[stargazer] [Mon Jul 13 12:12:36 1998] Flow=yacheng/pstim...
(15260)-15286_15334	INPUT JOB COMPLETED - READ [1] OF [1] VOLUMES. User:[reb] Host:[stargazer] [Mon Jul 13 14:28...
(20181)-20204_20235	PROCESS HAD ERROR - User:[reb] Host:[stargazer] [Mon Jul 13 15:09:34 1998] Flow=yacheng/p...
(22224)-22249_16505	116 MEGABYTES WRITTEN TO [000006] ON [/dev/rmt/tps6d1nsv] - User:[reb] Host:[stargazer] [Tu...
(27513)-27530_27580	PROCESS HAD ERROR - User:[reb] Host:[stargazer] [Tue Jul 14 22:38:50 1998] Flow=yacheng/p...
(22224)-22249_9881	2147 MEGABYTES WRITTEN TO [000016] ON [/dev/rmt/tps6d1nsv] - User:[reb] Host:[stargazer] [W...

Device	Dev Status	Pool	Pool Status	Priority	Request
/dev/rmt/tps2d1nsv	free	3490_dp	free		
/dev/rmt/tps2d2nsv	reserved	3490_dp	free		25821
/dev/rmt/tps2d3nsv	reserved	dlt_dp	free		23462
/dev/rmt/tps2d4nsv	reserved	dlt_dp	free		23462
/dev/rmt/tps1d5nsv	free	8mm_dp	free		
/dev/rmt/tps0d4nsv	free	8mm_dp	free		

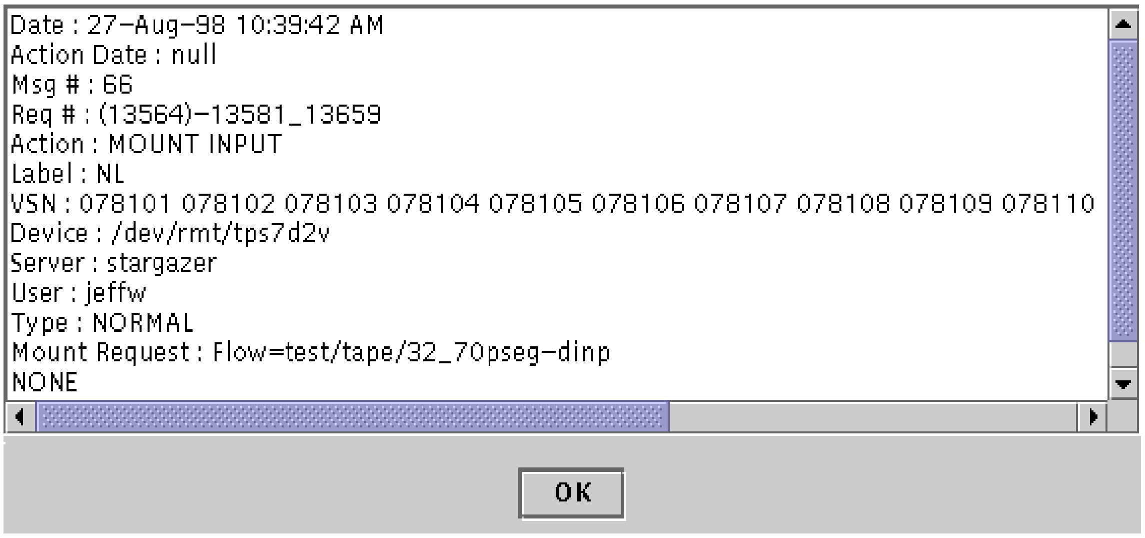
Mount Request

Tape mount requests appear in the top pane. When a mount request appears, mount the identified tape in the indicated device, then click on the mount request line to include. It becomes highlighted. Click **Mount** to begin accessing the tape, or click **Cancel Mount** to abort the job. All answered mount requests are sent to the **Mount Request History** pane.



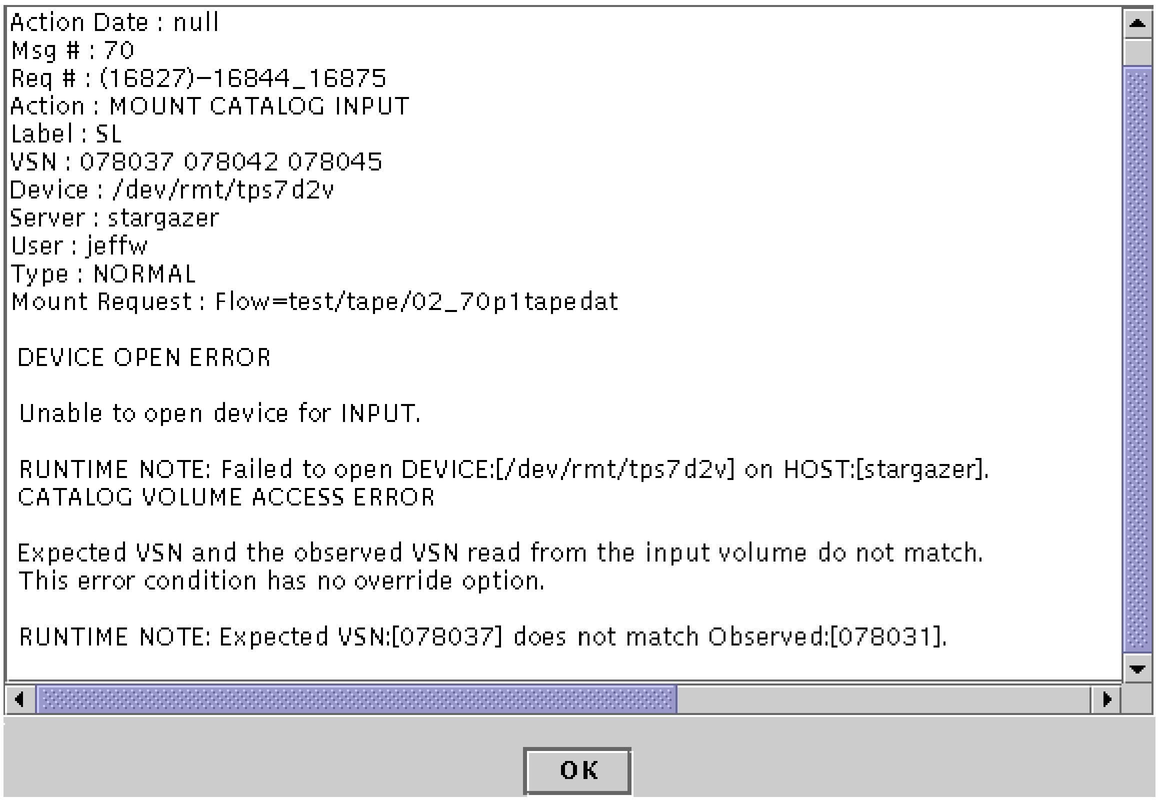
Click **Cancel Mount With Message** to display the window shown to the left. From this text window, an operator can communicate to the user the reason the mount request was cancelled. Click **OK** to print this message is printed to your job.output. Otherwise, click **Cancel** to cancel the mount. This is the same as clicking **Cancel Mount**, but no message is sent.

Click **MB2** on the mount request message to display a window similar to the one below. Most of the information here is already showing on the **Mount Request** pane, but this window contains extra information such as the Server and User of the request.



```
Date : 27-Aug-98 10:39:42 AM
Action Date : null
Msg # : 66
Req # : (13564)-13581_13659
Action : MOUNT INPUT
Label : NL
VSN : 078101 078102 078103 078104 078105 078106 078107 078108 078109 078110
Device : /dev/rmt/tps7d2v
Server : stargazer
User : jeffw
Type : NORMAL
Mount Request : Flow=test/tape/32_70pseg-dinp
NONE
```

Depending on how the tape system was configured, some mount errors can cause the job to return with another mount message. In these cases, more diagnostic information can often be found by clicking **MB2** on the mount request to get an explanation of the problem. For example:



```
Action Date : null
Msg # : 70
Req # : (16827)-16844_16875
Action : MOUNT CATALOG INPUT
Label : SL
VSN : 078037 078042 078045
Device : /dev/rmt/tps7d2v
Server : stargazer
User : jeffw
Type : NORMAL
Mount Request : Flow=test/tape/02_70p1tapedat

 DEVICE OPEN ERROR

 Unable to open device for INPUT.

 RUNTIME NOTE: Failed to open DEVICE:[/dev/rmt/tps7d2v] on HOST:[stargazer].
 CATALOG VOLUME ACCESS ERROR

 Expected VSN and the observed VSN read from the input volume do not match.
 This error condition has no override option.

 RUNTIME NOTE: Expected VSN:[078037] does not match Observed:[078031].
```

Operator Notification

This pane shows all notification messages sent by ProClient via the $PROMAX_HOME/port/bin/TAPE_NOTIFICATION Java client.

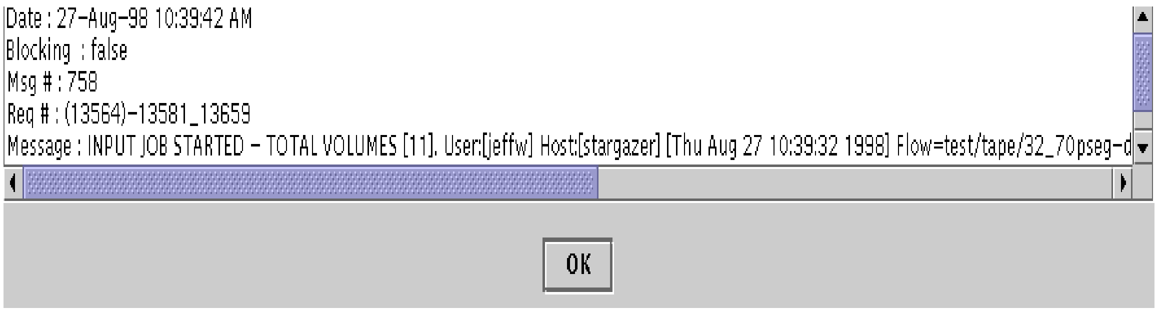
The **Req #** is a triplet of process IDs (PIDs) representing:

(u#####-file image of packet file)-(superexec)_(proclient)

There is no tracking of the ProMAX executive (exec.exe) because the exec has not yet started when the OpCon begins tracking the job.

Highlight a notification message and click **Acknowledge** to send that message to the **Operator Notification History** pane. To acknowledge a set of messages at the same time, select all of the messages using **Ctrl-MB1** and click **Acknowledge**. A continuous range of messages are acknowledged by using **Shift-MB1** or by dragging **Ctrl-MB1** over a range of messages, and then clicking **Acknowledge**.

Like the **Mount Request** pane, click **MB2** on each **Operator Notification** message to display a window containing extra information. For example:


```
Date : 27-Aug-98 10:39:42 AM
Blocking : false
Msg # : 758
Req # : (13564)-13581_13659
Message : INPUT JOB STARTED - TOTAL VOLUMES [11]. User:[jeffw] Host:[stargazer] [Thu Aug 27 10:39:32 1998] Flow=test/tape/32_70pseg-d
```

Device Status

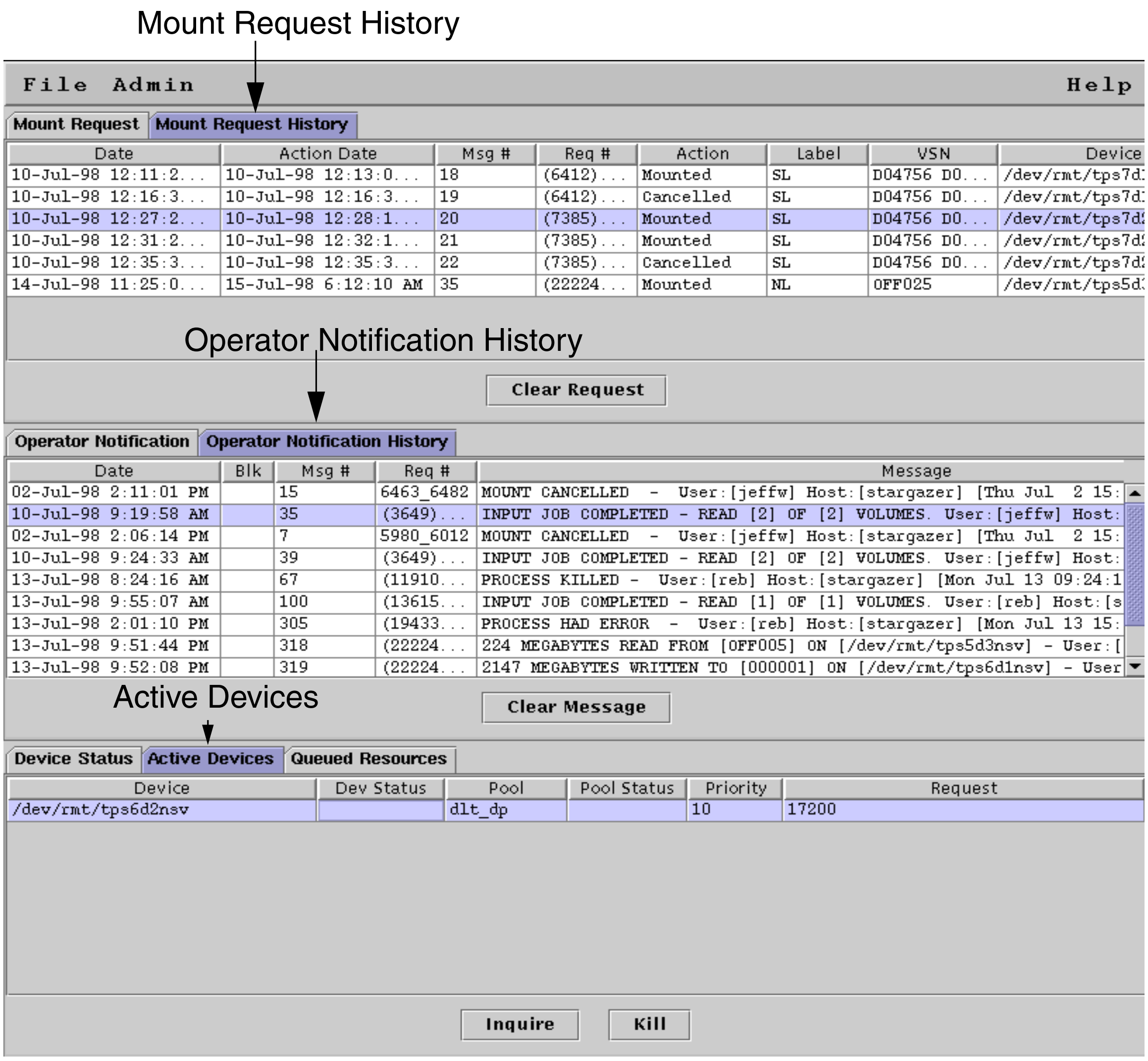
Click the **Device Manager** option of the **Admin** menu to display this pane. You can also enter **Ctrl-D**. Click the **Device Manager** option or enter **Ctrl-D** a second time to remove the **Device Manager** pane from the OpCon window. The number in the **Request** column is the PID of the executing "u#####" script that ProMAX stores in a runtime subdirectory within the flow directory during a job.

Every time you click **Inquire**, Opcon executes $PROMAX_HOME/sys/bin/pmxDev to update the status information of all devices and pools managed by the PMX Resource Manager routines. If you click either the **Active Devices** or **Queued Resources** tabs and then click the **Device Status** tab, the status information is updated.

If you highlight one of the devices and click **Toggle Pool**, Opcon executes $PROMAX_HOME/sys/bin/pmx. Toggle all devices belonging in the same device pool as the highlighted device between **free** (enabled) and **disabled**.

If you highlight a device and click **Toggle Device**, Opcon executes $PROMAX_HOME/sys/bin/pmx. You will be able to toggle that device between **free** (enabled) and **disabled**.

Click **Force Device** on a drive to execute $PROMAX_HOME/sys/bin/pmx. Set to forcefully free a drive.



Date	Action Date	Msg #	Req #	Action	Label	VSN	Device
10-Jul-98 12:11:2...	10-Jul-98 12:13:0...	18	(6412)...	Mounted	SL	D04756 D0...	/dev/rmt/tps7d
10-Jul-98 12:16:3...	10-Jul-98 12:16:3...	19	(6412)...	Cancelled	SL	D04756 D0...	/dev/rmt/tps7d
10-Jul-98 12:27:2...	10-Jul-98 12:28:1...	20	(7385)...	Mounted	SL	D04756 D0...	/dev/rmt/tps7d
10-Jul-98 12:31:2...	10-Jul-98 12:32:1...	21	(7385)...	Mounted	SL	D04756 D0...	/dev/rmt/tps7d
10-Jul-98 12:35:3...	10-Jul-98 12:35:3...	22	(7385)...	Cancelled	SL	D04756 D0...	/dev/rmt/tps7d
14-Jul-98 11:25:0...	15-Jul-98 6:12:10 AM	35	(22224...	Mounted	NL	0FF025	/dev/rmt/tps5d

Date	Blk	Msg #	Req #	Message
02-Jul-98 2:11:01 PM		15	6463_6482	MOUNT CANCELLED - User:[jeffw] Host:[stargazer] [Thu Jul 2 15:
10-Jul-98 9:19:58 AM		35	(3649)...	INPUT JOB COMPLETED - READ [2] OF [2] VOLUMES. User:[jeffw] Host:
02-Jul-98 2:06:14 PM		7	5980_6012	MOUNT CANCELLED - User:[jeffw] Host:[stargazer] [Thu Jul 2 15:
10-Jul-98 9:24:33 AM		39	(3649)...	INPUT JOB COMPLETED - READ [2] OF [2] VOLUMES. User:[jeffw] Host:
13-Jul-98 8:24:16 AM		67	(11910...	PROCESS KILLED - User:[reb] Host:[stargazer] [Mon Jul 13 09:24:1
13-Jul-98 9:55:07 AM		100	(13615...	INPUT JOB COMPLETED - READ [1] OF [1] VOLUMES. User:[reb] Host:[s
13-Jul-98 2:01:10 PM		305	(19433...	PROCESS HAD ERROR - User:[reb] Host:[stargazer] [Mon Jul 13 15:
13-Jul-98 9:51:44 PM		318	(22224...	224 MEGABYTES READ FROM [0FF005] ON [/dev/rmt/tps5d3nsv] - User:[
13-Jul-98 9:52:08 PM		319	(22224...	2147 MEGABYTES WRITTEN TO [000001] ON [/dev/rmt/tps6d1nsv] - User

Device	Dev Status	Pool	Pool Status	Priority	Request
/dev/rmt/tps6d2nsv		dlt_dp		10	17200

Mount Request History

This pane collects all mount requests answered from the **Mount Request** tabbed pane. Highlight these entries and click **Clear Request** to remove them from the listing.

Operator Notification History

This pane collects all notification messages acknowledged from the **Operator Notification** tabbed pane by clicking **Acknowledge**. Highlight a set of messages using **Ctrl-MB1**, or a range of messages using **Shift-MB1**, and click **Clear Message** to remove them from the listing.

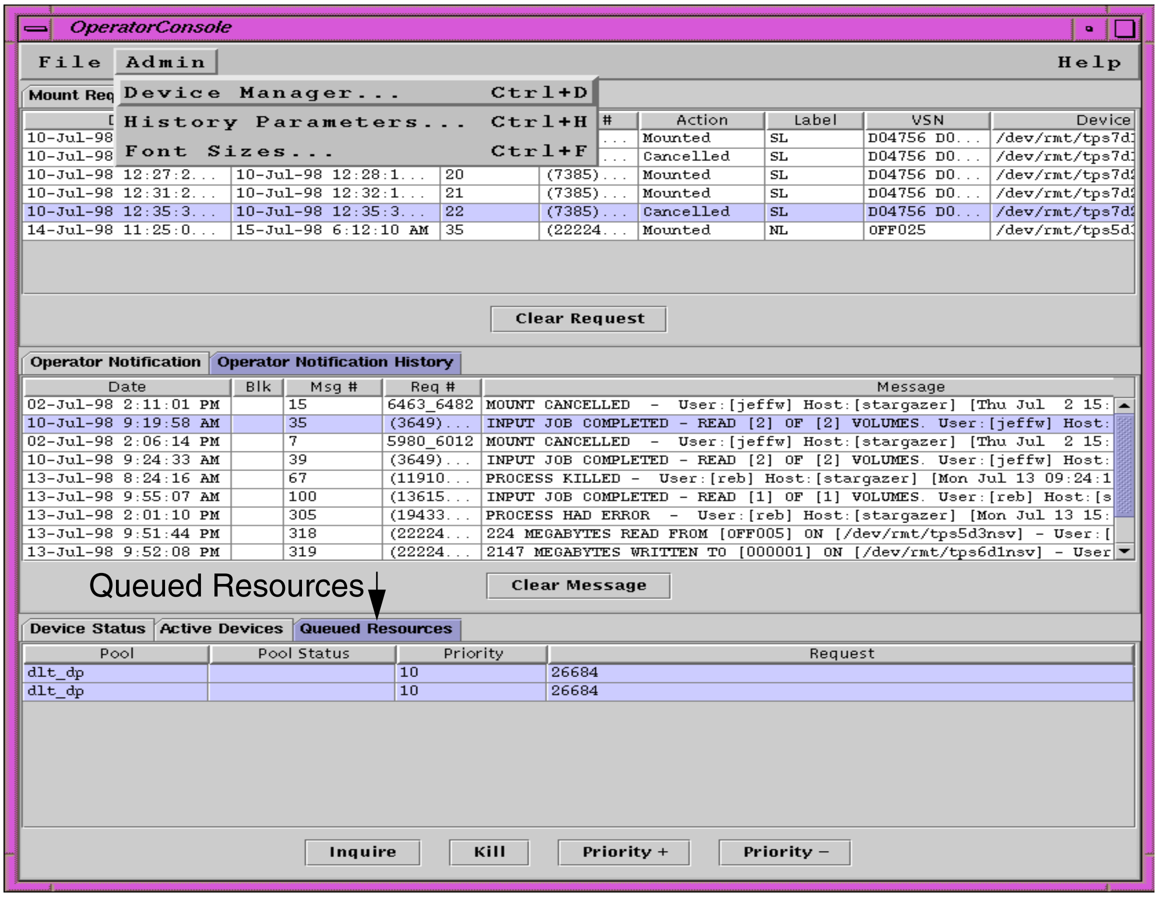
Active Devices

This tabbed pane of the **Device Manager** lists all devices currently reserved and in use. OpCon executes $PROMAX_HOME/sys/bin/pmxDev and uses the information from the **Active Requests** output listing of that command to update this pane. The **Priority** number is a PMX priority number between 0 (highest priority) and 19 (lowest priority); the default priority is 10. The number in the **Request** column is the PID of the executing "u#####" script that ProMAX stores in the flow directory during a job.

Inquire updates this display, as will clicking another one of the three **Device Manager** tabs and then clicking back on this tab.

Click **Kill** on a device to terminate the entire job associated with that device.

Queued Resources



[			#	Action	Label	VSN	Device
10-Jul-98			...	Mounted	SL	D04756 D0...	/dev/rmt/tps7d1
10-Jul-98			...	Cancelled	SL	D04756 D0...	/dev/rmt/tps7d1
10-Jul-98 12:27:2...	10-Jul-98 12:28:1...	20	(7385)...	Mounted	SL	D04756 D0...	/dev/rmt/tps7d2
10-Jul-98 12:31:2...	10-Jul-98 12:32:1...	21	(7385)...	Mounted	SL	D04756 D0...	/dev/rmt/tps7d2
10-Jul-98 12:35:3...	10-Jul-98 12:35:3...	22	(7385)...	Cancelled	SL	D04756 D0...	/dev/rmt/tps7d2
14-Jul-98 11:25:0...	15-Jul-98 6:12:10 AM	35	(22224...	Mounted	NL	OFF025	/dev/rmt/tps5d3

Date	Blk	Msg #	Req #	Message
02-Jul-98 2:11:01 PM		15	6463_6482	MOUNT CANCELLED - User:[jeffw] Host:[stargazer] [Thu Jul 2 15:
10-Jul-98 9:19:58 AM		35	(3649)...	INPUT JOB COMPLETED - READ [2] OF [2] VOLUMES. User:[jeffw] Host:
02-Jul-98 2:06:14 PM		7	5980_6012	MOUNT CANCELLED - User:[jeffw] Host:[stargazer] [Thu Jul 2 15:
10-Jul-98 9:24:33 AM		39	(3649)...	INPUT JOB COMPLETED - READ [2] OF [2] VOLUMES. User:[jeffw] Host:
13-Jul-98 8:24:16 AM		67	(11910...	PROCESS KILLED - User:[reb] Host:[stargazer] [Mon Jul 13 09:24:1
13-Jul-98 9:55:07 AM		100	(13615...	INPUT JOB COMPLETED - READ [1] OF [1] VOLUMES. User:[reb] Host:[s
13-Jul-98 2:01:10 PM		305	(19433...	PROCESS HAD ERROR - User:[reb] Host:[stargazer] [Mon Jul 13 15:
13-Jul-98 9:51:44 PM		318	(22224...	224 MEGABYTES READ FROM [OFF005] ON [/dev/rmt/tps5d3nsv] - User:[
13-Jul-98 9:52:08 PM		319	(22224...	2147 MEGABYTES WRITTEN TO [000001] ON [/dev/rmt/tps6d1nsv] - User

Pool	Pool Status	Priority	Request
dlt_dp		10	26684
dlt_dp		10	26684

This pane lists all pending mount requests and their device pools. OpCon executes $PROMAX_HOME/sys/bin/pmxDev and uses the information from the **Queued Requests** output listing of that command to update this pane.

As in the **Active Devices** pane, the **Priority** number is the PMX priority and the **Request** number is the PID of the executing "u#####" script that ProMAX stores in the flow directory during a job. Click **Inquire** to update this display, or click another tab and click back on this tab.

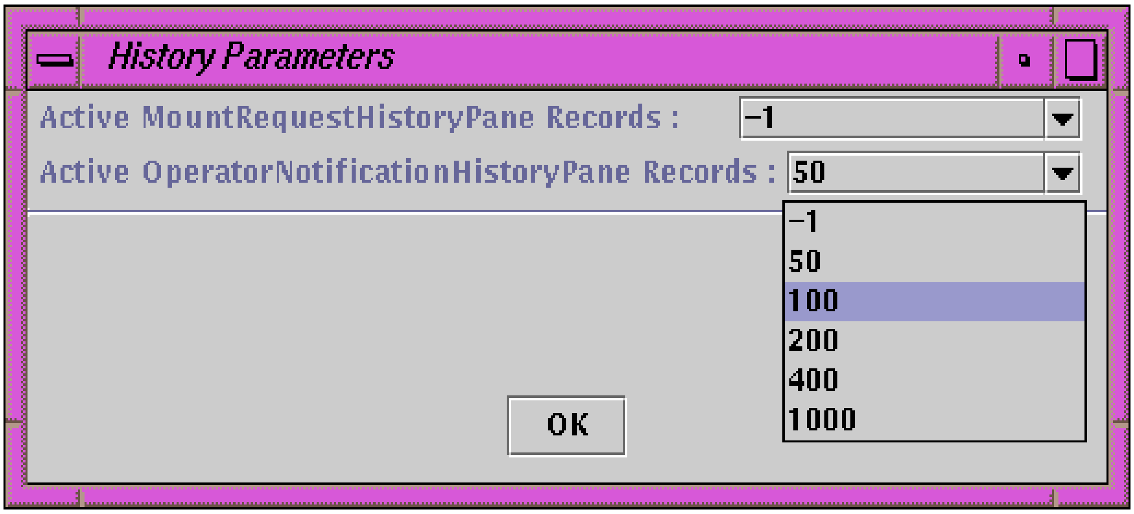
Click **Kill** on a device to terminate the entire pending job associated with that device.

Highlight a request and then click **Priority+** or **Priority-** to adjust the priority for that request (0 is the highest priority and 19 is the lowest). This will probably start this request before other requests requiring the same number of drive resources.

Menu Bar

Admin

- **Device Manager** toggles between opening and closing the third OpCon pane. However, we recommend keeping all three panes in view.
- **History Parameters** or **Ctrl-H**, displays the History Parameters window:


History Parameters

Active MountRequestHistoryPane Records : -1

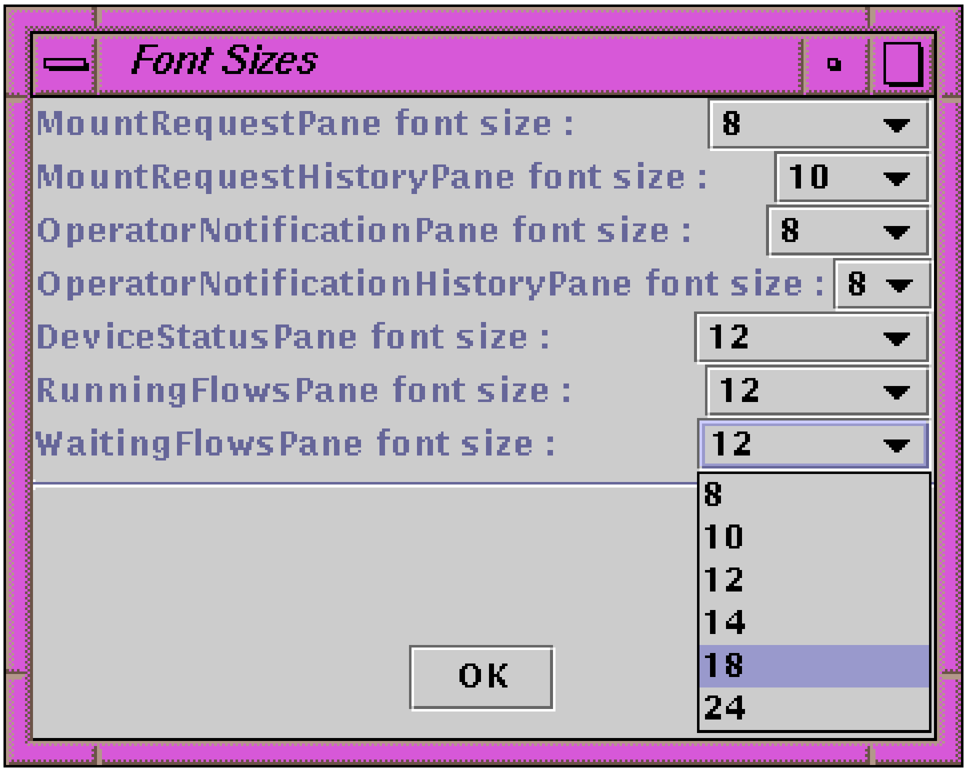
Active OperatorNotificationHistoryPane Records : 50

These settings set the maximum number of mount

Other Docs Known Problems

requests and operator notification messages to collect in the History panes. A value of **-1,** the default, collects an unlimited number of mount requests or operator notification messages or maximum memory storage.

- **Font Sizes** or **Ctrl-F** displays the Font Sizes window to customize the font sizes for all seven OpCon tabbed panes:



MountRequestPane font size : 8
MountRequestHistoryPane font size : 10
OperatorNotificationPane font size : 8
OperatorNotificationHistoryPane font size : 8
DeviceStatusPane font size : 12
RunningFlowsPane font size : 12
WaitingFlowsPane font size : 12

8
10
12
14
18
24

Help

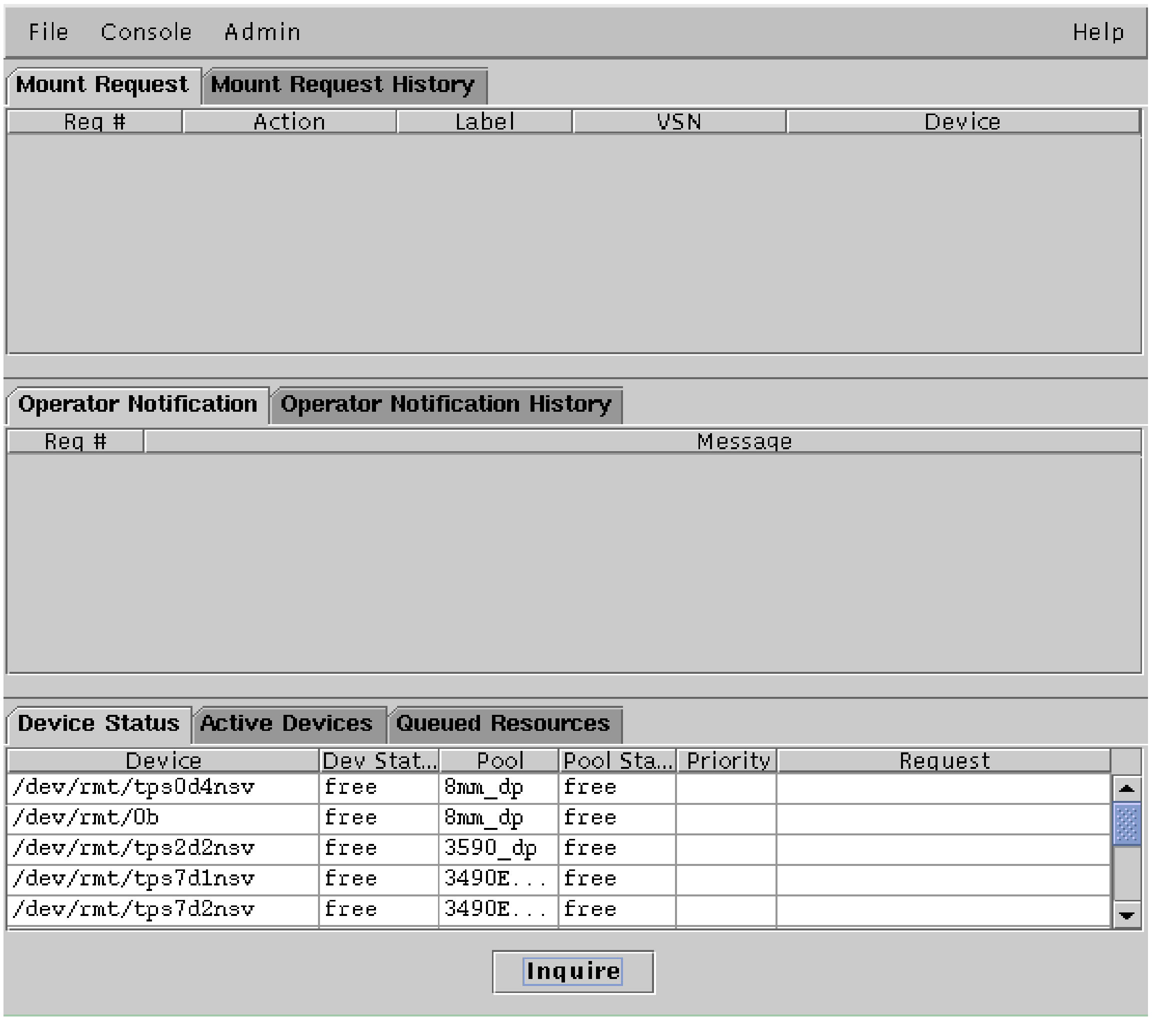
This button at the upper right of the OpCon window brings up the following help:

- **Console** or Ctrl-H within the window displays this Operator Console UI Guide documentation.
- **Catalog** displays the main TTS documentation.
- **Devices** displays the PMX Guide.

OpCon Console Monitor

If an OpCon user interface is already running on a system, you can view mount message and device status by running

$PROMAX_HOME/port/bin/ConsoleMonitor. Panes are the same as for the OpCon interface, but no job, mount or device interaction is available. **Console** ➛ **Message** or Ctrl-M displays a text box that you can enter and send a message to the Operator's **Operator Notification** window.



Device	Dev Stat...	Pool	Pool Sta...	Priority	Request
/dev/rmt/tps0d4nsv	free	8mm_dp	free		
/dev/rmt/0b	free	8mm_dp	free		
/dev/rmt/tps2d2nsv	free	3590_dp	free		
/dev/rmt/tps7d1nsv	free	3490E...	free		
/dev/rmt/tps7d2nsv	free	3490E...	free		

TopCat Guide

This is a guide to installing and using of the TopCat tape catalog. The installation instructions in this document conclude with command line tests. These tests validate that the system is correctly setup and functioning. However, you should also validate the setup by first running a few ProMAX tape jobs.

In This Chapter

Installation and Setup

This chapter describes how to configure and use the TopCat tape catalog system. If you are not using the TopCat catalog, you can skip this chapter.

We provide instructions for:

- Defining and Initializing Tape Device for Linux
- Configuring the TopCat
- Prelabeling tapes
- Configuring secondary or imported tcat catalogs
- Validating the Configuration
- Converting Tcat to TopCat

Sites that used the ***tcat*** catalog in prior releases should note that ***tcat*** is obsolete and will not work with this release. Later in this chapter we outline the steps to import your ***tcat*** catalog into the ***TopCat*** catalog.

Configuring TopCat

A ***TopCat*** Tape Catalog is comprised of the following file triplets. These files exist in the directory identified by the environmental variable, **PROMAX_TOPCAT_HOME:**

- *catadmin* describes the attributes and access control for the catalog.
- *catfname.catdata* is the actual catalog data file.
- *catfname.catindx* describes the media pools and volume ranges known to the catalog.

Relocate the TopCat Directory

An example *catadmin* file is provided in the $PROMAX_HOME/TopCat directory. If you are setting up the catalog for the first time, copy this TopCat directory to another location, independent of any ProMAX installation. This allows you to use the same Catalog with $PROMAX_HOME directories that are installed later. This directory does not contain ***catfname*.catdata** or

***catfname*.catindx** files. These are created during the configuration process.

As with any crucial data files, we strongly advise that your TopCat directory be included in your nightly system back-ups.

Set PROMAX_TOPCAT_HOME Environmental Variable

Include the PROMAX_TOPCAT_HOME environment variable in your ProMAX environment (either your ProMAX UI startup script, .cshrc or equivalent). If PROMAX_TOPCAT_HOME is not defined, the tape system tries to access the tape catalog in $PROMAX_HOME/TopCat/. For this reason, we do not include a **live** catalog with the installation.

Edit the catadmin file

An example ***catadmin*** tape catalog configuration file is in the $PROMAX_HOME/TopCat/ directory. The following is an edited example of a ***catadmin*** file; the details for editing each entry follow this example:

```
CATALOG

#  Comments begin with a '#' in column 1
   CATNAME:    "topcat_catalog"
   CATFILE:    "seismic_proc"
   CATDESC:    "ProMAX Seismic Processing"
   CATLINGER:  "days_linger_time"
   CATLOGFIL:  "logfile_name"
   CATADMIN:   "joe,karen,proadm"
#  CATRDONLY


   CATRESTRICT
     "VSN"      READ
     "DSN"      READ
     "VOLSEQ"   READ
     "NXTVOL"   READ
     "PRVVOL"   READ
     "PASSCODE" NONE
    CATEND


   CATPOOL
     POOLNAME:   "3490_fuji"
     POOLDESC:   "3490 Fujitsu tape pool"
     DEVPOOL :   "3490_dp"
#    GRPACCESS:  "staff"
  CATEND
```

```
      CATPOOL
          POOLNAME:   "8mm_expl"
          POOLDESC:   "8mm exploration tape pool"
        DEVPOOL :   "8mm_dp"
        GRPACCESS:  "exploration"
       CATEND


       CATPOOL
        POOLNAME:   "dlt_stk"
        POOLDESC:   "DLT StorageTek Robot"
        DEVPOOL :   "dlt_dp"
       CATEND


    CATEND
       CATINCLUDE:catadmin.tcat
       CATINCLUDE:catadmin.3dfield
```

- CATNAME is the catalog name appearing in the menu selection lists.
- CATFILE is used in the catalog's UNIX filenames, such as *seismic_proc.catindx* and *seismic_proc.catdata*.
- CATDESC describes the catalog in the ShowCat UI.
- CATLINGER specifies the number of days to wait before returning a volume to scratch status after a tape dataset is deleted.
- CATLOGFIL is the UNIX filename where all TopCat transactions are recorded.
- CATADMIN is the list of usernames with administrative privileges in accessing catalog files.
- CATRDONLY / CATEND - see below.
- CATRESTRICT / CATEND - see below.
- CATPOOL / CATEND - see below.
- CATINCLUDE identifies the name of a secondary catalog that is accessed via the main catalog.

Define CATALOG Stanzas

The *catadmin* file contains several general keyword and setting entries and several CATxxx/CATEND pairs. These pairs enclose stanzas of related keywords and settings. Preceding spaces and tabs are used only to allow easier viewing and editing.

The ***catadmin*** file of the primary catalog includes multiple CATINCLUDE entries for *including* the catalog administration files of other secondary TopCat catalogs. This allows datasets to be selected from either ShowCat or the ProMAX menus. Field tapes are often registered in a secondary catalog that has read only status.

In this example, two secondary catalogs are included in the TopCat catalog: ***catadmin.tcat*** and ***catadmin.3dfield***. The secondary catalog referenced by ***catadmin.tcat*** can contain all volumes imported from the old ProMAX tcat catalog, while the secondary catalog referenced by ***catadmin.3dfield*** can contain all 3D field data tapes. Field data tapes are logically separated from the primary catalog tapes, allowing tapes to be set to scratch status for new data to be written to them.

For a more detailed list of the ***catadmin*** file definitions, see ***catadmin File Catalog Definitions*** in the TopCat User guide.

Define CATPOOL Media/Pool Stanzas

A POOL definition consists of a combination of the keywords POOLNAME, POOLDESC, DEVPOOL and GRPACCESS, contained within CATPOOL/CATEND pairs:

```
CATPOOL
     POOLNAME: "pool_name"
     POOLDESC: "pool description"
     DEVPOOL: "device_pool"
     GRPACCESS: "group1,group2 ..."
CATEND
```

- POOLNAME: "pool_name" is the reference name stored in the MPOOLNAM catalog field.
- POOLDESC: "pool description" is the media pool descriptive name. This is displayed by the ShowCat UI.
- DEVPOOL: "device_pool" is the device pool name that reads or writes the media described by this media pool definition. It must match the value of DPOOL in the ***device.tts*** file.

- GRPACCESS: "group1,group2 ... is a list of group login names, belonging to the specified UNIX group ID (gid), having access to this media. CATADMIN authority does not override GRPACCESS. Therefore, catalog administrators do not have access to data contained within a media pool unless they are also a member of that group.

You should decide how you want to associate tape media pools and tape drives before you define the CATPOOL stanzas. A media pool is a name given to a group of tapes of the same type. A given media pool can only be used on devices appropriate for that medium. To ensure the correct associations, the value for DPOOL in the ***device.tts*** file must be the same as the value for DEVPOOL in the relevant CATPOOL stanza.

You should have one CATPOOL/CATEND stanza pair for each media pool defined.

In our example, three different media pools are defined by the POOLNAME parameter:

- **3490_fuji** set up as 3490 tapes on Fujitsu drives
- **8mm_expl** is allotted to the exploration department
- **dlt_stk** describes a pool of DLT tapes in a robotic tape system

For a complete list of pool definitions, see ***Pool Definitions***.

Populate the Catalog

When the media pools are defined, use the topcat command option, POPULATE, to initialize volume serial numbers (VSNs) for the tapes in the catalog. Like most **topcat** command options, you can use the POPULATE option directly from the command line, or use the topcat CMDFILE option to read parameters from an ACII text file. POPULATE the catalog for each of the media pools you defined in the ***catadmin*** file.

Command Line Option

The command line topcat execution:

```
> topcat 'POPULATE[POPMPOOL:"8mm_expl" POPVRNG:"700000-
        701000"]'
```

initializes VSNs 700000 to 701000 with a MPOOLNAM of **8mm_expl**. This example assumes POOLNAME: 8mm_expl was defined in your *catadmin* file.

CMDFILE option

The topcat CMDFILE option can be used by putting the following in an ASCII text file:

```
POPULATE[POPMPOOL:"8mm_expl" POPVRNG:"700000-701000"]
```

When executing these options from a file, you do not enclose the option string in apostrophes as is done for the command line execution. Instead, execute topcat with this file using the syntax (the file name *cmdfile.populate* is arbitrary):

```
> topcat 'CMDFILE:"cmdfile.populate"'
```

We include the following example command files (CMDFILE) with the $PROMAX_HOME/TopCat/ directory:

- *cmdfile.catinfo* to display catalog information
- *cmdfile.deldsn* to delete datasets
- *cmdfile.import* to import external catalogs
- *cmdfile.populate* to populate the catalog with new volumes
- *cmdfile.vdisp* to display catalog fields
- *cmdfile.vedit*. to edit catalog fields

In these command files are several lines of instructional comments which have a # in column one. To execute properly you must specify a valid value for USECAT and specify only one type of function. Using the command file method ensures valid syntax.

Files Created with POPULATE

When you POPULATE a new catalog for the first time, the following files are created in your TopCat directory:

- **.catdata*
- **.catindx*

For example, in your *catadmin* file, if CATFILE is set to **seismic_proc**, after successfully running the POPULATE command, the files *seismic_proc.catdata* and *seismic_proc.catindx* are created.

Validate the Catalog VSNs using Showcat

After creating the TopCat catalog, validate the setup using the ***showcat*** script. The ***showcat*** script runs on any host with access to the TopCat Catalog directory files.

1. Set the environment variable PROMAX_TOPCAT_HOME to the location of your catalog directory.
2. Start the ShowCat Tape Catalog UI by executing the script:

   ```
   > $PROMAX_HOME/port/bin/showcat &
   ```

   Your primary catalog and all included catalogs defined via the catinclude option should appear as entries in the catalog browser.
3. Click the toggle icon next to a catalog name to view a drop down list of available queries and summary information.
4. Right click on **Media** to display a list of media types and associated device pools defined in a catalog.
5. Right click on **Volumes** to view VSN ranges populated in each media pool.
6. Right click **Queries** to create a new query in the query editor.
7. Edit or execute previously saved queries by selecting **Edit** or **Run** from a right click popup on a query name.

Saved queries are stored in $HOME/.savedQueries/. This directory is automatically created by ShowCat.

If all users are to share the same query definitions, edit the "-DstoredQueryDirectory=" setting in the $PROMAX_HOME/port/bin/showcat script:

```
> java -DstoredQueryDirectory=${HOME}/.savedQueries ...
```

See ***Using the ShowCat Catalog*** for more details on how to access catalog and create catalog queries.

Prelabeling catalog tapes

When writing data to a volume in the catalog, the VSN must be a labeled tape. We provide the following script to facilitate prelabeling tapes:

```
> $PROMAX_HOME/port/bin/tapediag_prelabel device_pool_name vsns
```

- **device_pool_name** is the name of the device pool matching the media type (MPOOLNAM) of the VSNs to label.
- **vsns** is the set of VSN numbers to write. Enter on the command line as a space-delimited list, such as **000001 000008 000222**, or a min-max form, such as **888001-888088**.

Prelabeling can be done before or after VSNs are POPULATEd to the Catalog. For easier management, we urge you to be careful to keep the catalog population and the labeled tape pool synchronized.

ProClient CONFIG="SMNT" Mode

This mode is recommended when using stacker devices used for writing data in an unattended mode.

When CONFIG = "SMNT" is set for a device in the ***device.tts*** file, the ProClient Scratch Mount option is enabled. This option is used for manual devices, like stackers, and must not be used for robotic devices. When SMNT is set for a device or pool, ProClient operates in a **user scratch select** mode; that is, the tape mount message asks for a VSN of "SCRATCH" rather than a specific VSN number. You select the scratch tape to mount. That is, the operator can mount any tape with a "SCRATCH" status in the catalog, subject to the following system verifications:

- that the tape has a VSN matching the TopCat catalog's VSN, media pool, and device pool associations
- that the VSN's volume status attribute, VOLSTAT, is SCRATCH.

Because the SMNT mount option verifies the tape label, all scratch tapes must be prelabeled and have a status of SCRATCH. Use the ***$PROMAX_HOME/port/bin/tapediag_prelabel*** script to prelabel scratch tapes. After labeling the tapes, use the ShowCat UI to change the status of newly labeled tapes from NEW to SCRATCH. SMNT mode does not work if the VOLSTAT=" NEW"; it must be "SCRATCH".

Prelabeled tapes ensures that a stacker device is loaded with known SCRATCH tapes. It also ensures that the system writes to those tapes without operator intervention and without reserving specific tapes ahead of time. This prevents continuous monitoring tape devices for overnight or weekend operations.

If CONFIG = "SMNT" is not set when writing to manual devices, ProClient operates in a **catalog scratch select** mode. When a scratch volume is needed, the system selects the first available VSN that has VOLSTAT= "SCRATCH", and a mount request is issued for that specific VSN. The next scratch VSN is always the next sequential VSN in the Catalog with a VOLSTAT of SCRATCH.

If a non-labeled tape is mounted, this mount mode is satisfied, and standard labels are written to the tape, followed by the requested output data. For this reason we strongly advise that any non-labeled field tapes be physically ***write-protected*** to ensure that important data is not accidently overwritten.

Configuring secondary or imported tcat catalogs

Many sites find that a single primary TopCat catalog handles their data management requirements, and have no need for secondary catalogs. However, there is a benefit in having separate secondary if you are:

- isolating field data tapes in a separate read-only catalog for an additional level of data security
- maintaining an old tcat catalog
- separating a group of volumes from others via a catalog

Skip this section if you do not need a secondary catalog or do not need to import a previously existing tcat catalog.

This section describes how to use these topcat options to convert an existing tcat tape catalog and import it as a secondary TopCat catalog. These instructions are the same for defining other non-tcat secondary catalogs of tapes; just skip ahead a few paragraphs to *Run tcat to export a catalog dump file* for importing non-tcat catalogs. ***Using the TopCat Tape Catalog*** provides detailed information on the IMPORT and POPULATE command line options of the topcat program.

Make a Temporary Directory

Make a temporary TOPCAT_HOME directory to create a ***catadmin*** file for the secondary catalog and for outputting the ***.catdata*** and ***.catindx*** catalog files. ***$PROMAX_HOME/TopCat.tmp*** can exist in any directory. However, temporarily set the

PROMAX_TOPCAT_HOME environment variable to this directory.

Caution: If you forget to reset the PROMAX_TOPCAT_HOME environment variable, all of the following steps will be performed on your primary catalog.

Set Up a catadmin File

You need a ***catadmin*** file for the temporary TopCat directory. To add the file, copy the ***catadmin*** file from the primary catalog directory into this directory. Assuming your primary TopCat catalog is in the default location $PROMAX_HOME/TopCat/:

```
> mkdir $PROMAX_HOME/TopCat.tmp
> setenv TOPCAT_HOME $PROMAX_HOME/TopCat.tmp
> cp $PROMAX_HOME/TopCat/catadmin
   $PROMAX_HOME/TopCat.tmp
```

Each secondary catalog can have settings that are independent of the those defined by the ***catadmin*** file of the primary catalog. However, you must change the CATNAME, CATFILE, CATDESC fields. Secondary catalogs cannot have CATINCLUDE entries. That is, there is no support for tertiary catalogs. Set CATRDONLY, for this catalog to be **read-only**; data cannot be written to volumes included in a read-only catalog.

Run tcat to export a catalog dump file

Skip this step if you are defining other non-tcat secondary catalogs.

Point the PROMAX_PORT_TCAT_HOME environment variable to your old tcat directory. To dump standard ProMAX AL catalog tapes, run the following:

```
tcat AL output_file_name
```

or, to dump SL catalog tapes, run:

```
tcat SL output_file_name
```

The tcat catalog information is dumped into ***output_file_name*** and ***output_file_name.catmap*** as ASCII text files. The ***.catmap*** file defines the order of the TopCat catalog fields that are imported from the dump file.

For example, if you enter ***tcat*** as output_file_name, your output files are named: ***tcat*** and ***tcat.catmap***. ***tcat.catmap*** has a third line called MAPFLDS. It contains the following list of

catalog fields delimited by the pipe "|" character:
VSN|LABEL| FMTNAME| MPOOLNAM| VOLSTAT| NXTVOL| PRVVOL| VOLSEQ| UNAMCREAT| GRPCREAT| GRPOWN| DSCDATE| CREATSTAT| MNSEQNO| MXSEQNO| MNPORD| MXPORD| DATATYPE| PRYORD| DOMAIN| FLOWCREAT| DSN.

Note: Do not edit the ***.catmap*** file. You should not edit the ***tcat*** file, except for the fourth field, MPOOLNAM, the media pool name.

The ***tcat*** output file contains the actual catalog dump. The MPOOLNAM field that is output in this dump file is directly translated from its MEDIA entry in the old tcat catalog. Every MPOOLNAM type defined in this file must match a corresponding POOLNAME of a CATPOOL stanza in your ***catadmin*** file of the ***$PROMAX_HOME/TopCat.tmp*** directory.

If you want to change the pool names in this dump file to match the pools defined in your ***catadmin*** file, carefully edit this file and make the global substitutions now. The following tcat case-sensitive MEDIA field values can be exported to the dump file: **9track**, **1/4**, **1/2**, **8mm**, **dat**, **c2**, **tk50**, **dlt**, and **unknown**.

The easiest way to make sure your TopCat media pool names match those used previously in the old tcat catalog is to match the POOLNAME entries in the ***catadmin*** file to the media names in the dump file.

Initialize the VSNs in the catalog

Execute the POPULATE command from either the command line or by entering the command in a CMDFILE to initialize the VSNs that are in the imported catalog (allocating empty fields for each VSN). In order to populate the catalog, you need to know the VSN ranges of the catalog you are importing, including all discontinuous gaps; this can require looking through the VSN ranges in the ***tcat*** dump file and sorting the dump file. For example, if you had two different media in the old tcat catalog, **8mm** and **1/2**, you can run the following POPULATE commands:

```
> topcat 'POPULATE[POPMPOOL:"8mm" POPVRNG:"000001-
  000400,002000-004000"]'
> topcat 'POPULATE[POPMPOOL:"1/2" POPVRNG:"500001-
  500300"]'
```

This initializes the catalog with an **8mm** media pool having two VSN ranges and a **1/2** media pool having one contiguous range of VSNs.

Execute the IMPORT Command

Execute the topcat command to IMPORT your prior catalog. If you exported the tcat *label information file* dump filename as ***tcat***, enter the topcat command:

```
> topcat 'IMPORT[IMPFILE:"tcat" IMPMAP:"tcat.catmap"]'
```

This creates ****.catdata*** and ****.catindx*** TopCat catalog files. These names are defined by the CATFILE entry in the ***catadmin*** file. For example, if your CATFILE entry is tcat, then ***tcat.catdata*** and ***tcat.catindx*** are created in the ***$PROMAX_HOME/TopCat.tmp*** directory.

Move the catadmin files to $PROMAX_TOPCAT_HOME

Reset the PROMAX_TOPCAT_HOME variable in your environment to point to your primary TopCat directory.

Copy or move the ***.catdata*** and ***.catindx*** catalog files from the temporary PROMAX_TOPCAT_HOME directory to the main $PROMAX_TOPCAT_HOME directory. The ***catadmin*** file in the temporary PROMAX_TOPCAT_HOME directory must be renamed to ***catadmin.newname*** before or when copying it to the primary catalog directory. If you do not do this you may overwrite the ***catadmin*** file of your main catalog.

Caution: Do not forget to rename your secondary ***catadmin*** before copying it to your main PROMAX_TOPCAT_HOME directory.

The following is an example of PROMAX_TOPCAT_HOME:

```
$TOPCAT_HOME> ls

catadmin

catadmin.tcat

promax.catdata

promax.catindx

tcat.catdata

tcat.catindx
```

The filename ***catadmin*** is a reserved catalog administration filename. It defines the primary catalog and has a CATFILE entry of **promax**. The filename *catadmin.tcat* is the catalog administration file of the secondary catalog with a CATFILE entry of *tcat*. This points to the *tcat.catdata* and *tcat.catindx* secondary catalog files.

Include the Secondary Catalog in the Primary Catalog

Include the secondary catalog as a part of the primary catalog by adding a CATINCLUDE line to the bottom of the *catadmin* file:

```
CATINCLUDE:catadmin.tcat
```

To add additional secondary catalogs, repeat the four steps, and include another CATINCLUDE line to the bottom of the *catadmin* file.

You can delete the temporary PROMAX_TOPCAT_HOME directory or leave it as a backup record.

Validating Your Configuration

Run a few output jobs using **Synthetic Trace Generation** or **Disk Data Input**, and **Tape Data Output**. Then read the tape datasets. Use ShowCat to see what information is entered in the catalog for tape datasets. Check the *job.output* files to get familiar with the information made available. Also try reading in a few field tapes and long-block field data.

When you delete a cataloged tape dataset from the datasets list of the ProMAX UI, you are asked if you want to delete the dataset from the catalog. If you enter **yes**, the volume is set to SCRATCH status in the catalog. If you enter **no**, the dataset remains in the catalog (and on tape) and can be accessed using **Tape Data Input** and choosing that the dataset is in the catalog but not in the datasets list.

Converting Tcat to TopCat

You must convert your Tcat catalog to the Topcat catalog. The following steps describe how to convert your old catalog:

1. setenv PROMAX_HOME
2. setenv PROMAX_PORT_TCAT_HOME
3. setenv PROMAX_TOPCAT_HOME
4. verify you have read/write permissions in your old Tcat and your new TopCat directory
5. run the script, $PROMAX_HOME/port/bin/Import_TCAT

Catalog Definitions and References

TopCat is the ProMAX tape catalog. Its usage is based on a catalog home directory, a topcat program which updates, queries, and displays the catalog contents, a showcat Java user interface, and client programs which access the catalog using topcat.

Directory Environment Variables

To determine the pathname to the tape catalog home directory, the catalog searches for the following two environment variables, in the order listed:

- PROMAX_TOPCAT_HOME is the absolute pathname to the tape catalog directory.
- PROMAX_HOME looks for a $PROMAX_HOME/TopCat directory.

Home Directory Files

The following are the files found in the TopCat home directory:

- **catadmin** is a reserved filename, this catalog administrator file contains catalog configuration information.
- **catfname.catindx** is the catalog index file. It contains VSN range information for quick VSN information retrieval.
- **catfname.catdata** is the catalog data file. Contains VSN record information.

Note: *catfname.catindx* and *catfname.catdata* are a matched pair of files; one cannot exist without the other. The actual catalog filename (*catfname*) is set in the catadmin file. The file extensions are set internally (and are always *.catindx* and *.catdata*).

Catfname points to a subdirectory within PROMAX_TOPCAT_HOME if the subdirectory already exists; the catalog does not create directories. *Catfname* is combined with a CATALOG path to resolve the location.

TopCat Tape Catalog Attributes

The following is a listing of the 85 attributes in the TopCat tape catalog, matched with any corresponding fields in the old 28-attribute tcat tape catalog.

Catalog Attribute	Description	Length	Data Type	TCAT Attribute	TCAT Description
VSN #define 0	Volume serial number.	6	ASTRN	SERIALNO	Volume serial number. base26 string
BORN #define 1	Date when VSN was initialized.	6	DATE		
NACCESS #define 2	Number of times this VSN has been mounted.	9	INT		
VOLSTAT #define 3	Volume status. (active | scratch | reserved | offsite | other)	16	NBSTRN	STATUS	Status of volume. (ACTIVE | RESERVED | SCRATCH | NEW | OFFSITE| NOTLOCATED | READONLY | BAD)
STATDATE #define 4	Date status was changed.	6	DATE		
PRSTAT #define 5	Previous volume status.	16	NBSTRN		
LABEL #define 6	NL | AL | SL)	4	NBSTRN		
MPOOLNAM #define 7	(Media Pool Name)	24	GSTRN	MEDIA	Media type. (Integer: 0-15)

Catalog Attribute	Description	Length	Data Type	TCAT Attribute	TCAT Description
LOCATION #define 8	Location of tape: rack, robot, etc.	24	GSTRN	SLOT_NO	Tape rack slot number. (Integer)
GRPOWN #define 9	Group Id which owns the tape.	16	SECUR		
SDSN #define 10	Super dataset name; used for logical grouping.	64	GSTRN		
DSN #define 11	Dataset name - user creation name.	64	GSTRN	DESCRIPT	Dataset description search key ('30-character dataset name' 14-char_area/ 14-char_line / 8-char_unixname)
PASSCODE #define 12	Dataset entry key; owner may change value.	24	SECUR		
DSCDATE #define 13	Date when dataset was created.	6	DATE	DATE_CRE	Date created. (String: day month date time year)
DSEXPIRE #define 14	Dataset expiration date.	6	DATE		
DSVERSION #define 15	Dataset version number.	6	INT		

Catalog Attribute	Description	Length	Data Type	TCAT Attribute	TCAT Description
CREATSTAT #define 16	Dataset creation status. (NORMAL | KILLED| NOTCLOSED | ABORTED)	16	NBSTRN	CLOSED_P	Volume-closed-properly flag. (Integer: 0 | 1 | null)
UNAMCREAT #define 17	User Id which created the DSN.	24	SECUR	UID	User Id. (String)
GRPCREAT #define 18	Group Id which created the DSN.	12	SECUR	GID	Group Id. (Integer)
HOSCREAT #define 19	Host machine where dataset was created.	24	NBSTRN		
DRIVCREAT #define 20	UNIX device name where dataset was created.	128	NBSTR		
SYSNAME #define 21	Software name and version that created dataset.	24	GSTRN		
FLOWCREAT #define 22	Flow path that created dataset. area/line/flow	128	GSTRN		
VOLSEQ #define 23	Sequential volume number within the dataset(zero based).	8	INT	NXT_IBKT	Next catSeqno in search list. (Integer: 0 | catSeqno)

Catalog Attribute	Description	Length	Data Type	TCAT Attribute	TCAT Description
NXTVOL #define 24	Next VSN in dataset. The last VSN in the dataset will have a special reserved NXTVOL value of "LAST".	6	ASTRN	NEXT_VOL	Next catSeqno in dataset. (Integer: 0 | catSeqno | null)
PRVVOL #define 25	Previous VSN in dataset. The first VSN in the dataset will have a special reserved PRVVOL value of "FIRST".	6	ASTRN	FIRSTVOL	First volume in dataset flag. (Integer: 1 | null)
CURSTAT #define 26	Current volume status. (READY | OPEN | RESERVED)	12	NBSTRN		
CURPID #define 27	PID of current process using volume.	12	INT		
CADATE #define 28	Last date accessed.	6	DATE		
DSUSAGE #define 29	Number of times VSN has been accessed.	8	INT		

Catalog Attribute	Description	Length	Data Type	TCAT Attribute	TCAT Description
CURUSER #define 30	User Id who last accessed the VSN.	24	NBSTRN		
CURHOST #define 31	Machine hostname that last accessed the VSN.	24	NBSTRN		
DEVPA #define 32	UNIX device name that last accessed the VSN.	24	NBSTRN		
FLOWPA #define 33	Flow path that last attempted to access VSN, area/line/flow	128	GSTRN		
CMPPA #define 34	Job completion status of last access.	16	NBSTRN		
DATATYPE #define 35	Data type. (PRESTACK | STACKED | ARCHIVE| FIELD | LOG | SUPPORT | OTHER)	16	NBSTRN	DATATYPE	Data type. (String: field | unstacked | stacked | archiv)
VAULTFILE #define 36	If DATATYPE = archive, this is the archive file.	128	NBSTRN		
DOMAIN #define 37	Domain of dataset. (DEPTH | TIME | SPATIAL| OTHER)	16	NBSTR		

Catalog Attribute	Description	Length	Data Type	TCAT Attribute	TCAT Description
FMTNAME #define 38	Data format. (ProMAX | SEGD | SEGY | SIERRA | WESTERN4 | OTHER)	24	GSTRN		
BINFMT #define 39	Binary format of data. (IBM32 | IEEE | 2bInt | 4bInt | ASCII | OTHER)	16	GSTRN		
SMPLINT #define 40	Sample rate in ms.	16	NBSTN		
UNITS #define 41	Domain units (sample units)	16	GSTRN		
NSAMPLE #define 42	Maximum samples per trace.	12	INT		
NOUTBLKS #define 43	Number of output data blocks written to volume.	12	INT		
NOUTSIZE #define 44	Total capacity used (in bytes).	12	INT		
MNSEQNO #define 45	Minimum sequential trace number in dataset.	10	INT	MINSEQDS	Minimum sequential-trace-number-in-dataset.
MXSEQNO #define 46	Maximum sequential trace number in dataset.	10	INT	MAXSEQDS	Maximum sequential-trace-number-in-dataset.

Catalog Attribute	Description	Length	Data Type	TCAT Attribute	TCAT Description
PRYORD #define 47	Primary sort order of data.	16	GSTRN	PKEYNAME	Primary sort key name.
MNPORD #define 48	Minimum primary sort order ensemble number.	24	GSTRN	MIN_ENS	Minimum ensemble (primary sort key) number.
MXPORD #define 49	Maximum primary sort order ensemble number.	24	GSTRN	MAX_ENS	Maximum ensemble (primary sort key) number.
SNDORD #define 50	Secondary sort order of data.	16	GSTRN		
MNSORD #define 51	Minimum secondary sort order ensemble number.	24	GSTRN		
MXSORD #define 52	Maximum secondary sort order ensemble number.	24	GSTRN		
MNXCOORD #define 53	Minimum X-coordinate	24	GSTRN		
MXXCOORD #define 54	Maximum X-coordinate	24	GSTRN		
MNYCOORD #define 55	Minimum Y-coordinate	24	GSTRN		
MXYCOORD #define 56	Maximum Y-coordinate	24	GSTRN		
MNZCOORD #define 57	Minimum Z-coordinate	24	GSTRN		

Catalog Attribute	Description	Length	Data Type	TCAT Attribute	TCAT Description
MXZCOORD #define 58	Maximum Z-coordinate	24	GSTRN		
COMMENT #define 59	User comment field.	160	GSTRN		
PARNTDSN #define 60	Parent dataset name.	64	GSTRN		
PARDSVER #define 61	Version number of parent dataset.	6	INT		
AQAREA #define 62	Geographical area name.	80	GSTRN		
ORIGCLIENT #define 63	Original client company this data was acquired for	80	GSTRN		
AQCO #define 64	Name of data aquisition company.	80	GSTRN		
AQCREW #define 65	Aquisition crew information.	80	GSTRN		
SURVEYOR #define 66	Name of surveyor	80	GSTRN		
AQDATE #define 67	Acquisition date range.	16	DATE		
AQULEN #define 68	Acquisition units of measured lengths.	16	GSTRN		
AQUELV #define 69	Acquisition units for elev.	16	GSTRN		

Catalog Attribute	Description	Length	Data Type	TCAT Attribute	TCAT Description
VRTMCFAC #define 70	Acquisition vertical misclosure factor.	80	GSTRN		
HOZMCFAC #define 71	Acquisition horizontal misclosure factor.	80	GSTRN		
AZMMCFAC #define 72	Acquisition azimuth misclosure factor.	80	GSTRN		
ELIPSIOD #define 73	Ellipsoid.	80	GSTRN		
SMAJAX #define 74	Semi major axis.	16	GSTRN		
RECFLAT #define 75	1/f Reciprocal flattening.	16	GSTRN		
PROJECTN #define 76	Map projection.	32	GSTRN		
GRIDDESC #define 77	Grid description.	32	GSTRN		
ZONE #define 78	Zone.	80	GSTRN		
ORIGLAT #define 79	Latitude at origin.	24	GSTRN		
ORIGLONG #define 80	Longitude at origin.	24	GSTRN		
FALSEAST #define 81	False easting.	16	GSTRN		
FALSNORTH #define 82	False northing	16	GSTRN		
ORIGSF #define 83	Scale factor at origin.	16	GSTRN		

Catalog Attribute	Description	Length	Data Type	TCAT Attribute	TCAT Description
PROJUNIT #define 84	Projection units.	24	GSTRN		

Unavailable Attributes

The following attributes are not currently available:

- PMX_PROJ: "ProMAX project"
- PMX_AREA: "ProMAX area"
- PMX_LINE: "ProMAX line"
- DS_NAME: "ProMAX dataset name"
- SITE: "Site number" (Integer)
- MODE: "UNIX Permissions" (Integer: octal mode | null)"
- MIN_CDP: "Minimum CDP number"
- MAX_CDP: "Maximum CDP number"
- MIN_SIN: "Minimum SIN" (shot index number)
- MAX_SIN: "Maximum SIN" (shot index number)

catadmin File Catalog Definitions

The catalog administration file, ***catadmin***, contains catalog configuration information, usually set up by the catalog administrator. The CATALOG/CATEND stanza encloses everything except the CATINCLUDE entries.

CATALOG begins the catalog definition.

CATNAME: "robocat" is a catalog reference name. This is the catalog name referenced by the ProMAX I/O menus and by the ShowCat catalog UI.

CATFILE: "riocat" is a catalog file name. The catalog files used in this example are ***riocat.catdata*** and ***riocat.catindx***.

CATDESC: "main promax tape catalog" is the catalog description. It is displayed by the ProMAX I/O menus and by the ShowCat catalog UI.

CATINBUF: "15000" is the catalog memory buffer.

CATOUTBUF: "15000" is the catalog memory buffer.

CATLINGER: "5" is the number of days a deleted dataset lingers in the catalog before it is fully removed; a value of **0** implies instantaneous deletion. If a vvalue if **5** is set, when a dataset is deleted, all the volumes belonging to that dataset are not overwritten until 5 days after the dataset was deleted. The advantage of setting this linger period is that you can salvage the data if a dataset was accidentally deleted. The disadvantage is that the longer the CATLINGER period is set, the less available scratch tapes.

CATLOGFIL: "logfile_name" is the catalog log file name/path.

CATADMIN: "rocky,mjs,crs,prouser" is the list of authorized catalog adminstrator user ids; only users defined in this comma-delimited list can directly edit catalog fields from the ShowCat UI. This CATADMIN list also works in conjunction with the CATRESTRICT.

CATRDONLY specifies if this catalog is read-only; read-only catalogs cannot be modified. We suggest not setting the primary catalog as CATRDONLY, use it for secondary catalogs which are included via CATINCLUDE.

CATRESTRICT restricts catalog fields. Within the CATRESTRICT/CATEND stanza, you can define which of the 85 TopCat fields are either read-only (READ) or not-readable (NONE) to anyone not defined in the CATADMIN stanza. Access on each catalog field is restricted to either a read-only or no-access permission.

"catalog_field1" READ Read-only; anyone defined in CATADMIN list can write to these attributes.

"catalog_field2" NONE No read; anyone defined in CATADMIN list can read these attributes. Fields defined as NONE do not appear in Showcat for selection or display.

When you run ProMAX Tape Catalog I/O jobs, various TopCat fields are updated to reflect which volumes were used. The CATRESTRICT stanza defines which fields cannot be externally edited, such as from the ShowCat UI. Any undefined field in the stanza, or that is listed but commented out can be edited by all users. Letting all users directly edit various Catalog fields can potentially cause data loss if those edits result in the the loss of VSN associations to datasets.

Other Docs Known Problems

Note: We suggest leaving most of the 85 fields with READ set.

CATEND ends CATRESTRICT stanza.

CATPOOL described in the Pool Definitions section.

CATEND ends CATPOOL stanza.

CATEND ends CATALOG definition.

CATINCLUDE: "catadmin_filename1" is a secondary catalog catadmin file.

CATINCLUDE: "catadmin_filename2" is a secondary catalog catadmin file.

# This is a comment line. A "#" in column 1 comments out the line.

Pool Stanza Definition

A POOL definition consists of a combination of the keywords POOLNAME, POOLDESC, DEVPOOL and GRPACCESS, contained within CATPOOL/CATEND pairs:

```
CATPOOL
     POOLNAME: "pool_name"
     POOLDESC: "pool description"
     DEVPOOL: "device_pool"
     GRPACCESS: "group1,group2 ..."
CATEND
```

POOLNAME: "pool_name" is the reference name to be stored in the Catalog field of MPOOLNAM.

POOLDESC: "pool description" is the media pool descriptive name. It is displayed by the ShowCat Catalog UI.

DEVPOOL: "device_pool" is the device pool name that reads or writes the media described by this media pool definition.

Usage

This information is essential for end user convenience because the DEVPOOL stanza provides a mechanism for matching media and device, when a third party tape system (such as NetTAPE) is used. The main purpose of the DEVPOOL/POOLNAME matching is to resolve unknown information. If media pool is known, the device pool can be determined. Conversely, if the device pool is given, the matching media pool can be resolved.

When a device pool is picked from the ProMAX menu it must match the following, in order to find the correct dataset:

- secondary catalogs, defined with CATINCLUDEs in the main catadmin file, matches the Catalog name you select
- the name matches the device pool entry in your ***device.tts*** file and the ***PMX config_file***
- the device pool name matches the media described by the media pool definition. In other words, the CATPOOL which matches the selected DEVPOOL must have a POOLNAME entry that matches the MPOOLNAM Catalog value for the dataset.

If either the wrong secondary catalog or wrong device is selected, or if the derived media pool mismatches the media pool values in the Catalog, then the dataset may not be found. This is usually due to an erroneous configuration of the Catalog's catadmin file or the device.tts file pool entries.

Defining Group Access

GRPACCESS defines the user group names that have access to this media pool. Restrictive access is active only if GRPACCESS is defined.

GRPACCESS: "group1,group2" is a list of group login names, belonging to the specified UNIX group ID (gid), that have access to this media. CATADMIN authority does not override GRPACCESS. Therefore, catalog administrators do not have access to data contained within a media pool unless they are also a member of that group.

Example: Unlimited access

```
CATPOOL
    POOLNAME: "pool_name"
    POOLDESC: "exploration 3490"
    DEVPOOL: "3490_e"
CATEND
```

Volumes defined for this media pool are available for all users.

Example: Limited access

```
CATPOOL
    POOLNAME: "pool_name"
    POOLDESC: "exploration 3490"
    DEVPOOL: "3490_e"
    GRPACCESS: "group1,group2"
CATEND
```

Only users with group login name of "group1" or "group2" are granted access to this media pool.

Query and User Update Commands

USECAT: "catalog_reference_name"

Select which catalog to use; will use the default standard if not set.

Usage

catalog_reference_name is the CATNAME catalog reference name as defined in the catalog administrator's file. The values may be obtained from CATINFO[INFCATNAME].

CMDFILE: "cmdfile_name"

Filename containing TopCat catalog commands to execute.

Usage

cmdfile_name is the filename containing TopCat catalog commands to execute. Tabs and line breaks are removed from the input file and multiple lines are concatenated into one single string. The character "#" in column 1 comments out the line.

Example

```
- - - top of CMDFILE file - - -
USECAT: " stdocat"
POPULATE[ POPMPOOL: " 8mm " POPVRNG: " VOL001-
    VOL015,V1S0N1-V1S1N5 " ]
POPULATE[ POPMPOOL: " 8mm " POPVRNG:
"V00813
-V00822" ]
```

- - - end of CMDFILE file - - -

The resulting input command string from the above file is equivalent to the following TopCat command line execution:

```
topcat 'USECAT: "stdocat" POPULATE[ POPMPOOL: "8mm"
    POPVRNG: "VOL001-VOL015,V1S0N1-V1S1N5" ]'
```

and

```
topcat 'USECAT: "stdocat" POPULATE[ POPMPOOL: "8mm"
    POPVRNG: "V00813-V00822" ]'
```

Common commands may be stored in command files for reuse by other TopCat queries, just as TopCat invocations may be used within shell scripts.

CATINFO[INFCATNAM | INFMPOOL | INFSCAN | INFTYPES | INFLABEL | INFDESC | INFLIST]

Shows catalog information.

Options

INFCATNAM: List catalog name(s) (CATNAME, read/write status, CATDESC).

INFMPOOL: List defined catalog pools and the default if the catalog to use is not set (CATALOG["catalog_reference_name"]).

INFSCAN: Show summary of catalog index file (pool name, vsn range, sequential record range (zero-based), and number of volumes in range).

INFTYPES: List of defined catalog field types. Returns pairs (int string).

INFLABEL: Header string that may be used to describe columns.

INFDESC: List of field lengths, types, and descriptions.

INFLIST: Special lists for pre-defined field values.

Usage

The seven CATINFO options can be used in any order and their use is optional. If you were to execute:

```
topcat 'CATINFO[ INFLIST INFLIST INFLIST ]'
```

the INFLIST option would be executed 3 times.

Example

topcat ’CATINFO[INFTYPES]’

returns...

```
CATSTAT:GOOD
NITEMS: 7 MAX_SIZE: 96
0 GSTRN  General string input field - accept all characters.
1 ASTRN  Character set defined by ISO 646, A-Z 0-9 blank!%&’()*+‘-./:;
2 NBSTRN String with no blanks
8 INT    Integer value
9 FLT    Float
10 SECUR  Security or data access related field
11 DATE   Date field - dates are stored as julian
```

where the first column (0,1,2,8,9,10,11) is field type, the second column (GSTRN) is field name, and this is followed by a brief description.

Example

topcat ’CATINFO[INFLIST]’

returns...

```
NITEMS: 8 MAX_SIZE: 87
6   3 NL,AL,SL
3   8
ACTIVE,RESERVED,SCRATCH,NEW,OFFSITE,NOTLOCATED,READO
NLY,BAD
16   4 NORMAL,KILLED,NOTCLOSED,ABORTED
26   3 READY,OPEN,RESERVED
35   6 PRESTACK,STACKED,ARCHIVE,FIELD,LOG,SUPPORT,OTHER
37   4 DEPTH,TIME,SPATIAL,OTHER
38   6 ProMAX,SEGD,SEGY,SIERRA,WESTERN4,OTHER
39   6 IBM32,IEEE,2bInt,4bInt,ASCII,OTHER
```

where the first column (6,3,16,26,35,37,38,39) is the field number which uses predefined values, the second column

(3,5,4,3,6,4,6,6) is the number of values in the predefined list, and this is followed by the comma-delimited list values.

Example "topcat.h" entry for LABEL, field #6 (zero-based):

```
#define LABEL 6
LABEL ,"LABEL",4,NBSTRN,"Volume label type (NL,AL,SL).",
```

Example

topcat 'CATINFO[INFLABEL]'

returns...

```
CATSTAT:GOOD
NITEMS: 1 MAX_SIZE: 34
Fld# Nmeumonic Size Type    Description
```

Example

topcat 'CATINFO[INFDESC]'

returns...

```
CATSTAT:GOOD
NITEMS: 85 MAX_SIZE: 100
0       VSN   6  1 Volume serial number.
1      BORN   6 11 Volume creation date (born on).
2    NACCESS   9  8 Number of times this volume has been mounted.
3    VOLSTAT  16  2 Volume status (active|scratch|offsite|other).
4   STATDATE   6 11 Date status was changed.
5     PRSTAT  10  2 Previous status.
6      LABEL   4  2 Volume label type (NL,AL,SL).
7   MPOOLNAM  24  0 Media pool name.
...
```

where the first column (e.g. 0) is the sequential field number, the second column (e.g. VSN) is the field name, the third column (e.g. 6) is the maximum field size, the fourth column (e.g. 1) is the field type as defined by INFTYPES, and this is followed by a brief description.

QRY[...]

Query options set.

Other Docs Known Problems

Query Options

DISP["FLD1" "FLD2" "FLD3" ...]: Display set; describes which fields to display from a successful query.

QRYFILE: "queryfile": Use precomputed query list (from a previous QRYSAVE output) as input to this query.

QRYSAVE: "queryfile": Save query results to this temporary file. queryfile will only exist for the life of the QRY and is used to combine queries via QRYFILE and QRYAPPEND.

QRYAPPEND: "queryfile": Append query results to existing query results file.

QRYFLD: "FLD": Specifies catalog field for query.

QRYRETN: "n": Return n values of "kind".

Logical Operators

QRYAND[...]: ombines two atomic logical operations with "AND".

QRYOR[...]: Combines two atomic logical operations with "OR".

QRYEQ: "n": Return values which are equal.

QRYLT: "value": Less than atomic operator.

QRYLE: "value": Less than or equal atomic operator.

QRYGT: "value": Greater than atomic operator.

QRYGE: "value": Greater than or equal atomic operator.

QRYWCEN: "string": Wildcard equal to end of string.

QRYWCBG: "string": Wildcard equal to beginning of string.

QRYWCSUB: "string": Wildcard equal to first substring.

Usage

- Query operations may be cascaded.

  ```
  QRY[ ]QRY[ ]QRY[ ]...
  ```

- Query results of cascading queries automatically become input to the next query operation.

  ```
  catalog disk read -> QRY[ ] -> QRY[ ] -> QRY[ ] -> ...
  ```

- Query results may be saved to a file and selected for query input.

  QRY[... QRYSAVE: "qry_a"] -> query results saved to file qry_a

  QRY[QRYFILE: "qry_a" ...] -> previous query results in file qry_a are input to new query

- Query results may be appended to existing query file, combining (logical OR) results of unrelated querys.

  QRY[... QRYSAVE: "qry_a"]

  QRY[QRYFILE: "qry_a" ... QRYSAVE: "qry_b"]

  QRY[QRYFILE: "qry_a"... QRYAPPEND: "qry_b"]

  QRY[QRYFILE: "qry_b" ...]

In this example, query results are saved to file qry_a. The query result file qry_a is used as input to the second query operation and a second query results file, qry_b, is generated. The third query operation also uses qry_a as input but appends its query results to the file qry_b. Finally, the results file qry_b are used as input to the fourth and final query.

- If multiple atomic logical operations are used, these operations must be grouped by either the QRYAND or the QRYOR operator. The atomic operation "QRYEQ" is implicitly assumed to be an OR.

Inclusive range: value1 > value < value2

QRY[... QRYAND[QRYGT: " value1" QRYLT: "value2"]]

Exclusive range: value < value1, value > value2

QRY[... QRYOR[QRYLT: "value1 " QRYGT: "value2 "]]

Multiple exact selection: (implicit OR assumed)

QRY[... QRYEQ: " value1" QRYEQ: "value2"]

Multiple atomics: (implicit OR assumed)

QRY[... QRYAND[ ] QRYAND[ ] QRYOR[ ] QRYEQ: " value1" QRYEQ: "value2"]

- QRYWC - Query wild cards are implicitly assumed to be an EQUAL atomic.

  QRY[QRYFILE: "qry_results.a" QRYFLD: "DSNAME" QRYWCSUB: "my dataset" QRYRETN:" 1 "]

Example

In this example, all VSNs with ACTIVE status will be reported with the five listed attributes. topcat output is to stdout with the hidden character Ctrl-V delimiting fields. The ’sed’ command shown here changes the Ctrl-V to a Tab. When entering the command, type Ctrl-Ctrl-V (rather than ^V) and the 'Tab' key (rather than spaces). The '>' uses UNIX redirection to save the output to a disk file.

```
topcat 'CMDFILE:" q_active"' | sed 's/^V/   /g' > active_VSNs
 - - - q_active cmdfile - - -
USECAT: " stdocat"
QRY[QRYFLD:"VOLSTAT" QRYEQ:"ACTIVE"]
DISP["VSN" "DSEXPIRE" "UNAMCREAT" "VOLSEQ" "DSN"]
- - - active_VSNs output file - - -

VSN DSEXPIRE UNAMCREAT VOLSEQ DSN
000001  273221  dave    0     SALT3D.THOMPSON3D.DECONTEST
000015  001140  roger   0     HAL3D.YOUNGER73.80962648
000020  027078  prouser 0      BBK.SW1L1.GEOMSHOTS
000021  027078  prouser 1      BBK.SW1L1.GEOMSHOTS
000022  199365  roger   0     HAL3D.IMPORTEDGEOMET.88089826
000031  027078  prouser 2      BBK.SW1L1.GEOMSHOTS
```

Example

The following query reports all 8mm tapes:

```
QRY[ QRYFLD:"MPOOLNAME" QRYEQ:"8mm_dp"]
```

Example

The following query would report all multi-volume datasets with 'GEOM' in the dataset name. The first query asks VOLSEQ>0 (multi-volume); the next selects the last volume in each dataset (NXTVOL=LAST); the final step does a wildcard string match equivalent to '*GEOM*'. Intermediate files R0 and R1 are used to pass results from step to step:

```
QRY[ QRYFLD:"VOLSEQ" QRYGT:"0" QRYSAVE:"R1"]
QRY[ QRYFILE:"R1" QRYFLD:"NXTVOL" QRYEQ:"LAST"
   QRYSAVE:"R0"]
QRY[ QRYFILE:"R0" QRYFLD:"DSN" QRYWCSUB:"GEOM"]
```

VEDIT[...]

Edit set; describes which fields to edit and their replacement values based on volume serial numbers.

Syntax

```
VEDIT[ EDVRNG: "fromvsn-tovsn,singlevsn,fromvsn-tovsn,..."
EDFLD: "field_name" "new value"
EDFLD: "..." "..."
 . . .
ENDVRNG
EDVRNG: "fromvsn-tovsn"
EDFLD: "field_name" "new value"
]
```

Options

EDVRNG Volume serial number range to apply edits to. This may be a single VSN, continuous VSN range, multiple single volumes, multiple continuous ranges, or any combination thereof.

EDFLD Defines which field to edit and its new value. Two string values are expected. The first value is the field name. The second value is the replacement value or new value to be stored in the catalog. As many EDFLD commands may be strung together to complete the overall desired edit.

ENDVRNG Terminates a volume range edit group if more than one EDVRNG/EDFLD is to be executed. You can apply many different edits in one VEDIT session. A big advantage for performing multiple edits in one VEDIT session is that the catalog is opened only once for multiple edits. We could achieve the same result by stringing multiple VEDIT[]VEDIT[] groups together, but there is added overhead of opening and closing the catalog between each operation.

Example

The folowing would reassign tapes from one media pool to another and change group ownership, editing 31 volumes in the default catalog, switch catalogs and edit 1000 more volumes.

```
VEDIT[ EDVRNG: "VOL001-VOL015,VOL092,V1S0N1-V1S1N5"
EDFLD: "MPOOLNAM" "3490e" EDFLD: "GRPOWN" "exploration" ]
USECAT: "prod_cat"
VEDIT[ EDVRNG: "RIO3001-RIO4000"
EDFLD: "MPOOLNAM" "3490e" EDFLD: "GRPOWN" "production" ]
```

Since all catalog commands are atomic, the first VEDIT opens the default tape catalog, applies its edits, then closes. A different catalog, prod_cat, is selected and edits are applied in a similar manner.

VDISP[...]

Displays field values for VSN ranges.

Syntax

```
VDISP[ DISPVRNG: "fromvsn-tovsn,singlevsn,fromvsn-tovsn,..."
DISDELIM: " "
DISPFLD: "field_name" DISPFLD: "field_name"
. . .
ENDVRNG
DISPVRNG: "fromvsn-tovsn"
DISPFLD: "field_name" DISPFLD: "field_name"
]
```

Options

DISPVRNG Volume serial number range for which to display values. This may be a single VSN, continuous VSN range, multiple single volumes, multiple continuous ranges, or any combination thereof.

DISDELIM Character to delimit each displayed field on output.

DISPFLD Field name to display within a VSN record. The order in which these values are strung together also determines the order in which they will be displayed.

ENDVRNG Terminates a volume range display group if more than one value display is to be executed. You can display many different volume ranges as well as different values in one VDISP session. A big advantage for performing multiple displays in one VDISP session is that the catalog is opened only once for multiple calls. We could achieve the same result by stringing multiple VDISP[]VDISP[] groups together, but there is added overhead of opening and closing the catalog between each operation.

Example

```
topcat ’CATINFO[ INFDESC ]’
topcat ’CMDFILE: "vdisp_example"’
```

```
- - - top of file vdisp_example - - -
VDISP[ DISPVRNG: "V1S0N1-V1S1N5"
            DISPFLD: "DSN "
            DISPFLD: "BORN"
# DISPFLD: "NACCESS"
# DISPFLD: "VOLSTAT"
            DISPFLD: "STATDATE"
            DISPFLD: "PRSTAT"
            DISPFLD: "LABEL"
            DISPFLD: "MPOOLNAM"
            ENDVRNG
            DISPVRNG: "VOL001-VOL015,VOL092,V1S0N1-V1S1N5"
            DISPFLD: "BORN"
            DISPFLD: "NACCESS"
            DISPFLD: "VOLSTAT"
            DISPFLD: "STATDATE"
            DISPFLD: "PRSTAT"
            DISPFLD: "LABEL"
# DISPFLD: "MPOOLNAM"
# DISPFLD: "LOCATION"
# DISPFLD: "GRPOWN"
]
- - - end of file vdisp_example - - -
```

QRYEDIT[...]

Edit values based upon query results (edit query list).

Syntax

```
QRYEDIT[ MEDIANAM: " 3490e" GRPOWN: " exploration" ]
```

Usage

All fields get same value.

INHERIT[...]

Copies information from one dataset to another.

Syntax

```
INHERIT[ PARENT: "dsn" CHILD: "dsn" "field1" "field2" ... ]
```

PASSWORD[...]

Access for password-protected datasets (optional). Currently non-functional

Commands used by Tape I/O routines

OPNDSN[...]

Opens a dataset name.

Note: While OPNDSN could be called from the commandline or a user script, it is most commonly used by Tape I/O operations from within a ProMAX job.

Syntax

OPNDSN[OPNMODE: open_mode ODSN: "dataset name" DSVER: "n" LABL: "SL | AL | NL"

MPOOL: " media pool" DPOOL: "device pool" RETD: "retention_period"

EXPD: "expiration_date" RESVN: "nvolumes"

]

OPNMODE options

DSCREATE Create a new dataset, obtaining volumes from media scratch list.

DSOVRWRT Replace existing dataset with a new one, reusing existing volumes.

DSOLD Open an existing dataset.

DSAPND Append to an existing dataset. Primarily used for archives.

ODSN Dataset name to open.

DSVER Version of dataset to open (default is one).

LABL Volume label type. Mandatory for OPNMODE:DSCREATE|DSOVRWRT

NL = No label

SL = IBM standard label (EBCDIC)

AL = ANSI label (ASCII)

MPOOL Defines which media pool to retrieve scratch volumes from. Mandatory for OPNMODE:DSCREATE

DPOOL If MPOOL is not known, a device pool name may be used to cross reference the apropriate media pool. Mandatory for OPNMODE:DSCREATE if MPOOL is not defined.

RETD Retention period. Optional for OPNMODE:DSCREATE|DSOVRWRT Expiration date is calculated by: current_julian_date + retention period.

EXPD Expiration date. The volume is to be marked scratch at this Julian date (default 2099 or 099000). Optional for OPNMODE:DSCREATE|DSOVRWRT

RESVN Reserve nvolumes for output (default is one). Optional for OPNMODE:DSCREATE

Usage

Returns number of volumes, label type, media pool name, device pool name. Next, OPNDSN returns the volume list (all VSNs on one line). Memory requirements for returned VSN list: (NVOLS+1)*(7); the VSN list is delimited by blanks " ". The dataset (type) information is simply echoed back for DSCREATE, but is neccessary for opening existing datasets.

For all OPNMODE open modes, the user's group id must be in the GRPACCESS stanza list for a given media pool (if it has been defined). Otherwise, access to media will be denied.

For DSCREATE open mode, the dataset name must be unique in the catalog. If it is not, a new version number must be specified; a new version number will not be assigned automatically.

For DSOVRWRT open mode, you may write over a dataset if VOLSTAT is not RDONLY. If media name is not specified, the open will re-occupy existing volumes. The dataset name may be recreated by specifying a new media type; this operation will cause a delete of the old dataset (volumes returned to scratch). If a RESVN other than 1 is desired, it must be specified. If the dataset does not exist, it will be created and follow the same rules as DSCREATE. If RESVN is specified, the number of RESERVED volumes will be either RESVN or number in dataset (which ever is greater).

For both DSCREATE and DSOVRWRT open modes, VOLSTAT is set to "RESERVED" at the conclusion of a successful open for all volumes currently in the group (RESVN or number in dataset).

Return Example

```
CATSTAT: GOOD
NVOL: 5 LABL: SL MEDIA: robo_dlt DEVICE: robot1
VOLS: VOL211 VOL212 VOL001 VOL002 VOL003
```

CLOSEDSN[...]

Close dataset name.

Note: Like OPNDSN, CLOSEDSN could be called from the commandline or a user script, but is most commonly used by ProMAX Tape I/O operations from within an executing job.

Syntax

```
CLOSEDSN[ CLSDSN: " dataset name" CLSVER: "n"
DSNVAL: "field name1" "new value1"
DSNVAL: "field name2" "new value2"
 . . .
]
```

Options

CLSDSN Name of dataset to close (name used to open the dataset).

CLSVER Version of dataset to close.

DSNVAL Describes the a new set of values which will be applied to the entire dataset (such as completion status). It defines a catalog field to edit and its new value. Two string values are expected; the first value is the VSN’s catalog field name; the second value is the new or replacement value to be stored in the catalog. As many DSNVAL commands may be strung together to complete the overall desired edit.

Usage

During a dataset creation job, CLOSEDSN will return all volumes marked as "RESERVED" back to "SCRATCH". If a job terminates abnormally and CLOSEDSN is called before any volumes were written to, the dataset is not created and all reserved volumes returned to scratch. The return status will reflect this condition with "NVOL: 0".

If the process is killed and this function is called in clean-up, all volumes with "CURSTAT" of "OPEN" will be returned to "READY".

Return Example

```
CATSTAT: GOOD
CLOSED: RIO.TEST.TAPE.SEGY0044  NVOL: 5
```

DELDSN[DDSN: "dataset name" DVER: "n"]

Deletes dataset name.

Note: DELDSN is called both by executing jobs and by the ProMAX User Interface, upon user deletion from the ProMAX datasets list of a tape dataset associated with the catalog.

Options

DDSN Dataset name to delete from the catalog.

DVER Version of dataset to delete (default = 1).

Usage

A dataset with VOLSTAT equal to "RDONLY" cannot be deleted with this function. The user must change VOLSTAT from "RDONLY" to "ACTIVE".

Return Example

```
CATSTAT: GOOD
DELETED: RIO.TEST.TAPE.SEGY0044 NVOL: 5
```

OPNVSN[...]

Opens Volume Serial Number.

Note: While OPNVSN could be called from the commandline or a user script, it is most commonly used by ProMAX Tape I/O operations from within an executing job.

Syntax

```
OPNVSN[ OVSN: "PB070361" OMOD: open_mode OPID:
    "process_id" OHOST: " hostname" ]
```

Options

OVSN Volume serial number to open. CURSTAT is set to "OPEN".

OMOD Open mode (same as "OPNMODE" in OPNDSN).

OPID Process ID of caller.

OHOST Machine hostname of caller.

Return Example

normal completion:

```
CATSTAT: GOOD
VSN: VOL000 now owned by: casper
```

volume is in use - we may wait for the volume to become available:

```
CATSTAT: ERROR
VSN: VOL000 is locked by user: sneezer
```

volume is write protected - open mode is not DSOLD:

```
CATSTAT: ERROR
VSN: VOL000 has status of RDONLY
```

CLSVSN[...]

Closes Volume Serial Number.

Note: Like OPNVSN, CLSVSN could be called from the commandline or a user script, but is most commonly used by ProMAX Tape I/O operations from within an executing job.

Syntax

```
CLSVSN[ CVSN: "PB070361" CMOD: open_mode
VSNVAL: "field name1" "new value1"
VSNVAL: "field name2" "new value2"
. . .
]
```

Options

CVSN Volume serial number to close or release. CURSTAT is set from "OPEN" to "READY".

CMOD Open mode (same as "OPNMODE" in OPNDSN). Open mode is important for handling of the end of the vsn list. If the CMOD open mode is "DSOLD" and end of list is encountered, a value of "LAST" is returned. A new scratch volume is assigned if the end of current list encountered. Contents such as DSN, label type, and expiration date are copied into the new volume and VSN chain updated.

VSNVAL Allows volume-specific information to added to the catalog at the completion of output (such as ensemble range and number of traces). Two string values are expected; the first value is the VSN's catalog field name; the second value is the new or replacement value to be stored in the catalog. As many VSNVAL commands may be strung together to complete the overall desired edit.

Return Example

normal completion"

```
CATSTAT: GOOD
NXTVOL: VOL001 CLOSED: VOL000
```

normal completion - end of dataset"

```
CATSTAT: GOOD
NXTVOL: LAST CLOSED: VOL001
```

scratch pool empty - detailed message follows:

```
CATSTAT: ERROR
NXTVOL: NOSCRATCH
```

REJVSN[RVSN: "vsn" RSTAT: "BAD "]

Reject Volume Serial Number; volume was unusable by application. Applies for output only.

REJVSN is called by ProMAX Tape I/O operations when problems occur with a specific VSN.

Options

RVSN Volume serial number rejected.

RSTAT Rejection status; value is stored in VOLSTAT header value.

Usage

This routine is used in the event that a volume is unusable by the application. The application may mark the volume as "BAD" or "NOTFOUND" in the event of a mount and open error. The current VSN is removed from the chain and a new scratch volume is assigned.

Return Example

normal completion:

```
CATSTAT: GOOD
NXTVOL: VOL001 REJECTED: VOL000
```

scratch pool empty - detailed message follows:

```
CATSTAT: ERROR
NXTVOL: NOSCRATCH
```

Administrator Commands

POPULATE[POPMPOOL: "media_pool" POPVRNG: "fromvsn-tovsn"]

Populate (create/append) the Catalog with new VSNs and initialize volume records with defaults.

Options

POPMPOOL Media pool name (MPOOLNAM) to which the new VSNs will belong.

POPVRNG Volume serial number range with which to populate the catalog. This may be a single VSN, continuous VSN range, multiple single volumes, multiple continuous ranges, or any combination thereof. If the VSNs already exist in the current Catalog, the POPULATE operation will fail; if the VSNs do not exist, they will be added/appended to the Catalog.

Usage

VSN strings should be ASCII (but not limited to) as defined by ISO 1001.

To interpolate POPVRNG ranges, POPULATE will look for numeric characters and increment from zero to nine. Alphas are not incremented within the interpolation.

The POPVRNG VSN range must not coincide with any existing VSNs currently in the Catalog or the POPULATE operation will abort with a message like: "VSN ranges overlap: [mmmmmm-nnnnnn] - [mmmmmm-nnnnnn]"

REMOVE[REMVRNG: "fromvsn-tovsn"]

Physically remove volume records from catalog.

** Currently non-functional **

IMPORT[IMPFILE:" filename" IMPMAP: "mapfile"]

Import ASCII catalog values.

Options

IMPFILE File containing ASCII catalog values to be imported. IMPFILE is assumed to be a field-delimited text file.

IMPMAP File which describes the import mapping.

Contents of IMPMAP:

MAPDESC Comment describing what type of import map this represents.

MAPDELIM Defines the character which delimits each field.

MAPFLDS Defines which fields are to be imported from the IMPFILE. The order in which these fields appear represents the mapping.

One of the MAPFLDS must be "VSN"; the import cannot function without this piece of information.

Example

This example will map the first field (e.g. 710001) as the VSN, the second field (SL) as LABEL, the third field (ProMAX) as FMTNAME, the fourth field (1/2) as MPOOLNAM, etc.

```
 - - - top of IMPFILE - - -
710001|SL|ProMAX|1/2|ACTIVE| ... other values
710002|SL|ProMAX|1/2|ACTIVE| ... other values
710003|SL|ProMAX|1/2|ACTIVE| ... other values
. . .
- - - end of IMPFILE - - -
- - - top of IMPMAP - - -
```

```
MAPDESC:"Catalog imported from ProMAX tcat"
MAPDELIM:"|"
MAPFLDS["VSN" "LABEL" "FMTNAME" "MPOOLNAM" "VOLSTAT"
    ... ]
- - - end of IMPMAP - - -
```

Usage

Using the token or MAPDELIM field delimiter, the input record is IMPMAP file, these values are then mapped to the corresponding catalog field position. Note that the values imported will simply replace the respective values currently in the volume record fields. All other fields in the volume record are left unchanged. So if other information was already stored in other catalog fields for these volume records, they will remain and all imported values will be overlaid onto their corresponding fields, overwriting what was previously there.

Each delimited field must be represented by a corresponding MAPFLD value. If the MAPFLDS list is underdefined (i.e. there are more IMPFILE entries than MAPFLDS entries), unrepresented delimited IMPFILE fields are simply ignored (input truncated). So if the number of MAPFLDS is 12 and the number of delimited fields is 20; the first 12 delimited fields are imported and the other 8 are left off. It is recommended that information which is not to be imported be edited out of the input file (IMPFILE).

If the imported catalog contains information which does not map to a predefined field, you may clump all of these values into one larger field by removing the delimiter(s) and map this value into the user comment field "COMMENT" (the comment field can store up to 160 bytes).

"VSN" must be one of the MAPFLDS which is imported from the IMPFILE.

Note: IMPORT overwrites the Catalog fields of all VSNs without warning. This is not an issue if you are IMPORTing the contents of an external Catalog (like the old ProMAX *tcat* Catalog), but it also means that you can use IMPORT in a way similar to TopCat's VEDIT option.

Example

An example usage of the IMPORT option is described in the Configure secondary or imported tcat catalogs section of the

Installation & Administration Guide where it is used to import a previous ProMAX tcat tape catalog.

MERGE[OUTCAT: "newcatname" CAT1: " catfile1" CAT2: " catfile2"]

Merge two catalog files.

Options

OUTCAT Output file name or name of newcatalog file (can use name of one of the input files).

CAT1/CAT2 Catalog files to be merged.

Code Example

```
int get_cat_info(void)
{

char cat_cmd[80];
FILE *cathan;
int status;
char bfr[256];
char *b;
sprintf(cat_cmd,"topcat ’CATINFO[INFTYPES INFLIST INFLABEL
    INFDESC ]’");
printf("Issue tape catalog command: %s\n",cat_cmd);

cathan = popen(cat_cmd,"r");
b = bfr;

while(b !=NULL) {
b = fgets(bfr,256,cathan);
if(b!=NULL) printf("%s",bfr);
}
status = pclose(cathan);
return(0);

}
```

PMX Guide

This guide explains configuring and operating PMX. This component of TTS allows automated device management, including the use of device pooling, and the operator console. You should validate the setup by running a tape job prior to starting production work.

The PMX device management allows you to access tape data by identifying logical device names within device pools, rather than selecting specific physical devices. Reservation and allocation of devices is handled automatically, allowing you to focus on processing data, rather than managing devices and datasets.

Note: The use of PMX device management requires a key piece of information that is ***not*** provided in this documentation. You must contact ProMAX Support in Denver before using PMX. We leave this information out so we can review your intended configuration and ensure effective use of the system.

In This Chapter

Installation and Setup

Preliminary considerations

It is helpful to review where all the TTS-related files reside prior to configuring PMX. See ***TTS Files*** for the list of file names, locations and their primary functions.

We assume that the devices controlled by PMX have been configured for normal ProMAX use, including appropriate settings for variable block size and long block driver installation if needed. PROMAX_HOME is set and PATH includes $PROMAX_HOME/sys/bin, $PROMAX_HOME/sys/exe and $PROMAX_HOME/port/bin.

General device configuration

All local and remote devices under PMX control should be configured for access as described in ***Basic TTS Configuration***. You will be making changes to the ***device.tts*** file from this basic configuration.

When using PMX, all users should share a common ***device.tts*** file. If different ***device.tts*** files are used, the resource management functionality can be compromised and unexpected behavior could occur.

Device pool definitions

A device pool is a group of devices seen as ***equivalent*** under PMX management. You select a logical device name in a pool without caring which physical device is allocated because all devices are equivalent.

Requirements

All devices within a given device pool:

- use the same media type
- are located on the same machine
- are of the same type (simple or stacker or robotic, not a mixture)
- have the same behavior characteristics, such as stackers should all advance to the next tape in the same manner

- have a maximum block size set no larger than the smallest value of any device in the pool
- do not include a given physical device in more than one device pool

Recommendations

We make the following recommendations:

- it is acceptable to define a single device within a pool
- the number of logical devices defined for a pool should be equal to or less than the number of physical devices in the pool
- most ProClient parameter options should be set at the pool level in order to ensure consistent behavior among devices

If stacker devices are used for output, create different pools for output and for input. This facilitates use of CONFIG="SMNT" mode, such that an output stacker can be loaded with scratch tapes that will be used sequentially without generating tape mount messages.

Configuring the PMX Resource Manager

An example PMX Resource Database directory setup is included in the $PROMAX_HOME/etc/rdb ProMAX installation directory. The ***config_file*** is used for defining your Resource Manager database. Do not confuse this file with the $PROMAX_HOME/etc/config_file.

Content and structure of the config_file

The $PROMAX_HOME/etc/rdb/***config_file*** is the source of information for the PMX Resource Manager to comprehend the devices, device pool names, and device pool associations. The ***config_file*** contains one or more device pool definitions consisting of the following stanzas:

- **device** stanzas; one stanza stating the hostname and Unix name for each physical device in the pool
- **pool** stanza; followed immediately by the hostname and Unix name for each physical device in the pool

A blank line after the list of devices in the pool stanza serves as a flag for the end of the pool definition.

ProClient options can be set in both the device and pool stanzas. Options set in a device stanza only affect that one device; options set in a pool stanza affect all devices within the pool. Additional options, associated with logical device names, are set in the ***device.tts*** file.

ProClient options are cumulative, with the last specification of a property taking precedence. The hierarchy is that a value is taken from:

1. the ***device.tts*** file
2. the PMX database device stanza
3. the PMX pool stanza

For example, if MAXBLK is specified in the pool stanza and in ***device.tts***, the value in ***device.tts*** is used if that physical device is allocated. Therefore, we recommend that all devices in a pool share common settings at the pool level.

Syntax

Each property-value pair is delimited at the end by a semicolon.

Each device stanza and each pool stanza must be one continuous line.

The PMX Resource Manager currently supports up to 12 property options defined per device. Be aware of the possibility of accumulating redundant entries in device and pool stanzas and exceeding this limit.

Example config_file

The following is an example ***config_file***. The pound sign (#) indicates a "comment".

```
# Comments begin with a '#' or '*'

# Preceding spaces and tabs are ignored.

**********************************

# 3490 Fujitsu stacker device pool

device: stargazer:/dev/rmt/tps43d1v CONFIG = SMNT;

device: stargazer:/dev/rmt/tps43d2v CONFIG = SMNT;

pool: 3490_dp DEVTYPE = STACKER; DEVCAP = 10;

stargazer:/dev/rmt/tps43d1v

stargazer:dev/rmt/tps43d2v

*************************

# 8mm Exabyte device pool

device: stargazer:/dev/rmt/tps1d5nsv DEVTYPE = STACKER; DEVCAP
= 10; CONFIG = SMNT;

device: stargazer:/dev/rmt/tps0d4nsv DEVTYPE = SIMPLE; DEVCAP =
1;

pool: 8mm_dp

stargazer:/dev/rmt/tps1d5nsv

stargazer:/dev/rmt/tps0d4nsv

*****************************************

# DLT StorageTek robotic silo device pool

device: stargazer:/dev/rmt/tps6d1nsv DEVINFO = 0,0,2,0;

device: stargazer:/dev/rmt/tps6d2nsv DEVINFO = 0,0,2,1;

device: stargazer:/dev/rmt/tps6d3nsv DEVINFO = 0,0,2,2;

device: stargazer:/dev/rmt/tps6d4nsv DEVINFO = 0,0,2,3;

pool: dlt_dp  DEVTYPE = STKROBOT; MAXBLK = 3072000; NRET =  2;
RWAIT = 5;

stargazer:/dev/rmt/tps6d1nsv

stargazer:/dev/rmt/tps6d2nsv

stargazer:/dev/rmt/tps6d3nsv

stargazer:/dev/rmt/tps6d4nsv
```

Naming Conventions

The example above defines three device pools with the following names:

- 3490_dp
- 8mm_dp
- dlt_dp

The **dp** in the names suggests **d**evice **p**ool, which is distinct from media pools defined for the **TopCat** catalog. It is helpful to adopt a naming convention to easily differentiate between device pool and media pool names but still allows you to recognize meaningful pairings of media and device pools.

The PMX Resource Manager currently supports up to 8 device pools, with up to 24 devices in each pool.

Note: The TopCat CATPOOL definitions include the association between device pool names and media pool names. In order to use PMX device management with TopCat volumes, you must make sure you have consistent naming conventions for DEVPOOL entries in the ***catadmin*** file of TopCat and the pool names you declare in the PMX config_file. The entry for DEVPOOL must be identical to the pool name of devices that will use those corresponding volumes from the catalog.

Define devices and device pools

The parameters: DEVHOST, DEVTYPE, DEVCAP, CONFIG, DEVINFO MAXBLK, NRET, RWAIT are all ProClient options. These are set to alter the runtime behavior of the tape system. A full listing of ProClient options is found in the ***ProClient Command Line Notes and Examples***. We use some of the more common options in the above example. At a minimum, you should define DEVTYPE, DEVCAP, and DEVHOST for devices and/or pools.

In the above example:

- device pool ***3490_dp*** consists of two STACKER drives with a DEVCAP (device capacity) of 10.
- device pool ***8mm_dp*** consists of two drives: one 8mm drive with a 10-tape STACKER and a single SIMPLE drive.

- device pool *dlt_dp* consists of four drives: They are in a STKROBOT StorageTek robotic silo; the DEVINFO parameter passes the four StorageTek ACS, LSM, Panel, driveunit numbers to the tape system. The MAXBLK parameter is set to write data in 3072000-byte block sizes. We strongly urge you to use the **TapeDiag** utility to identify optimal block size for each device. See ***Using TapeDiag Tape Diagnostics*** for details. The NRET/RWAIT parameters tell the tape system to retry 2 more times if there is an unsuccessful I/O operation, waiting 5 seconds between each retry.

Initialize the PMX Resource Manager

After you have defined your devices and device pools in the *config_file*, initialize the PMX Resource Manager:

```
> cd $PROMAX_HOME/etc/rdb
> $PROMAX_HOME/etc/rdb/pmxres_init
```

The script *pmxres_init* assumes the *config_file* is in the current directory. Alternatively, you can issue the following command in your PMX directory to give the explicit pathname to the *config_file* and the PMX logfile:

```
> $PROMAX_HOME/sys/bin/pmxInit -c config_file -m logfile.pmx -u -z
```

This creates two files in $PROMAX_HOME/etc/rdb called *static* and *dynamic*. The **-u** option initializes all device pools in the *up* state and the **-z** option overwrites any existing static/dynamic database files.

- **static** stores all device/host/pool definitions that were set in the config_file
- **dynamic** tracks all active and queued requests as they are acted upon by **pmxRsv** and **pmxRls**. (**queued requests** refers to active jobs that have not allocated the necessary resources; this is entirely unrelated to any batch job queues.)

We include a call to ***$PROMAX_HOME/sys/bin/pmxDev*** in ***pmxres_init.*** This call displays a message similar to this:

```
ProMAX Resource Management ... Version : 1.0 May 1998

 =====================================================

Default Log File = local

Resource Requests

Active Requests :

|====== Request ======|==== Pool ====|======== Device
========| status ...

Queued Requests :

|====== Request ======|==== Pool ====|======== Device
========| status ...

Static Pools and Devices

|====== Request ======|==== Pool ====|======== Device
========| status ...

3490_dp                    free ..

3490_dp     /dev/rmt/tps43d1v    free ..

3490_dp     /dev/rmt/tps43d2v    free ..

...


=====================================================

ProMAX Resource Management ... Version : 1.0 May 1998
```

This display shows all device pools and the devices within each pool. If some devices or pools are missing, or if the properties of some of the devices are missing or wrong, go back and check your ***config_file*** entries for spelling, and confirm that each **property=value** setting ends with a semicolon.

Using the PMX Resource Manager describes all command line options to the five PMX commands. In normal conditions, these commands are executed automatically from within other programs, and not from the command line.

Set PROMAX_ETC_RDB_HOME

The home directory for the PMX Resource Manager Database (rdb) is explicitly defined using environment variable PROMAX_ETC_RDB_HOME; the default is

$PROMAX_HOME/etc/rdb. For maximum flexibility, we recommend that you set environment variable PROMAX_ETC_RDB_HOME in your ProMAX login environment.

Configuring $PROMAX_HOME/etc/device.tts

We assume that you have created a ***device.tts*** file that defines all local and remote devices located under PMX control. We strongly recommend that all users relying on PMX device management share a common ***device.tts*** file.

You should have already defined the pool names and the physical devices contained in each pool. Update the ***device.tts*** file to define *logical device names*. These logical device names are displayed in the tape processes. The system handles all device allocations. Therefore, devices within a pool should have the same characteristics.

Example

This example show a device.tts configuration.

```
# devices managed by PMX

tape_robot_1 DPOOL=dlt_dp CLIENT="sgimips4/exe/ProClient"
  REMHOME="/release/1998.6/ProMAX"

tape_robot_2 DPOOL=dlt_dp

tape_robot_3 DPOOL=dlt_dp

tape_3490_1  DPOOL=3490E_dp

tape_3490_2  DPOOL=3490E_dp

tape_magstar DPOOL=3590_dp  REMHOME="/promax/1998.6"

#  devices NOT managed by PMX

tape_8mm_1  DEVICE=stargazer:/dev/rmt/tps7d1nsv  DPOOL=8mm_dp
  DEVTYPE="STACKER"  CLIENT="ProClient"
  REMHOME="/release/1998.6/ProMAX"

tape_8mm_2  DEVICE=spitfire:/dev/rmt/0b  DPOOL=8mm_dp
  DEVTYPE="SIMPLE"  CLIENT="ProClient" REMHOME="/data2/TTS"

tape_8mm_3  DEVICE=willy:/dev/rmt/1b  DPOOL=8mm_dp
  DEVTYPE="SIMPLE"  CLIENT="ProClient"
  REMHOME="/sdt1/release/1998.6/ProMAX" CONFIG="SMNT+FMNT"
```

Logical Device Names

It is important to keep in mind that you have defined the physical devices in the ***rdb/config_file***. The *devices* under PMX control described in ***device.tts*** are *logical device names*.

Each **logical device name** in the ***device.tts*** file must be unique.

In the device.tts configuration example, there are two groups of logical devices.

- Group 1: Those devices under PMX control, including three logical robotic devices in one device pool, two 3490 drives in another pool, and a single 3590 device in a pool. Notice that there is no DEVICE entry in ***device.tts*** for devices that are under PMX control. The DEVICE entry identifies physical device names defined in the PMX config_file.

  The DPOOL entry must be the same for all devices in the same pool. This is the key to associating the logical device to the PMX device pool.

- Group 2: Those devices that are not under PMX control. The use of these devices must be coordinated among users if the devices are to be shared. Alternatively, this could be a user-owned ***device.tts*** file and only this user has access to these last three devices.

Selecting logical device names in flow menus

Even though a device pool entry in ***device.tts*** does not specify a specific device on a specific host, it still represents one specific tape device for the duration of that ProMAX job. The PMX manager handles the association of logical to physical devices.

Example

The following situation is an example for selecting logical device names:

- There are four physical devices in a device pool.
- Your job requires a device for input and two separate output tape datasets.
- You need three physical devices.

- You do not care which of the four devices you get

Then you would specify in your flow:

Tape Data Input: dlt_1

Tape Data Output: dlt_2

Tape Data Output: dlt_3

where **dlt_1**, **dlt_2** and **dlt_3** are logical device names from *device.tts*.

You must remember that each dataset requires a unique logical device. However, when **Tape Data Input** is followed by one or more **Tape Data Inserts**, the same logical device name can be used for all input datasets. This exception occurs because the earliest dataset must be read completely before the job can begin reading the next tape dataset. Therefore, the same logical device can be re-used in the job. A logical device is always associated with the same physical device for the duration of a job. Therefore, all input dataset tapes can be loaded in a stacker.

Device Characteristics

You can customize behavior for logical devices by specifying ProClient options in the *device.tts* file. For example, you would specify a particular logical device if you want one to behave in a particular manner that is different than *normal*.

Configuring TTS environment variables

ProMAX Environment Variables used by TTS

PROMAX_ETC_RDB_HOME - Path to PMX Resource directory. Defaults to $PROMAX_HOME/etc/rdb.

PROMAX_TOPCAT_HOME - Path to TopCat Catalog directory. Defaults to $PROMAX_HOME/TopCat.

Use of the TopCat catalog is not required when using PMX device management. However, if you do enough tape processing to benefit from device management, you probably will benefit by using the tape catalog as well.

Note: You must contact ProMAX support in Denver to get additional information to make PMX device management operational. We require this so we can review your configuration and ensure that it will work effectively.

Validating the TTS setup

After installing TTS, you should validate the setup by using the following steps.

1. Start the OpCon Operator Console

Start the OpCon Operator Console by executing the script:

```
> $PROMAX_HOME/port/bin/opcon &
```

Make sure your DISPLAY environment variable and any relevant xhost/xauth settings are correctly set.

You can test the message handling ability of the OpCon, outside of ProMAX, by sending messages to the Operator Console from the command line.

To send a tape mount request to the **Mount Request** pane of the OpCon, enter:

```
> $PROMAX_HOME/port/bin/TAPE_REQUEST op_host mount
   vsn123 /dev/rmt0 host_name jeffw 1234 rtype NL 5678
```

op_host is the hostname where OpCon is running. You can **cancel** this mount message.

To send a tape notification message to the Operator Notification pane of the OpCon, enter:

```
> $PROMAX_HOME/port/bin/TAPE_NOTIFICATION host_name test
   "hello world"
```

Command line executions of both the TAPE_REQUEST and TAPE_NOTIFICATION shell scripts should cause the OpCon to display a mount request or notification message

Menu Bar

The bottom panel of the operator console is composed of three tabs: **Device Status**, **Active Devices**, **Queued Resources**, and four buttons: **Inquire**, **Toggle Pool**, **Toggle Device**, **Force Device**.

Inquire displays all the devices and pools defined in the *$PROMAX_HOME/etc/rdb/config_file* for the PMX Resource Manager.

Click anywhere on the line of a given a device to select it, then click **Toggle Device** or **Toggle Pool** disable/enable that device or all devices belonging to that device pool.

2. Run a few ProMAX tape jobs

Run a few output jobs using **Synthetic Trace Generation** or **Disk Data Input**, and then read them back in. Also try reading in a few field tapes and long-block field data, if they are used at your site.

Using the PMX Resource Manager

PMX Resource Management reads the ProMAX configuration file and builds a database containing the devices, their pools and their status. This section describes the commands for requesting either a specific device or a device from a pool, returning devices to the available state, reporting the current state of the resource management system, and providing for resource database maintenance.

PMX Resource Manager Commands

pmxInit -c config_file -m /path/logfile -d database_file -z

Initialize the ProMAX resource database (rdb).

Description

pmxInit initializes or updates the ProMAX resource database using an input configuration file.

pmxInit currently opens up the write permissions on the dynamic/static rdb files which are created. Due to the use of internal file locking, accesses by all pmx___ routines, including pmxDev, require write permission to these files. To prevent accidental deletion of these files by users, the system administrator may consider locking down this PROMAX_ETC_RDB_HOME directory to be writeable only by a central user account (e.g. user 'oper') and then making all four pmx___ programs setuid binaries.

Options

-c config_file The resource configuration file to read. Defaults to PROMAX_ETC_CONFIG_FILE_HOME

-d database_file The database to update or create. Defaults to variablePROMAX_ETC_RDB_HOME, or if PROMAX_ETC_RDB_HOME is not set, the default resource database directory is "&PROMAX_HOME/etc/rdb". All other PMX routines should also honor this variable. In addition, the two static/dynamic component files may be set individually using PROMAX_ETC_RDB_STATIC_HOME or PROMAX_ETC_RDB_DYNAMIC_HOME. This latter approach is not recommended since the static and dynamic components of the rdb database should be kept in one location. Likewise, the PROMAX_ETC_RDB_HOME variable should be the same for all users using the same devices.

-m /path/logfile Optional pathname to logfile to receive warnings and errors. If not supplied or if supplied as syslog, all messages will be sent to the system log. Use of this option will output messages to both stderr (job.output) as well as to the specified logfile.

The pmx___ routines use a log facility of LOCAL0. So to capture any messages in syslog the syslog.conf file might contain, for example:

```
local0.debug    /var/adm/LOCAL0
```

which results in messages of all priorities written into /var/adm/LOCAL0

-u Initialize all device pools in the "up" state. The default is to initialize all pools in the "disabled" state, which can then be enabled by using the **Toggle Pool** button of OpCon’s **Device Manager** window.

-z Flag to replace an existing database, providing the database is not in use. If the database is actively managing resources, an error will occur from using this flag.

Configuration File Format

The pmxInit configuration file is expected to have stanzas of this format:

```
device: stargazer/dev/rmt/tps6d1nsv PROPERTY1 = VALUE1; ...
    PROPERTYn = VALUEn;
device: stargazer/dev/rmt0 PROPERTY1 = VALUE1; ... PROPERTYn
    = VALUEn;
pool: pool_name1 PROPERTY1 = VALUE1; ... PROPERTYn =
    VALUEn;
stargazer:/dev/rmt/tps6d1nsv
stargazer:/dev/rmt/...
pool: pool_name2 PROPERTY1 = VALUE1; ... PROPERTYn =
    VALUEn;
/dev/rmt0
/dev/...
```

The properties defined in the "device:" and "pool:" stanzas are ProClient command arguments, all of which are documented in the ProClient Command Guide. At a minimum, you should

Other Docs Known Problems

define DEVTYPE, DEVCAP, and MACHINE for the devices and pools. The order of the property listings is not important, but every device name referenced in a pool must exist. Every device does not have to be in a pool. Duplicate pool names create an error and duplicate device names will cause a warning. The property-value assignment pairs should be delimited by semicolons (;). Properties are cumulative with the last specification of a property taking effect.

Pool-specific attributes should be defined on the "pool:" stanza, as opposed to being defined for every "device:" stanza in that pool.

The following is an example resource configuration file:

```
#*****************************************************************

# 3490 stackers

#*****************************************************************

device: stargazer:/dev/rmt/tps6d1nsv DEVTYPE = STACKER; DEVCAP
= 10; CONFIG = SMNT; MAXBLK = 262144;

device: stargazer:/dev/rmt/tps6d2nsv DEVTYPE = STACKER; DEVCAP
= 10; CONFIG = SMNT; MAXBLK = 262144;

pool: 3490_dp

stargazer:/dev/rmt/tps6d1nsv

stargazer:/dev/rmt/tps6d2nsv

#*****************************************************************

# STK robotics

#*****************************************************************

device: stargazer:/dev/rmt0 DEVTYPE = STKROBOT; DEVINFO =
0,0,2,0;

device: stargazer:/dev/rmt1 DEVTYPE = STKROBOT; DEVINFO =
0,0,2,1;

pool: dlt_dp  DEVTYPE = STKROBOT;

stargazer:/dev/rmt0

stargazer:/dev/rmt1

#****************************************************************

# 8mm stackers

#****************************************************************

device: stargazer:/dev/rmt/tps1d5nsv

device: stargazer:/dev/rmt/tps0d4nsv

pool: 8mm_dp DEVTYPE = STACKER; DEVCAP = 10; CONFIG =
SMNT;

stargazer:/dev/rmt/tps1d5nsv

stargazer:/dev/rmt/tps0d4nsv
```

This example defines three device pools: a "3490_dp" pool of two 3490 stackers, a "dlt_dp" pool of two DLT STK robotic drives, and a "8mm_dp" pool of two 8mm stackers. All devices are identified a being on a machine called stargazer. Devices might be found found on any host in the network. All lines beginning with a '#' are treated as comments.

pmxDev -r request_name -p pool_name -d device_name -s free | reserved | suspended | deleted -n priority

Display the ProMAX resource database.

Description

pmxDev shows information about all devices and pools managed by this tool suite. If no parameters are input, all details are displayed. The parameters act as selection keys on the total table of data.

Options

-r request_name Show any pools and devices used by this request.

-p pool_name Show details of the status of any devices in this pool.

-d device_name Show details of the specific device.

-s free|reserved|suspended|deleted Show details of any devices with a status agreeing with the supplied keyword.

-n priority Show details of all requests with this priority.

pmxRsv -m logfile -r request_name -n priority device_name/pool_name=flag1[,flag2]... device_name/pool_name=flag1[,flag2]...

Reserve a ProMAX resource.

Description

pmxRsv creates a request for the named resources; the program will block until the resources are available.

Options

-m logfile Log file for messages specific to this request. If not supplied, status information will not be generated except for error messages that are sent to the logfile passed to pmxInit.

-r request_name Request name used to track resource management requests. If not supplied, then the process id of the request is used. If -r is supplied and is numeric, it is assumed to be the process id of a request whose demise indicates that the resources may be released.

-n priority Priority of this reserve request, the default value is 10.

device_name/pool_name The specific device or a pool of devices from which a specific device will be selected.

flag1[,flag2]... Tokens to be associated with the allocated device. If supplied, this will be written into the ProMAX runtime directory's DEVICES file followed by the allocated device and its properties. Once access has been granted to the requested resources, details of the specific resources and their properties (if any) are written to the DEVICES file in PROMAX_RUNTIME_DIRECTORY if that environment variable is set or in the user's home directory otherwise. The device record will be written once per flag. It has the following format:

```
{flag 1} device {device properties} {pool properties}
{flag i} device {device properties} {pool properties}
```

Note: If a DEVICES file already exists, pmxRsv will fail with an error message. pmxRls must be run first.

pmxRls -d device_name -r request_name

Release a ProMAX resource.

Description

pmxRls releases a device or all devices reserved under the request name.

Options

-d device_name The device name as it appears in the config file, if this device is not currently reserved, an error is reported to the relevant log files.

-r request_name All devices reserved by this request (and still reserved) are released.

pmxSet { -p pool | -d device suspend | resume } { -r request_name -n priority -f }

Set various database maintenance options.

Description

pmxRls releases a device or all devices reserved under the request name. pmxSet is an operator management routine. It allows for the temporary suspension of device or pool availability and their subsequent resumption. It also allows for the priority of a request to be raised or lowered or for the request to be destroyed and its resources returned to the free pool(s).

Options

-p pool_name Name of pool to be suspended or resumed.

-d device_name Name of pool to be suspended or resumed.

suspend Remove the specified pool or device from consideration for usage. Running requests will complete but no new ones will be granted access.

resume Resume normal consideration of the device or pool.

-r request_name Apply a new priority (via the -n flag) to this request.

-n priority Assign this request a new priority number between 0 and 19.

-f This request is forced to release any reserved devices.

Using TapeDiag Tape Diagnostics

You use tapediag to a diagnose tape problems, determine tape performance issues, and prelabel tapes when using the proclient SMNT scratch mount mode. There are also various tapediag script wrappers in $PROMAX_HOME/sys/bin/ which perform specific functions (e.g. tapediag_prelabel to pre-label tapes, tapediag_maxblock to determine the maximum block size supported on a device, etc).

The executable $PROMAX_HOME/sys/bin/tapediag is built from the same code base from which $PROMAX_HOME/sys/exe/proclient is compiled. In fact, it accepts the same arguments in the same positional order that proclient does. With this implementation, it gives the user the ability to perform unit tests on tape devices and tape media without actually running ProMAX jobs. tapediag can be run from the command line or packaged within scripts.

Before beginning diagnostics, you should verify that the device is ready to test. These tests should be run on scratch tapes as labels and data on the tapes may be destroyed. Tests vary in length, taking from a few minutes to several hours to run.

TapeDiag Commands

tapediag DIAG="option:option:..."

The Tape Diagnostics options are invoked by setting the DIAG variable to a colon-delimited list of diagnostic options. You may either run these diagnostics as standalone tests or, since tapediag is very similar to proclient, they may be included as ProClient arguments.

FBRWV

Fixed Block Read Write Verify test.

Usage

This test will write 7 pre-determined bit patterns to tape, rewind the tape, then read and verify the results. The number of blocks written per test is 256 at a configurable block size.

VBRWV

Variable Block Read Write Verify test.

Usage

This test will write 7 bit patterns to tape, rewind the tape, then read and verify the results. The block size is determined via a random number generator and will be limited to greater than four and less than maximum block size. The size of block is stored in the first four bytes of the block written. The block size read is compared against the value stored in the block. The bit patterns are also verified. The number of blocks written per test is 256 blocks by 7 bit patterns.

MFRWV

Multi-File Read Write Verify.

Usage

This test writes 64 files, 8 blocks per file of variable block size, with 7 different bit patterns and serves as a device stress test. The block size is stored in the first 4 bytes of the block and verified against the size returned from the read as in VBRWV. 3584 blocks are read and written: 64 files by 8 blocks by 7 bit patterns.

VBPOS

Variable Block POSition test.

Usage

This test writes 256 blocks to tape then generates 256 random positions between blocks 1 and 256 and issues space forward and reverse commands. The actual block position is stored in the block so we will know if we are located properly. 256 blocks at maximum block size and 256 position commands.

MFPOS

Multi-File POSition test.

Other Docs Known Problems

Usage

This test is identical to VBPOS, except that positioning is file-based rather than block-based. This test checks sequential file number and block number within the file for correct location. 64 files, 3 blocks per file by 64 postion commands. File positioning is by far the most taxing of all the tape operations, particularly the skip file reverse. This test will take the longest of all (tests) to complete.

MXBLK

MaXBLocK test.

Usage

This test will empirically determine the maximum block size supported by the device and operating system. This test is particularly useful for identifying long block configuration problems with the operating system. This test runs very quickly.

PERF

PERFormance test.

Usage

Given a starting block size, the test writes 50 blocks at 5, 10, 25, 50, 75, and 100% percent. Perfomance is given in megabytes per second and the I/O response time given in milliseconds. The I/O response is the amount of time it takes for the system to complete the I/O operation. Both read and write for all block sizes are tested. You will notice that read is typically the slower of the two. And while the I/O response increases with larger blocks, it is outweighed by the additional data written. Usually, larger blocks equate to better performance. This is a good test for system device drivers. This test can take awhile depending on the device driver and the block size you start the test with (e.g. a DLT drive will support blocks greater than 4 megabytes) . At the maxumum, the test will write 200 megabytes and read it back.

Using the TapeDump Utility

The TTS based Tape Dump is usually accessed through the script tapedump found in $PROMAX_HOME/port/bin. Being TTS based, the environment that this script is called from needs to have access to the Resource Manager and the Operator Console.

Once the environment is set up, calling tapedump with an argument of the device pool and VSN should allocate a specific device from the pool and request via the operator console that the VSN be mounted on it. The device will then be initialised by the tapediags command and the command line interpreter for the Tape Dump entered. At this stage the screen should look something like :

```
... usual TTS initialisation and option debug data

...

Tapedump, using a similar command set to fb(copyright JH Swaby)

type '?<return>' to get the list of available options

>80[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15  0123456789012345]

00: 56 4F 4C 31 41 41 41 41 44 58 20 20 20 20 20 20 |VO1AAAADX      |

16: 20 20 20 20 20 20 20 20 50 72 6F 4D 41 58 20 35 |        ProMAX 5|

32: 2E 30 20 20 20 32 30 39 20 20 20 20 20 20 20 20 |.0   209        |

48: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 |                |

64: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 34 |               4|

>80[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15  0123456789012345]
```

This is the first block on the tape, or if this block is larger than the default display matrix (16x16) the first 256 bytes of the first block. The number following the '>' at the left end of the header line is the size in bytes of the current block.

Various commands to influence the display format, tape read location and to analyse the tape contents are available. Type '?' followed by <return> to get a complete list of the options.

Options

Display format and address

+[number]	Set increment INC to [number] and increment.
-[number]	Set increment INC to -[number] and increment.
*[number]	Increment [number] * INC bytes.
<return>	Increment INC bytes.
-	Set INC to -INC and increment.
+	Set INC to default (rows * columns) and increment.
#	Display current tape file, block and INC.
#[number]	Display [number] address in various forms.
[number]	Go to block offset position [number].
b|B	Binary display (default 8|16 columns).
c|C	Character display only (default 64|128 columns).
d|D	Decimal display (default 16|32 columns).
h|H	Hexidecimal display (default 16|32 columns).
i|I	IEEE 32|64 bit floating point (4 or 2 columns).
o|O	Octal display (default 16|32 columns).
r|R(’x’)	Move to ’x’th previous block or file on tape.
f|F(’x’)	Skip forward ’x’ block(s) or file(s) on tape.
p|P’x’	Go to block (in current file) or file ’x’.

Matrix, Option and Exit

[r].[c] If r is nonzero, set number of rows to r (range [1..64]). If c is nonzero, override column defaults of above alphabetic commands and set number of columns to c (range [1..128])

A.[c] Display according to A (=b|B|d|D|h|H|o|O|c|C) and restore default column settings if c not present or zero; otherwise, override column defaults and set number of columns to c.

| Toggle between supplemental character display present or absent.

/ Toggle between column labels or no column labels.

! Toggle between compressed or uncompressed numeric display.

~ Toggle EBCDIC character mapping.

<option Mark current position and execute option.

> Go to previously marked position.

^ Toggle between current and previously marked position.

S|s[n] Display synopsis of current tape or file, (limited to n blocks).

? Display this list of commands.

Q|q Terminate program.

Details of commands and results

+[number]

Subsequent <return> commands (including the one excercised here) will cause the display matrix to hold the next [number] bytes.

Subsequent block movement commands will be followd by [number] bytes in the matrix. <return> will move to the next [number] bytes

-[number]

Subsequent <return> commands (including the one excercised here) will cause the display matrix to hold the previous [number] bytes. Subsequent block movement commands will be followd by [number] bytes in the matrix. <return> will move to the previous [number] bytes provided this does not cause a movement to before the current block. Use the r command to get to earlier blocks on tape.

***[number]**

Set the move up between the bytes in the matrix to be [number] times the current increment value

<return>

Move to the next set of bytes, <return> following the initial display shown above displays the second block on tape, in this case the HDR1 record of a standard label.

```
>80[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15  0123456789012345]

00: 48 44 52 31 20 20 20 20 20 20 20 20 20 20 20 20 |HDR1|

16: 20 20 20 20 20 20 20 20 20 20 20 30 30 30 31 30 | 00010|

32: 30 30 31 30 30 30 31 35 30 30 34 31 38 39 34 39
|0010001500418949|

48: 39 39 39 39 39 20 30 30 30 30 30 30 50 72 6F 4D |99999
000000ProM|

64: 41 58 20 35 2E 30 20 20 20 20 20 20 20 20 20 20 |AX 5.0|

>80[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15  0123456789012345]
```

+

Set the display start increment to exactly n-row * n-columns and display the next matrix worth (n-row * n-column) of bytes. Succeeding <return> will display the next matrix worth of displays until the end of file is reached.

-

Set the increment between succesive displays to be equal to the number of bytes visible in the current matrix and step backwards this number of bytes with each succeeding <return> until the start of the block is reached.

#

Display tape position information. For example :

```
[File = 2; Block = 12; INC = default (80)]
```

Other Docs Known Problems

#[number]

Analyse the bytes starting at address [number] in several forms. Including binary and little endian, IEEE floating point and MSbinay floating point. An example follows.

```
Showing interpreted datatypes at File [0], Block [1], Offset [30]

Characters -    bcd     -     Ones     -    Twos

<8-bit>   [1]    - [31]       - [  4.90e+01] - [  9.80e+01]

<16-bit b>[10]   - [3130]     - [  1.26e+04] - [  2.52e+04]

<16-bit l>[01]   - [3031]     - [  1.23e+04] - [  2.47e+04]

<32-bit b>[1000] - [31303030] - [  8.25e+08] - [  1.65e+09]

<32-bit l>[0001] - [30303031] - [  8.08e+08] - [  1.62e+09]

Characters -    binary    -     IEEE     -   MSbinary

<32-bit b>[1000] - [  8.25e+08] - [-3.09e+231] - [-2.68e+154]

<32-bit l>[0001] - [  8.08e+08] - [-3.09e+231] - [-2.68e+154]

Characters    -  1st4   binary   2nd4  -      IEEE       -   MSbinary

<64-bit b>[10001000] - [  8.25e+08,  8.25e+08] - [    -3.0938e+231] -
[overflow]

<64-bit l>[00010001] - [  8.08e+08,  8.08e+08] - [    -3.0938e+231] - [
1.2650e-279]
```

### [number]

Set the address of the first byte in the display matrix to [number]. In the example that follows 32 was used. Since the record is only 80 bytes long only three rows are visible.

```
32: 30 30 31 30 30 30 31 35 30 30 34 31 38 39 34 39
|0010001500418949|

48: 39 39 39 39 39 20 30 30 30 30 30 30 50 72 6F 4D |99999
000000ProM|

64: 41 58 20 35 2E 30 20 20 20 20 20 20 20 20 20 20 |AX 5.0          |

>80[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15  0123456789012345]
```

b|B

Display each of the bytes in binay form with 8 or 16 bytes per row. Here is the 8 byte example with the auxillary character display turned off.

c|C

Display the auxillary character display only, either with 16 or 32 bytes per row. Here is a 16 byte example :

```
00: |HDR1            |

16: |           00010|

32: |0010001500418949|

48: |99999 000000ProM|

64: |AX 5.0          |

>80 [0123456789012345]
```

d|D

Display the bytes in decimal, either 16 or 32 per row. Here is an example of d again without the auxillary character display

```
00: 072 068 082 049 032 032 032 032 032 032 032 032 032 032 032
032

16: 032 032 032 032 032 032 032 032 032 032 032 048 048 048 049
048

32: 048 048 049 048 048 048 049 053 048 048 052 049 056 057 052
057

48: 057 057 057 057 057 032 048 048 048 048 048 048 080 114 111
077

64: 065 088 032 053 046 048 032 032 032 032 032 032 032 032 032
032

>80[  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15 ]
```

h|H

Display the bytes in hex, either 16 or 32 per row. The 16 byte form is the default at startup.

i|I

Display the main matrix as IEEE floating point numbers. The matrix width (in bytes) will be set to the next lower integer multiple of the number of bytes in the float type and the number of columns in the matrix will be nrows / n bytes in float.

Other Docs

Known Problems

o|O

Display the bytes in octal, either 16 or 32 per row. Here is an example of o again without the auxillary character display

```
00: 110 104 122 061 040 040 040 040 040 040 040 040 040 040 040
040

16: 040 040 040 040 040 040 040 040 040 040 040 060 060 060 061
060

32: 060 060 061 060 060 060 061 065 060 060 064 061 070 071 064
071

48: 071 071 071 071 071 040 060 060 060 060 060 060 120 162 157
115

64: 101 130 040 065 056 060 040 040 040 040 040 040 040 040 040
040

>80[ 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 ]
```

r|R('x')

Move back on tape to the previous block or file. If a numeric suffix is supplied it is the number of blocks or files to skip.

f|F('x')

Move forward on tape to the start of the next block or file. If a numeric suffix is supplied it is the number of blocks or files to skip.

p|P'x'

Move the tape to block 'x' or file 'x' and redisplay. The block and file numbers are relative to zero so p0 should move to the first block in the current file.

[r].[c]

Change both dimensions of the display matrix but not the dump type. Assuming the default (starting) h 16x16 the hex display can be made to show 10 rows by 12 columns with 10.12 which generates a display like this:

```
>80[ 0  1  2  3  4  5  6  7  8  9 10 11  012345678901]
00: 56 4F 4C 31 41 41 41 41 44 58 20 20 |VOL1AAAADX  |
12: 20 20 20 20 20 20 20 20 20 20 20 20 |            |
24: 50 72 6F 4D 41 58 20 35 2E 30 20 20 |ProMAX 5.0  |
36: 20 32 30 39 20 20 20 20 20 20 20 20 | 209        |
48: 20 20 20 20 20 20 20 20 20 20 20 20 |            |
60: 20 20 20 20 20 20 20 20 20 20 20 20 |            |
72: 20 20 20 20 20 20 20 34             |       4|
```

A.[c]

Here is the result of d.10 decimal, 10 columns:

```
>80[  0   1   2   3   4   5   6   7   8   9  0123456789]
00: 086 079 076 049 065 065 065 065 068 088 |VOL1AAAADX|
10: 032 032 032 032 032 032 032 032 032 032 |          |
20: 032 032 032 032 080 114 111 077 065 088 |    ProMAX|
30: 032 053 046 048 032 032 032 050 048 057 | 5.0   209|
40: 032 032 032 032 032 032 032 032 032 032 |          |
50: 032 032 032 032 032 032 032 032 032 032 |          |
60: 032 032 032 032 032 032 032 032 032 032 |          |
70: 032 032 032 032 032 032 032 032 032 052 |         4|
>80[  0   1   2   3   4   5   6   7   8   9  0123456789]
```

|

Toggle off or on the supplemental character (ASCII or EBCDIC) display. This is on by default, when turned off the 80 byte record looks like this:-

```
>80[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 ]
00: 56 4F 4C 31 41 41 41 41 44 58 20 20 20 20 20 20
16: 20 20 20 20 20 20 20 20 50 72 6F 4D 41 58 20 35
32: 2E 30 20 20 20 32 30 39 20 20 20 20 20 20 20 20
48: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
64: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 34
>80[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 ]
```

/

Toggle off or on the column labels, here is the 80 byte record above without headers :-

```
00: 56 4F 4C 31 41 41 41 41 44 58 20 20 20 20 20 20 |VOL1AAAADX
|
16: 20 20 20 20 20 20 20 20 50 72 6F 4D 41 58 20 35 |        ProMAX 5|
32: 2E 30 20 20 20 32 30 39 20 20 20 20 20 20 20 20 |.0   209        |
48: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 |                |
64: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 34 |               4|
```

!

Toggle between compressed and uncompressed display. For example here is the compressed version of the 80 byte record given in the introduction above:-

```
>8[  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 0123456789012345]
00:564F4C3141414141445820202020202020 |VOL1AAAADX      |
16:20202020202020202050726F4D41582035 |        ProMAX 5|
32:2E302020203230392020202020202020 |.0   209        |
48:20202020202020202020202020202020 |                |
64:20202020202020202020202020202034 |               4|
>8[  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 0123456789012345]
```

~

Toggles the auxillary character display to treat the input as EBCDIC and convert it to ASCII for display.

<Option

Register the current location and execute the option. The recorded position inclues the file, block and address information. To subsequently jump to the recorded position use the > or ^ options.

>

Go to an address registered by <option or as a side effect of ^ . To jump to a specific location see also p|P'x'.

^

Go to an address registered by <option or as a side effect of an earlier execution of this command. That is the current address is recorded so that subsequent executions toggle between the two locations. To jump to a specific location see also p|P'x'.

S|s[n]

Generate a synopsis of the Tape or File. For example here is the output from a S for a tape that just has standard labels on the front

```
Generating synopsis of current Tape

3 blocks of 80 bytes read.

Sequential File Mark . . . [1]

Sequential File Mark . . . [2]

. . . . . Logical End Of Volume . . . . .

Total number of blocks read . . : 3

Total number of file marks read : 2

Total number of bytes read  . . : 240
```

After completing the synopsis the program attemts to put the matrix display back to the position before S or s was invoked.

?

Display the on line help

Qlq

Exit the tape dump. The tapediags routine should unmount the tape and notify the operator console.

References

The folowing links take you to reference material for TTS.

Glossary

Term	Definition
<.> key	Keyboard keys will be referenced by the "<.> key," with the character(s) which appears on the key being enclosed in brackets.
[...] button	Buttons on graphical interfaces will be referenced by as "[...] button," with the text which appears on the button being enclosed inside square brackets.
/superuser/unix/prompt #	A '#' preceding a command-line example signifies the UNIX prompt of the 'root' superuser, which means that the execution should be performed only as 'root' superuser. If the '#' is also preceded by a directory path, it means the command-line execution should be performed from that directory.
/user/unix/prompt >	A '>' preceding a command line example signifies the UNIX prompt of a general user, meaning that the execution can be performed by any user. If the '>' is also preceded by a directory path, it means the command line execution should be performed from that directory.
compute host	Machine on which ProMAX executive is run.
device host	Host to which the drive is attached. A partial ProMAX installation must exist on this host. In addition, the hosts containing PROMAX_DATA_HOME and (if TopCat is being used) PROMAX_TOPCAT_HOME must be NFS mounted to this host on a parallel file system structure.
device management system (PMX)	Manages the allocation, deallocation, reallocation of device resources.
DSN	Catalog dataset name.

Other Docs Known Problems

Term	Definition
media pool	Name for specific tape media in catalog. For example, 3598,DLT.
OpCon host	Host running the OpCon tape operator.
operator console (opcon)	The graphical user interface for the tape operator to receive, monitor, and respond to tape mount requests and manage devices.
PMX host	Host containing the PMX device resource database: PROMAX_RDB_HOST, PROMAX_ETC_RDB_HOME.
ProClient/proclient	The primary tape management system executables. ProClient is a shell script wrapper for the proclient executable.
ProMAX data host	Host containing the flow directory (system where PROMAX_DATA_HOME physically resides).
ProMAX host	Host on which a full ProMAX installation exists and on which the flow is executed (system where PROMAX_HOME physically resides).
robot controller	Host on which the ACSLS server software is installed. The actual drives do not have to be attached to this machine; the robot is responsible for placing tapes in the drive(s) and noting which tapes are contained in the robot. However, certain ProMAX scripts and executables must be installed on this host.
ShowCat	Graphical interface to the TopCat catalog.
ShowCat host	Host running the ShowCat tape catalog GUI.
tape catalog media manager (topcat)	Stores, retrieves, and modifies information in a database that it creates to describe tape volumes.

Other Docs Known Problems

Term	Definition
tape host	Machine to which a device is physically attached.
tape management system (TTS)	Central control system which manages input/output interactions with simple, stacker and robotic devices, tape catalog communications during tape input/output, volume access logic, and generation of mount requests and job status notifications.
TopCat host	Host containing the TopCat tape catalog database.
VSN	Volume serial number for identifying tape. VSN can be internally labeled (SL or AL), where identifiying records are written at the front of the tape, or non-labeled (NL), where the only identification is a printed label on the outside of the cartridge..

Other Docs Known Problems

About ProClient

ProClient is the fundamental Total Tape Solution data service client. It supplies the basic functionality of reading and writing tape and tape-on-disk data and offers the following features:

- Completely self-contained module that can be installed on any ProMAX-compatible computers. No additional ProMAX installation or license is required for those machines.
- Supports the following operator mount request mechanisms:

  The traditional ProMAX tape mount X-Window popup (default).

  Java-based operator console. Activated when the OPHOST key in the PMX Resource database or device.tts file is set to the hostname where the console is running.
- ANSI and IBM standard label tape processing support. In these standards, a volume label is normally composed of three 80-byte records followed by a single file mark. Contained in these three 80-byte records is information we may use to limit or grant a user's access to tape. There are four basic volume access checks which can be made for label processing.

  VSN check: does the six-character VSN string on tape match what the operator was requested to mount?

  Label type: there are 3 types of labels: ANSI (AL), IBM (SL), or Non-labeled (NL). Is this the label type which was expected by the requesting job?

  Dataset name: this name is 17 characters in length and can be used to determine if an unknown tape is the correct one to be read.

  Under most circumstances, you will want all of these label checks activated. The system can, however, be customized to allow jobs to bypass any or all of these checks and with respect to how errors are handled.
- Full catalog support.

Other Docs Known Problems

- Extensible tape-on-disk encapsulation support. Encapsulation is a process in which the original tape data records are copied to disk in with extra encoding so that the records can be individually reconstituted. There are many ways to do this. ProClient has support for three common encapsulation formats. Contact Landmark customer support for information on developing additional local encapsulation formats.

Other Docs Known Problems

ProClient Command Arguments

Introduction

The following section details the command arguments available on ProClient. Arguments are categorized by their method of appearance: by user menu selections, device configuration, and executive. The section concludes with the arguments currently not available for use.

proclient hostname socket_port arg1 arg2 ...

The central TTS binary proclient is called almost exclusively by ProClient or other ProMAX scripts. It's arguments is assembled as follows:

Options

hostname is the hostname of machine where server program is running. It is ***mandatory***.

socket_port is the socket port number over which proclient communicates. proclient binds to this number. It is ***mandatory***.

arg1 arg2 ... are optional input arguments to proclient. You must use double quotes around any string which contains blanks.

Note: A blank is interpreted as an end of argument character, otherwise, the string "Your favorite dataset name" would become four separate command line arguments, making the decoding more difficult.

Usage

The catalog can be opened and closed internal to proclient program or externally as done in **NetClient**. Many of the command line arguments listed are in support of the internal catalog interface.

For information on catalog related options, refer to the ***OPNDSN*** and ***CLSDSN*** features in ***TopCat***.

Much of the tape system start-up information comes from the executive via the inband data path. There are command line arguments which provide a means of over-riding this inband configuration data.

Arguments provided by user menu selections

(explicitly or via catalog calls)

CATNAME="Exploration Dept"

Catalog name.

Usage

Catalog to use. TopCat supports multiple catalog definitions. CATNAME points to the default or an alternative catalog.

DSN="dataset name"

Dataset name.

Usage

Value obtained via user menu input, which may include blanks. The dataset name has two possible uses: catalog access and for storing in tape volume labels. DSN catalog access is a more common usage of the variable. Datasets created under catalog control are stored with the DSN reference as a component of the CIND file.

EXPD="julian_expiration_date"

Julian expiration date.

Usage

Julian date when the volume may be scratched. Value is stored both in the catalog and on the volume label. Needed only if writing labeled tapes or cataloged datasets.

LABL="NL | AL | SL"

Label type.

Options

NL: No label.

AL: ANSI label (ASCII)

SL: IBM standard label (EBCDIC)

Usage

Value obtained from user menu input. With full label processing enabled, label type is required to match. Other factors such as catalog control may come into play. However, there are features in place to override a label type access violation.

LOGICALDEV="tape0"

Logical device name.

Usage

This is the logical device name referenced by the first field in the ***device.tts*** file (e.g. tape0) and is obtained from user-selection of the device list from the ProMAX User Interface.

MASK="VOL000/:1-3/" or "01-10" or "8MM999-8MM995"

VSN mask and/or list. Alternate specification of VOL list.

Usage

Custom volume list is built by superimposing list on mask or expanding ascending or descending list range. For use if the catalog is being bypassed or with tapediag_prelabel.

RETD="n_days"

Retention period days.

Usage

Retain this tape for N days after creation. After that period, it may be scratched. Value is used to calculate a julian expiration date. This value is stored both in the catalog and on the volume label.

VOL="VOL001,VOL002,VOL003"

Volume list.

Usage

Mandatory option if the catalog is opened external to ProClient or the catalog is being bypassed.

Arguments provided by device configuration

(from association with LOGICALDEV)

ASYNC="n"

Asynchronous tape I/O switch. ***(Experimental)***

Usage

Set to nonzero to enable multithreading in *proclient*. One thread communicates with the ProMAX Executive while the other reads or writes the tape device.

Empirical tests have shown that in most cases the natural parallelism of *proclient* running as a separate server process from the ProMAX Executive allows data to stream as fast as the tape device can handle without resort to multithreading. Indeed, the extra overhead of threading simply slows the processing down.

CLIENT="ProClient"

Client program which will determine device options and initiate I/O communication with the ProMAX executive.

Usage

Value is obtained via user selection of the device list from the ProMAX User Interface. CLIENT is a required entry in either the device.tts or rdb/config_file device definitions.

The default value of "ProClient" invokes full functionality.

Setting CLIENT to a user-defined script or binary would allow integration with non-ProMAX tape systems.

DEVCAP="n"

Device media capacity.

Usage

Mainly for manual stackers (DEVTYPE=STACKER), this should be set to the maximum number of tapes which can be loaded into the stacker.

DEVHOST

DEVICE="hostname:/dev/rmt0"

Hostname and UNIX device name.

Usage

Value is obtained via user selection of the device list from the ProMAX User Interface. DEVICE may be specified with UNIX options such as density and rewind settings.

DEVINFO="0,0,2,1 /mnt/acshost/cmd_proc"

Device information.

Usage

Any additional information about a device such as drive info for ACSLS or ADSM.

DEVTYPE="SIMPLE | STACKER | STKROBOT | IBMROBOT"

Device type.

Options

SIMPLE is the simple device with a media capacity of one volume.

STACKER is the dvice which has a media capacity of greater than one.

STKROBOT is the StorageTek automated cartridge loader.

IBMROBOT is the IBM automated cartridge loader.

Usage

Value is obtained via user selection of the device list from the ProMAX User Interface. DEVTYPE causes other device information to be passed or commands to be invoked which determine device behavior.

DPOOL="device_pool_name"

Device pool name.

Usage

Value is obtained via user selection of the device list from the ProMAX User Interface. DPOOL is used as a cross-reference to determine which catalog media is required.

Note: media type is a factor only when creating a catalog dataset. The DPOOL is matched with those defined in the catalog administrator's file (catadmin). When no catalog is in use, DPOOL serves solely as a user/operator information field, since it is displayed in mount messages.

LIMO="output_limit"

Maximum amount to write to a tape, in kilobytes.

Usage

This value allows the user to specify the amount of data written to any one volume, before a new VSN is requested.

MAXBLK="block_size"

Maximum block size to read or write with the device, in bytes.

Usage

This value overrides the standard maximum block size of 65535. Most old multiplexed field tapes have tape records greater than this, often several million bytes, as do many new multitrace blocked tape datasets, and MAXBLK needs to be set to be at least as great as the longest tape record expected. (Note that IBM and ANSI tape labels allow only 5 digits for their tape block and record length fields, preventing *proclient* from automatically setting MAXBLK.) SCSI devices have an upper limit 16 Mb, which may not be achievable in practice without special hardware and operating system support.

You may use the TTS ***TapeDiag*** utility to determine what maximum length tape block an individual device will support and use this value in ***device.tts***. Be aware, however, that if you disable adaptive reads with CONFIG=BAR (described below), tape performance will usually be poorer than if MAXBLK is set to a value just big enough to accomodate the largest record on the tape dataset being read.

MFM="number_of_file_marks_between_files"

Number of file marks (EOF) to expect between files. Normally multiple consecutive EOFs are treated as signalling end of data (EOD) on a tape volume.

NRET="number_of_retries"

Number of retry attempts made on an I/O operation before reporting an error.

Usage

Default is to make 5 retry attempts, waiting 10 seconds (parameter RWAIT) after each unsuccessful attempt. Further action or error reporting proceeds after this 50 second period.

OPHOST="host_name" or "host_name:rmi_port"

Host where Operator Console is running.

Usage

Mount and status messages will be displayed on the JAVA Operator Console running on this host. If necessary, messages can be sent to Consoles running under different RMI port numbers on the same host. This might be useful for Consoles managing groups of devices in different physical locations.

OPNOT="TAPE_NOTIFICATION"

Script used to send notification to the Operator Console.

Usage

The named script defines the environment and sends tape status notification to the Operator Console via the java RMI server.

OPREQ="TAPE_REQUEST"

Script used to send operator requests to the Operator Console.

Usage

The named script defines the environment and sends tape mount requests to the Operator Console via the java RMI server.

REMHOME="PROMAX_HOME "

Full path to PROMAX_HOME on remote host.

Usage

Defaults to PROMAX_HOME on flow execution host.

RINTF="AAR_STKCMD"

Name of script used to address robotic interface.

Usage

The default setting uses the AAR_STKCMD script, found under $PROMAX_HOME/sys/exe to set options and call the binary which sends tape mount/dismount commands to the STK robotics interface via rsh or aar_promax communication. Users couldprovide their own routines to connect to other interfaces.

RWAIT="retry_wait_time"

Wait time in seconds between retry attempts on a pending I/O operation.

Usage

Default is to make 5 retry attempts (parameter NRET), waiting 10 seconds after each unsuccessful attempt. Further action or error reporting proceeds after this 50 second period.

TMHOST="host_name"

Host to which tape mount messages should be sent.

Other Docs Known Problems

Usage

Mount messages will be displayed on this host via the "tape_mount.exe" pop-up window. Used when the Operator Console is not running. The default is to display mount messages on the host to which the device is attached, or secondarily on the terminal from which the job was executed.

UNLDELY=nn

Wait time in seconds after issuing the unload request to reopen the stacker.

Usage

Default is to wait 10 seconds after each unload request.

CONFIG options:

The following section outline the nineteen (19) different options for tape runtime configuration.

CONFIG="option1+option2+... "

CONFIG="option1;option2;... "

CONFIG="option1=option2=... "

CONFIG="option1"; CONFIG="option2 "; CONFIG="... "

Runtime Options

AEEJ: Automatic Error EJect ejects tapes due to errors. If this is set, jobs which have non-fatal volume access violations will eject the problem tape and [abort the joprompt for the correct tape. If the device is a stacker, all tapes in the stack are checked before the mount is issued. In default mode (AEEJ not set), a mount message identifying the volume access violation appears in the Tape Mount or Operator Console. The error condition can be overridden by clicking the [**Mount**] button on the errored second mount request. Affects both input and output.

BAR: By-pass Adaptive Read. This disables the adaptive read that attempts to read a tape with the most commonly occuring blocksize.

BCAT: Bypass CATalog.

BCVC: Bypass Catalog VSN Cross-checking. Any input or output VSN which the user specifies as not under catalog control (e.g., Selecting "External VSN list") is still checked to see if it is an element of the active catalog. If so, the job fails. Setting BCVC bypasses this check to avoid coincidental clashes.

BLDP: Bypass Label Dataset name Processing. For input and output volumes not under catalog control, this disables the check that the dataset name specified in the menu matches that on the tape label.

BLEP: Bypass Label Expiration date Processing. For output volumes not under catalog control, this disables the check that expiration date on the tape label is expired. For example, less than or equal to current date.

BLLP: Bypass LabeL type Processing. For input and output volumes not under catalog control, this disables the check that the label type (e.g., SL) specified in the menu matches that of the tape label.

BLP: Bypass all Label Processing. For input and output volumes not under catalog control, this disables all checking to verify parameters specified in the menu match those on the label.

BLVP: Bypass Label VSN Processing. For input and output volumes not under catalog control, this disables the check that the VSN specified in the menu matches that on the tape label.

DBUG: Enable DeBUGging.

DEOF: Double End Of File. Forces ProClient to explicitly write a double EOF before closing the volume. The option is almost always not needed.

FMNT: First MouNT. If the tape is loaded in the drive and ready, the job continues without sending a mount message to or requiring a response from the Operator Console.

Otherwise, a mount message is generated. This behavior is equivalent to pre 7.3 (except 7.0p*) versions of ProMAX which used the tape_mount.exe. If FMNT is not set, a mount request will appear, even if the tape is already loaded in the drive.

LMNT: Last MouNT. Issue another mount after end of job. If this is set, the job prompts for another tape mount after it has processed its list of VSNs. If LMNT is not set, the job terminates after the last VSN that you specify is processed.

Note: If you select **Volume is mounted and device is ready** instead of selecting **Cancel Mount/End Process** after the last tape, the job waits for another tape.

MFIL: specifies that the labeled tapes have, or will have, multiple dataset files on each volume.

NCAT: No CATalog in use. This option automatically sets if no catalog is found.

NCON: No CONsole, that is, batch mode. This bypasses any mount messages. The job proceeds only if the tape is loaded in the drive and ready. Otherwise, the job fails.

Note: If the **mask list for external labels** contains more tapes than are in the drive, the job waits (without notification) until the next tape is loaded into the drive.

NREW: Do Not REWind at EOT or EOJ.

NSNS: Bypass device SENSE queries to verify device is online.

NUNL: Do Not UNLoad at EOT or EOJ.

PTOD: ProMAX Tape-On-Disk enabled. ***(Obsolete.)***

SEGYBLK: Read or write superblocked SEG-Y data, i.e. more than one trace per physical tape block.

SKIP128: Silently discard any 128 byte records found at the beginning of a tape or immediately after a file mark. This prevents unexpected tape labels from causing jobs to fail.

SLAB: Show (dump) tape LABels. Prints to job.output the contents of the tape label.

SMNT: Enable catalog Scratch MouNT option.

If this option is set, and it should only be used for manual devices, ProClient will operate in a "user scratch select" mode whereby the mount request to the OpCon or tape_mount.exe window will just be for "SCRATCH", and not for any specific VSNs. In this SMNT mode, the user/operator then mounts the tape(s) and ProClient will write to whatever tape it encounters after ascertaining from the tape label that the tape has a VSN which matches the Catalog's VSN, media pool, and device pool associations, and that the VSN's VOLSTAT is "SCRATCH" or "NEW". For this reason of label verification on scratch volumes using the SMNT mode, all scratch tapes being introduced into the Catalog must be prelabeled. The script $PROMAX_HOME/port/bin/tapediag_prelabel may be used to prelabel new scratch tapes.

If this option is not set, when writing to manual devices, ProClient will operate in a "catalog scratch select" mode whereby it selects scratch VSNs from the Catalog (VOLSTAT=SCRATCH|NEW) and issues a mount request to the OpCon or tape_mount.exe window for the specific scratch VSNS to be mounted. The next scratch VSN which is used will always be the next sequential VSN in the Catalog which has a VOLSTAT of "SCRATCH". When SMNT is not defined for a tape device, a scratch mount request may be fulfilled by mounting a non-labeled tape, such as a new tape.

DEBUG options

The following section outlines the ten (10) debugging options available.

DEBUG="option1:option2:..."

Usage

The debug options within ProClient will allow you to monitor a particular subsystem. The additional job.output lines from these options will help us to help you when something has gone wrong.

Options

DECO: ProClient command line inputs decoder.

DECEX: Exit ProClient after decode step.

CAT: Tape catalog (topcat) interface.

SVOL: Internal volume manager.

VERI: Volume access verification.

CONS: Mount control and operator console interface.

IO: I/O subsystem (open/close/read/write/ioctl).

CMDL: ProClient main command process loop.

STK: StorageTek robot interface

ALL: Monitor everything.

DIAG options

DIAG="option1:option2:..."

The DIAG options are described on the ***Tape Diagnostics*** page.

Arguments provided by executive

CATINFO="/tmp/dsn_name.1"

Dataset information file name.

Usage

Catalog option. Like VSNINFO, this option provides a conduit for information common to all volumes in a dataset to be stored in the catalog. This information must be available at the close of a catalog dataset operation.

MPOOL="media_pool_name"

Catalog media pool name.

Usage

Value obtained via cross reference from device pool name and catalog information.

OPNMODE="DSCREATE | DSOVRWRT | DSOLD | DSAPND"

Catalog open mode.

Usage

Primarily used for archives.

Options

DSCREATE creates a new dataset, obtaining volumes from the media scratch list.

DSOVRWRT replaces existing dataset with a new one, reusing the existing volumes belonging to that dataset.

DSOLD opens an existing dataset.

DSAPND appends to an existing dataset.

RESVN="n"

Number of volumes to reserve when creating a dataset.

Usage

Catalog option. Value defaults to the device capacity.

VSNINFO="/tmp/vsn_number.1"

VSN information file name.

Usage

Catalog option. This is a means of transfering runtime, individual volume-specific catalog information from the executive to the catalog. This information must be available when a catalog VSN close operation is executed. Header word ranges may be stored to the catalog at that time.

Currently not implemented

DSVER ="n"

Dataset version number.

Usage

Dataset version number is tracked by the tape catalog option. This option is currently not being used.

PASSCODE="password"

Dataset password.

Usage

Password used to restrict access to a catalog dataset. This option is currently not being used.

ProClient Rules For ProMAX Tape Volume Access

"When is it safe to write on a tape?"

"When do we fail on input access?"

The decision tree which follows clarifies what will happen under the given circumstances, such as running with or without a catalog, using these definitions:

! = not	Cat = ProMAX Tape Catalog
= = equals	!= = not equals
Out = mounted output tape	NA = Not Available or does not exist
NL = non-labeled	AL = ANSI label
SL = IBM standard label (EBCDIC)	BLP = bypass label tape processing
PASS = validated for access	FAIL = volume rejected (access denied)
USERR = fatal error condition raised at user level	OPERR = error to operator (on robotic tape system, there is no operator and all OPERR errors become USERR); can or cannot be overrideable. If auto-error eject (AEEJ) is set, non-fatal conditions result more operator prompts until either the user kills the job or the operator cancels the mount. If AEEJ is not set, mounting the same tape a second time implies an override of an non-fatal condition. Non-fatal conditions include mismatches in DSN name and label type.

General rule: Unless there is a catalog present, a damaged label tape always FAILs. Damaged refers to the ability to

Other Docs

Known Problems

make a decision based on expiration date. If Out is labeled, damaged, and exists in catalog, then enough information (in catalog) is present to make a safe decision.

Catalog Active: Volume Access Criteria

INPUT

Non-Catalog DSN mount request (External VSN list)

NL mount request

- NL mounted: PASS:BLP

  Input of NL non-Catalog tapes is always allowed.

- AL-SL mounted: FAIL:OPERR

  If expected tape was NL, but mounted tape was AL-SL, Operator Console prompts for Abort/Retry correction.

AL-SL mount request

- AL-SL-NL mounted: FAIL:USERR

  If the Catalog is bypassed via "External VSN List" usage in the menu, the VSN may not coincide with one already in the Catalog, unless the proclient switch BCVC (Bypass Catalog Volume Cross checking) is set.

Catalog DSN mount request

NL mount request

- NL mounted: PASS

  Non-label Catalog tapes are mainly field data, and are matched solely by their external label; i.e. there is no way to insure that the external label info is correct or that the correct tape has been mounted. Operator must confirm the external VSN of NL tapes by typing the VSN again into a mount window pop-up.

- AL-SL mounted: FAIL:OPERR

  If expected tape was NL, but mounted tape was AL-SL, Operator Console should prompt for Abort/Retry correction.

AL-SL mount request

- NL mounted: FAIL:OPERR

If expected tape was AL-SL, but mounted tape was NL, Operator Console should prompt for Abort/Retry correction.

- AL-SL mounted: VSN match: PASS

  Only VSN is checked currently. If it matches, the job continues.

- AL-SL mounted: VSN !match: FAIL:OPERR

  VSN matching can currently be bypassed via PROMAX_VSN_BYPASS. If not set, a VSN match causes the job to fail.

OUTPUT

Non-Catalog DSN mount request (External VSN list)

NL mounted: PASS

Output of NL non-Catalog tapes is always allowed.

AL-SL mounted

- VSN in Catalog: FAIL:OPERR

  If expected tape was NL, but mounted tape was AL-SL and the label exists in the Catalog, Operator Console prompts for Abort/Retry correction.

- VSN !in Catalog, !Expired: FAIL:OPERR

  If expected tape was NL, but mounted tape was AL-SL and the label does not exist in the Catalog but it has a current Expiration Date, Operator Console prompts for Abort/Retry correction. This may need to be configurable: Catalog operation which sole depends on "VolStatus" of active/scratch, or Catalog which depends on both "scratch" Status and expired ExpDate on label. Expiration Date can currently be bypassed via PROMAX_EXP_BYPASS. Should make this a 'catadmin' configuration.

- VSN !in Catalog, Expired: PASS

  If expected tape was NL, but mounted tape was AL-SL and the label does not exist in the Catalog and it has an expired Expiration Date, this is okay for output. This may need to be configurable: Catalog operation which sole depends on "VolStatus" of active/scratch, or Catalog which depends on both "scratch" Status and expired

ExpDate on label.

- External VSN List insufficient FAIL:USERR

  If there are no volume access violations, just that the user's specified external VSN list was exhausted when the job needed more output tapes, the job should abort with a diagnostic message requiring the user to input more VSNs to their list.

Catalog DSN mount request

NL mount request

- NL mounted: FAIL:USERR

  Because non-label Catalog tapes are mainly field data which will solely by matched by its external label (i.e. there is no way to insure that the external label info is correct or that the correct tape has been mounted), NL Catalog tapes are considered "read-only" and output to them is forbidden.

- AL-SL mounted: FAIL:USERR

  If expected tape was NL, but mounted tape was AL-SL, Operator Console should prompt for Abort/Continue correction.

AL-SL mount request

- NL mounted (VOLSTAT=NEW): PASS

  When a volume is first introduced to the Catalog via topcat POPULATE, its VOLSTAT will be flagged as "NEW". Any non-labeled tape may be used as output in this case.

- NL mounted (VOLSTAT=SCRATCH): FAIL:OPERR

  The operator can override this.

- NL mounted (SMNT mode): FAIL:OPERR

  In SMNT scratch mount mode, all output tapes must be prelabeled.

- AL-SL mounted: Catalog VSN match: PASS

  If the tape was prelabeled by MVS, it will only have VOL1 and HDR1, but not have HDR2. MVS prelabels HDR1 with 76 zeros so that the Expiration Date field of "000000" implies an ExpDate of year 2000. If ProMAX encounters an MVS prelabel tape (by checking for the absence of HDR2), it disables ExpDate checking and proceeds to output updated 3x80-byte labels.

- AL-SL mounted: Catalog VSN !match: FAIL:OPERR
- AL-SL mounted: nonCatalog VSN, !Expired: FAIL:OPERR

  If label does not exist in the Catalog but it has a current Expiration Date, Operator Console prompts for Abort/Retry correction. This may need to be configurable: Catalog operation which sole depends on "VolStatus" of active/scratch, or Catalog which depends on both "scratch" Status and expired ExpDate on label.

- AL-SL mounted: non-Catalog VSN, Expired: PASS

  If the tape was prelabeled by MVS, it will only have VOL1 and HDR1, but not have HDR2. MVS prelabels HDR1 with 76 zeros so that the Expiration Date field of "000000" implies an ExpDate of year 2000. If ProMAX encounters an MVS prelabel tape (by checking for the absence of HDR2), it disables ExpDate checking and proceeds to output updated 3x80-byte labels.

Note: The system must allow for data export to other processing centers, and users are able to write this external data without disabling an active Catalog.

- NL tapes should always be read/written without any Catalog checks.
- Certain Device Pools may be defined to automatically bypass the Catalog, e.g. 8mm, regardless of whether the tape is labeled ornot.
- You can list VSNs in the tape menu as 'External', meaning that they are external to the Catalog, and after checking fo coincidental VSNs, Catalog bypass is allowed.

Catalog Not Active: Volume Access Criteria

INPUT

NL mount request

- NL mounted: PASS:BLP

  Input of NL nonCatalog tapes is always allowed.

- AL-SL mounted: FAIL:OPERR

  If expected tape was NL, but mounted tape was AL-SL, Operator Console should prompt for Abort/Retry orrection.

AL-SL mount request

- NL mounted: PASS
- AL-SL mounted: VSN match & Owner match: PASS

  Only VSN is checked currently. Owner may be checked later, based on some security-related configuration option in 'catadmin'.

- AL-SL mounted: VSN match & DSN !match: FAIL:OPERR

  Override with BLDP/BLP.

- AL-SL mounted: VSN !match & Owner match: FAIL:OPERR

  Override with BLVP/BLP.

- AL-SL mounted: VSN match & Owner !match: FAIL:USERR

  Only VSN is checked currently. Owner may be checked later, based on some security-related configuration option in 'catadmin'.

OUTPUT

NL mount request

- NL mounted: PASS:BLP
- AL-SL mounted: !Expired: FAIL:OPERR

  Expiration Date can currently be bypassed via PROMAX_EXP_BYPASS. Should make this a 'catadmin' configuration.

- AL-SL mounted: Expired: PASS
- AL-SL mount request
- NL mounted: FAIL:OPERR

  Override with BLLP/BLP.

- AL-SL mounted: VSN !match: FAIL:OPERR

  Override with BLVP/BLP.

- AL-SL mounted: VSN match: !Expired: FAIL:OPERR

  Override with BLEP/BLP.

- AL-SL mounted: VSN match: Expired: PASS

  If the tape was pre-labeled by MVS, it will only have VOL1 and HDR1, but not have HDR2. MVS pre-labels HDR1 with 76 zeros so that the Expiration Date field of "000000" implies an ExpDate of year 2000. If ProMAX encounters an MVS pre-label tape (by checking for the absence of HDR2), it disables ExpDate checking and proceeds to output updated 3x80-byte labels.

ProClient Command Line Notes and Examples

ProClient supports four basic modes of operation, two input modes and two output modes: Input with Catalog, Input without Catalog, Output with Catalog, Output without Catalog.

When using ProClient's internal catalog interface, if OPNMODE is not specified, it will use the default of "DSOLD" on Input and "DSCREATE" on Output.

ProClient must always be aware of label type because of its internal volume access verification.

Recommended ProClient Command Line Arguments

Input with Catalog

Catalog opened from ProClient internal catalog interface.

DSN

Dataset name. Mandatory.

CATNAME

Catalog name. Optional if only one catalog is used.

DSVER

Dataset version number. Optional.

CONFIG

String returned from DEVICES file. Will contain physical device attributes and runtime behaviour switches.

Simplest case:

```
proclient $1 $2 CONFIG="config_opt" DSN="dsname"
```

With some debug turned on:

```
proclient $1 $2 DEBUG="debug_opts" CONFIG="config_opt"
    DSN="dsname"
```

Catalog opened from the client script:

```
proclient $1 $2 CONFIG="config_opt" VOL="volist"
```

Input without Catalog

DSN

Dataset name. Mandatory if label type not NL.

LABL

Desired label type. NL default, if not set.

VOL

Volume list. ***Mandatory***.

CONFIG

String returned from DEVICES file. CONFIG string will include NCAT or BCAT.

Simplest case:

```
proclient $1 $2 CONFIG="config_opt" VOL="volist"
```

or

```
proclient $1 $2 CONFIG=NCAT CONFIG="config_opt" VOL="volist"
```

or

```
proclient $1 $2 CONFIG="config_opt+NCAT" VOL="volist"
```

Labeled tapes:

```
proclient $1 $2 CONFIG="config_opt" LABL=label
    DSN="dsn_name" VOL="volist"
```

Output with Catalog

Catalog opened from ProClient internal catalog interface.

DSN

Dataset name. Mandatory if label type not NL.

LABL

Desired label type. Mandatory, must not be NL for catalogued output.

CATNAME

Catalog name. Optional if only one catalog is used.

OPNMODE

Catalog open mode. Optional DSCREATE or DSOVRWRT.

DPOOL

Device pool name. Mandatory.

DSVER

Dataset version number. Optional.

EXPD or RETD

Expiration date or retention period days. Both default to 5 days from current date.

CONFIG

String returned from DEVICES file. Will contain physical device attributes and runtime behaviour switches.

Simplest case:

proclient $1 $2 CONFIG="config_opt" DSN="dsn" DPOOL=devpool OPNMODE=omode LABL=label

Catalog opened from the client script:

```
proclient $1 $2 CONFIG="config_opt" DSN="dsn" LABL=label
    VOL="volist"
```

Output without Catalog

DSN

Dataset name. Mandatory.

LABL

Desired label type. NL default, if not set.

VOL

Volume list. Mandatory.

EXPD or RETD

Expiration date or retention period days. Both default to 5 days from current date.

Other Docs Known Problems

CONFIG

String returned from DEVICES file. Will contain physical device attributes and runtime behaviour switches.

Simplest case:

```
proclient $1 $2 CONFIG="config_opt+NCAT" VOL="volist"
```

Labeled tapes:

```
proclient $1 $2 CONFIG="config_opt+NCAT" DSN="dsname"
    LABL=label VOL="volist"
```

Tape Library Robotics Operations

The following information provides an overview to the setup, administration, and use of StorageTek robotic tape libraries. It is not meant to be a full installation or system administration guide, but serves as supplementary help to the StorageTek documentation. If you are the system administrator for the tape library, you should fully read the system manuals. If you are the ProMAX user, read this section to better understand the components of the robotics hardware and software, and to help isolate or diagnose problems that can occur.

StorageTek ACSLS

ACSLS (Automated Cartridge System Library Software) is the server software which controls the StorageTek robotic tape library. It can be set up to run on the same host as ProMAX or on a separate host.

Running a ProMAX tape job

When a ProMAX tape job executes, it starts ***$PROMAX_HOME/sys/exe/AAR_STKCMD*** on the tape client host for each instance of an input or output tape process in the flow. When the input/output device involved is not a robotic tape library such as a manual device, the proclient program controls the tape drive directly. However, if the input/output device involved is a StorageTek robotic tape library, the proclient program constructs ACSLS command files in the ProMAX flow directory which are remotely executed on the ACSLS host using the ***$REMOTE_TAPE_HOME/sys/exe/AAR_STKCMD*** shell script. Edit this script to set the PROMAX_STKEXEC variable to point to the correct $REMOTE_TAPE_HOME on the ACSLS controller. For example, if REMOTE_TAPE_HOME on ACSLS controller 'scorpion' were ~ACSSS/PMX, the script under the full ProMAX install would look like the following:

```
export PROMAX_STKEXEC=/export/home/ACSSS/PMX/sys/exe/AAR_STKREMOTE
        remote_host='scorpion'
```

Because of issues related to the different platforms supported for local and remote tape operations, the following items should also be set in the ***AAR_STKCMD*** file:

```
export PROMAX_SYS_EXE_REMSH_HOME=/usr/bin/rsh   or

export PROMAX_SYS_EXE_REMSH_HOME=/usr/bin/ssh

export PROMAX_HOME=<set this to the REMHOME path, if this is a
      remote tape operation>

exec=<local REMHOME path>/sys/bin/aar_stkcmd
```

The ***cmd_proc***, part of the ACSLS software provided by StorageTek, is the Command Processor screen interface to the ACS operations. There are six ACSLS cmd_proc commands which proclient executes on the ACSLS host while checking for drive/tape availability and tape mounts/dismounts:

- query drive all
- query volume all
- query drive
- query volume
- mount
- dismount

These will be described in more detail below along with other cmd_proc commands.

After installing ACSLS, you will also have set up an ACSSA (ACS System Administrator) user account; the home directory of this account is reachable as either user IDs ~acsss or ~acssa. On Solaris, going to the home directory of either ~acsss or ~acssa will put you in home directory /export/home/ACSSS/ where you can execute cmd_proc or any of the other utilities found in ~acsss/bin/. However, if you log in as user ~acssa, you will be automatically logged into the cmd_proc interface.

The ACS server software is started by executing the script file ~acsss/rc.acsss. Internally, this script executes ~acsss/.acsss_env to set necessary environment variables, starts up the database system, and then initiates the acsss_daemon. This script should be automatically executed by the /etc/rc* initialization scripts upon host boot-up, or you may manually execute ~acsss/rc.acsss after logging in as the ~acsss user ID. To check most of the ACSLS-related processes, you can `grep` on 'acs'. A typical listing of ACSLS processes might look like:

```
> ps -def | grep acs

USER  PID CPU MEM   SZ  RSS   START   COMMAND

acsss 403 0.0 1.6 3716 1924 11:31:07  acsss_daemon

acsss 412 0.0 1.0 1768 1220 11:31:07  acsel 403 50001 9 0

acsss 415 0.0 1.2 1932 1452 11:31:12  acssa 403 50004 23 0

acsss 416 0.0 1.9 4236 2408 11:31:15  acslm 403 50003 14 0 2 2

acsss 418 0.0 1.7 2772 2056 11:31:17  acslh 403 50002 13 0

acsss 423 0.0 2.0 4324 2460 11:31:21  acslock 403 50007 3 7 0

acsss 425 0.0 1.4 3780 1788 11:31:23  ss_ipc_clean 403 0 41 0

acsss 426 0.0 1.7 3972 2068 11:31:25  acssv 403 50008 39 0

acsss 447 0.0 2.0 4340 2444 11:31:28  acscm 403 50009 46 0

acsss 449 0.0 1.9 4120 2388 11:31:30  acsmt 403 50100 40 0

acsss 451 0.0 2.0 4184 2440 11:31:33  acsmt 403 50101 40 0

acsss 453 0.0 1.8 4032 2232 11:31:35  acsqy 403 50200 20 0

acsss 455 0.0 1.8 4032 2204 11:31:37  acsqy 403 50201 20 0

acsss 457 0.0 1.6 2400 1932 11:31:39  csi 403 0 6 0

acsss 498 0.0 5.0 8868 6260 11:34:30  st80VM
/export/home/ACSSS/bin/scsilh.im

acsss 502 0.0 1.0 1548 1184 11:34:55  /export/home/ACSSS/bin/scsiDP
/dev/mchanger0 0 false true
```

To bring up the cmd_proc interface, either log into the ~acssa user ID account, or execute execute ~acsss/bin/cmd_proc from any terminal type having a minimum size of 80 columns wide and 24 rows high. If your xterm/winterm/aixterm is too short or narrow, you will get an error message. Resize the terminal window to enlarge it will solve the problem.

When you are in the cmd_proc interface, you will see a split screen where the upper half outputs any ACSLS messages and the lower half is the command area where you enter ACSLS commands and which also outputs results from commands. For example, to query the ACSLS server:

```
ACSSA>  query server

09-15-98 09:39:05       Server Status

Identifier   State     Free Cell  Audit  Mount  Dismount  Enter  Eject

Count     C/P    C/P    C/P      C/P    C/P

run      0         0/0    0/0    0/0      0/0    0/0
```

If the "query server" command shows that ACSLS is offline (instead of "run"ing), you need to `vary` it online to a running state:

```
ACSSA> vary acs

ACS identifier(acs): 0

ACS identifier(acs):

State(diagnostic/offline/online): online
Vary: ACS   0 varied online.
```

To have the tape library fully functional, the LSMs and ports should all be online:

```
ACSSA>  q lsm

LSM identifier(acs,lsm): 0,0

LSM identifier(acs,lsm):

09-15-98 10:48:12              LSM Status

Identifier   State     Free Cell  Audit  Mount  Dismount  Enter  Eject

Count     C/P    C/P    C/P      C/P    C/P

0, 0      online      0         0/0    0/0    0/0      0/0    0/0
ACSSA>  q port

Port identifier(acs,port): 0,0

Port identifier(acs,port):

09-15-98 10:48:18              Port Status

Identifier   State

0, 0      online
```

Note from the above "query" example that most ACSLS commands accept an abbreviated form.

If a port is offline, you need to vary it online before you can vary its ACS online.

```
ACSSA>  vary
Type(acs/cap/drive/lsm/port): port
Port identifier(acs,port): 0,0
Port identifier(acs,port):
State(diagnostic/offline/online): online
Vary: port   0, 0 varied online.
```

Query all the drives to make sure they are available:

```
ACSSA>  q drive all
09-15-98 10:51:11          Drive Status
Identifier   State      Status    Volume    Type
0, 0, 2, 0 online     available          DLT4000
0, 0, 2, 1 online     available          DLT4000
0, 0, 2, 2 online     available          DLT4000
0, 0, 2, 3 online     available          DLT4000
0, 0, 2, 4 online     available          DLT4000
0, 0, 2, 5 online     available          DLT4000
```

You can also query the status of individual drives:

```
ACSSA>  q drive 0,0,2,2
09-15-98 10:55:27          Drive Status
Identifier   State      Status    Volume    Type
0, 0, 2, 2 online     in use    000034    DLT4000
```

Query all the volumes and check to see if the listing is what you expect:

```
ACSSA>  query volume all
09-15-98 10:53:35          Volume Status
Identifier  Status       Current Location   Type
000030      home           0, 0, 1, 7, 0    DLTIV
000031      home           0, 0, 0, 1, 3    DLTIV
000032      home           0, 0, 0, 2, 3    DLTIV
000033      home           0, 0, 0, 3, 3    DLTIV
000034      in drive     0, 0, 0, 4, 3    DLTIV
```

This interactive dipslay may quickly scroll off the listing of all volumes. You can save the full listing to a file by executing cmd_proc from the command line and using the shell's input redirection facility:

```
> ~acsss/bin/cmd_proc <<EOF > /tmp/vols

query volume all

EOF
```

The above example will save the output volume listing that "query volume all" output to a file named ***vols*** in directory /tmp.

To further verify the tape library set up, issue a mount for a volume:

```
ACSSA>  mount 000001

Drive identifier(acs,lsm,panel,drive): 0,0,2,0

Option(readonly/bypass/):

Mount: 000001 mounted on   0, 0, 2, 0


ACSSA> q drive 0,0,2,0

09-15-98 11:02:53          Drive Status

Identifier   State      Status    Volume    Type

0, 0, 2, 0 online      in use    000001    DLT4000


ACSSA> q vol 000001

09-15-98 11:02:56          Volume Status

Identifier  Status       Current Location   Type

000001      in drive         0, 0, 2, 0     DLTIV
```

You can also enter the 'mount' command as a single line:

```
ACSSA> mount 000001 0,0,2,0
```

to bypass the separate prompts. Once the tape is mounted, from the host where the robotic drive is physically attached, issue some trivial UNIX commands like `mt` to status the drive and `dd` to dump the label:

Other Docs

Known Problems

```
% mt -f /dev/rmt/tps6d1nsv status
Controller: SCSI
Device: Quantum: DLT4000       CD3CQ<
Status: 0x20262
Drive type: DLT
Media : READY, writable, at BOT

% dd if=/dev/rmt/tps6d1nsv bs=80 count=3 conv=ascii
VOL1000001                       JEFFW
HDR1DECONAMPSHT.01-08      1  1     98194001125     0
HDR2U   0   00
3+0 records in
3+0 records out
```

The point of this exercise is to make sure that in the ***$PROMAX_HOME/etc/rdb/config_file*** (or ***$PROMAX_ETC_RDB_HOME/config_file***), device entries, the device: stanza defining each robotics drive, match their DEVINFO=ACS,LSM,Panel,Unit parameter.

To dismount the volume:

```
ACSSA> dismount 000001
Drive identifier(acs,lsm,panel,drive): 0,0,2,0
Option(force/): force
Dismount: Forced dismount of 000001 from  0, 0, 2, 0

ACSSA> q drive 0,0,2,0
09-15-98 11:13:49         Drive Status
Identifier  State     Status   Volume   Type
0, 0, 2, 0 online     available         DLT4000

ACSSA> q vol 000001
09-15-98 11:13:57         Volume Status
Identifier  Status      Current Location   Type
000001     home          0, 0, 0, 1, 3    DLTIV
```

Lastly, exit the cmd_proc interface with 'logoff':

```
ACSSA> logoff
```

If you logged in as the ~acssa user, logging off will also log you off the ~acssa user account. If you executed cmd_proc as user ~acsss or another user, logging off will just return you back to the shell prompt.

As the ProMAX system administrator, another cmd_proc command you may sometimes need is 'audit':

```
ACSSA>  audit

CAP identifier(acs,lsm,cap): 0,0,0

Type(acs/lsm/panel/server/subpanel): acs

ACS identifier(acs): 0

ACS identifier(acs):

Audit: Audit completed, Success.

Audit: Audit of ACS   0, Valid.
```

or, again, to audit the entire server:

```
ACSSA>  audit * server

Audit: Audit completed, CAP   0, *,* in use.

Audit: Audit of storage server, CAP   0, *,* in use.
```

As seen by the audit options above, you can audit the entire library or just components of it. You typically would want to perform an ACSLS audit if you have:

- Populated a new library or removed/added volumes to the LSM units.
- Encountered a ProMAX error, in which a job could not find a request VSN.

The audit process will reconcile the silo contents with the ACSLS database.

StorageTek Hardware

If you are having problems with a newly installed tape library, some of the configuration issues to check are:

- Verify proper SCSI cabling. Single-ended SCSI cabling should not exceed 12 feet between the ACSLS host and the tape library; differential cabling should not exceed 25 meters.

- Connectors should all be firmed plugged in and cables should be terminated.
- The correct SCSI addresses should also be set for each drive within the silo.
- The control panels of each drive typically will have a 'normal' set of lit and unlit LEDs. If this light pattern is observed to be abnormal, that should either indicate a faulty drive, drives which were physically switched offline, or drives/LSMs which were varied offline by ACSLS.

Other Docs Known Problems

TTS Files

The following binaries, scripts, and configuration files in the ProMAX directory structure are used by TTS (with the directory hierarchy being represented by the level of text indentation):

$PROMAX_HOME

TopCat/

Home directory for the TopCat tape catalog.The directory location can be explicitly set with environment variable PROMAX_TOPCAT_HOME; if this is not set, the default TopCat directory location is $PROMAX_HOME/TopCat/. Catalog files in this directory are accessed and modified by the $PROMAX_HOME/sys/bin/topcat program. When you set up the TopCat catalog, you will always have one, and only one, "primary" catalog; you may optionally set up one or more secondary catalogs.

Each primary or secondary catalog is associated with a set of the following three files:

- catadmin configuration file
- *.catindx catalog index file
- *.catdata catalog data file.

catadmin

This catalog administration file contains catalog configuration information set up by the ProMAX system administrator. There must be one catadmin file (reserved filename) which defines the configuration of the primary catalog. The catalog administration files for all other secondary TopCat catalogs canhave arbitrary filenames.

*.catindx

The catalog index file. Contains VSN range information for quick VSN information retrieval. Each primary and secondary catalog has one of these files. For example, if the catalog filename you specify is geoproc, you will have a geoproc.catindx file in the TopCat directory.

*.catdata

The catalog data file. Contains catalogattribute data for all VSNs. Each primary and secondary catalog has oneof these files. For example, if the catalog filename you specify is geoproc, you will have a geoproc.catdata file in the TopCat directory.

etc/

As with the rest of ProMAX, this directory contains configuration file information.

device.tts

File which describes devices for use in tape jobs. It maps logical device names to physical devices or to device pools defined in PMX, as well as specifying any additional device attributes.

config_file

Four "config_file" entries are relevant to TTS operation:

- device file
- packet ladeing
- flow lisp
- Super Exec

rdb/

This is the home directory for the PMX Resource Manager database, containing all device/host/pool definitions. Its directory location may be explicitly set using environment variable PROMAX_ETC_RDB_HOME. It is accessed and modified by the six PMX programs in $PROMAX_HOME/sys/bin/:

- pmxDev
- pmxInit
- pmxRls
- pmxRsv
- pmxSet

- pmxMov

config_file

The contents of this config_file are devices, device pools, and device and/or device pool attributes used by ProClient. The script, $PROMAX_HOME/etc/pmxres_init, which in turn calls the executable, PROMAX_HOME/sys/bin/pmxInit, uses this file to construct the PMX Resource Manger database.

dynamic

One of the two Resource Database files created by the pmxInit program. This 'dynamic' file can change in size over time. It tracks all active and queued requests as they are acted upon by the programs pmxRsv (which reserves devices for use) and pmxRls (which releases devices after use).

static

One of the two Resource Database files created by the pmxInit program. This 'static' file does not change in size, but is set during the initial pmxInit invocation. It stores all device/host/pool definitions.

pmxres_init (formerly pmxinit)

Optional shell script wrapper for the pmxInit command to initialize a new Resource Database. Uses config_file located in this directory.

port/

bin/

ConsoleMonitor (New)

The shell script which starts the Console Monitor Operator Console Viewer Java UI. It accesses the file PROMAX_HOME/etc/java.config.

PromaxCatalogs

Shell script which returns the count or list of available TopCat catalogs (by executing topcat 'CATINFO[INFCATNAME]'). This script is called from the tape menus and, if more than one catalog is defined, the output is presented as pop-up selection list in the menu.

PromaxDatasets

Shell script which returns a list of TopCat catalogued datasets by executing a topcat catalog query. This script is called from the tape menus and the output is presented as a pop-up selection list in the menu.

PromaxHelp

Although not directly related to TTS operation, this script is executed by the ProMAX UI to display .lok, .help, or .html help documentation via FrameViewer, text window, or Web browser, respectively.

PromaxOpconFind

Called by the operator console to try and find if a request is running.

PromaxOpconKill

Called by the operator console to kill a request.

TAPE_NOTIFICATION

This Tape Notification Java clientshell script is executed by proclient to pass Java operator notification messages to the OpCon Operator Console UI.

TAPE_REQUEST

This Tape Request Java client shell script is executed by proclient to pass Java tape mount request messages to the OpCon OperatorConsole UI.

opcon

The shell script which starts the OpCon Operator Console Java UI. It accesses the file PROMAX_HOME/etc/java.config.

prompt

Used by other parts of ProMAX also, this TAM (TCL Motif) shell script (which needs to be executed as $PROMAX_HOME/sys/bin/tamsh) is used by the

$PROMAX_HOME/sys/exe/tapeDeleteExit shell script to pop up a confirmation window when a tape dataset is deleted from the ProMAX User Interface, prompting the user to either delete or retain the dataset's entry in the tape catalog.

releaseDevice

Shell script called by the superExec when a tape process completes. It executes pmxRls on a PMX Resource Manager tape request that was previously reserved.

reserveDevice

Shell script called by the superExec when a tape process starts up. It executes pmxRsv on a PMX Resource Manager tape request in order to wait for an available device, reserves it, and then creates the "DEVICES" runtime directory file.

showcat

The shell script which starts the ShowCat Tape Catalog Java UI. It accesses the file $PROMAX_HOME/etc/java.config.

stkrsh_cmdproc

Shell script which is executed by proclient. It interact with StorageTek robotics by building and executing ACSLS `cmd_proc` commands on the ACSLS host.

tapeCloseExit

Shell script which can get called by specifying a "c" for a dataset in the emacs window called by tapeDeleteExit; the "c" calls for the dataset to be "closed" if it is open.

tapeDeleteExit (New)

Shell script called by the ProMAX UI when the user deletes an area or line. An emacs window appears from which the user can specify which, if any, of the datasets in the area or line should be deleted from the catalog. Don't confuse this script with $PROMAX_HOME/sys/exe/tapeDeleteExit.

tapediag_diagnose

Shell script wrapper for $PROMAX_HOME/sys/exe/tapediag which runs diagnostics on a device.

tapediag_maxblock

Shell script wrapper for $PROMAX_HOME/sys/exe/tapediag which determines the maximum blocksize of a device.

tapediag_prelabel

Shell script wrapper for $PROMAX_HOME/sys/exe/tapediag which writes an AL or SL to a tape. The ProClient SMNT option requires labelled scratch tapes.

tapedump

Shell script wrapper for $PROMAX_HOME/sys/exe/tapediag which dumps the contents of a tape, using commands from an interactive terminal session.

java/ (formerly $PROMAX_HOME/java)

Directory containing all ProMAX and third party Java libraries.

ENV.java

Shell script which is sourced by all the ProMAX Java shell script wrappers in $PROMAX_HOME/port/bin/: opcon, showcat, TAPE_REQUEST, TAPE_NOTIFICATION and ConsoleMonitor. It sets some environment variables and parameters which are common to all the Java programs.

misc/

tape_err.msg

Error message file used by the proclient program.

topcat_err.msg

Error message file used by the topcat program.

sys/

bin/

multifileselect

Multiple tape-on-disk file browser optionally used by ProMAX tape input menus as an alternative to typing in a list of filenames. To use it, set the environment variable <*host*>_PROMAX_SYS_BIN_MULTIFILESELECT_HOME, where <*host*> is the name of the local or remote host on which the desired files reside, to the fully-qualified pathname of the executable multifileselect on that host. (This is a general ProMAX mechanism for tailoring ProMAX search paths.) As an example, if the files you want to select are on machine ***alpha*** and your remote tape installation on that machine is under the directory ***/apps/Landmark/RemoteTape***, then you would set the variable

ALPHA_PROMAX_SYS_BIN_MULTIFILESELECT_HOME

to the value

/apps/Landmark/RemoteTape/sys/bin/multifileselect

in your ProMAX environment. If you need to specify a fully-qualified hostname, e.g. ***alpha.lgc.com***, then replace *ALPHA* with *ALPHA_DOT_LGC_DOT_COM* in the above.

pmxDev

pmxInit

pmxRls

pmxRsv

pmxSet

pmxMov

The six programs comprising the PMX Resource Manager. pmxInit initializes the PMX resource database (rdb) files, and should be run by the ProMAX system administrator. pmxDev displays the resource database and can be run by administrators or users from either the command line or the OpCon Operator Console. pmxRsv and pmxRls reserve and release devices, respectively, and are normally called from within ProMAX I/O jobs. pmxSet sets various database maintenance options, and should be run by the ProMAX system administrator or computer room operator from either the command line or the OpCon Operator Console. pmxMov enables the operator to move a mount to a different device.

tapediag

Tape diagnostics tool to assist users in diagnosing tape problems, determining tape performance issues, and for prelabeling tapes when using the proclient SMNT scratch mount mode. Usually called by means of the various tapediag shell script wrappers in $PROMAX_HOME/sys/bin/ which perform specific functions. (see above)

topcat

The TopCat Tape Catalog program. It accesses the catadmin file in $PROMAX_TOPCAT_HOME and updates the *.catindx/*.catdata catalog files.

exe/

ProClient

The client shell script for the TTS tape management system. It runs a $PROMAX_HOME/sys/exe/proclient binary executable on the client host and handles remote tape drive access and interaction with the PMX Resource Manager programs.

proclient

The TTS tape management system client program which is called by the ProClient script.

tapeDeleteExit

Shell script called by the ProMAX UI which uses $PROMAX_HOME/port/bin/prompt to pop up a confirmation window when a tape dataset is deleted, prompting the user to either delete or retain the dataset's entry in the tape catalog.

scratch/

OperatorConsole.V2.state

This file is not directly used by the PMX Resource Manager, but is modified by the OpCon Operator Console UI. It stores state information about the OpCon (e.g. current and previous operator notification messages).

Trouble Shooting

Confusion can sometimes arise from the setting of environment variables in the .cshrc file of users having C-shell as their login shell. This can be even more problematic if the site uses different ProMAX versions. The .cshrc should only have a few UNIX environment variables set, such as DISPLAY or PATH. Environment variables such as PROMAX_HOME, LIBPATH, LD_LIBRARY_PATH, etc, are best set in the .login file or customized in start-up scripts for launching ProMAX.

In this chapter

Tape Catalog

How do I protect my tape system from unauthorized use/abuse or accidental deletions?

There are three areas of the tape system where you may tighten the security and permissions of users' ability to modify, access, or delete files:

- TOPCAT_HOME TopCat Catalog directory
- OpCon Operator Console and PMX Resource Manager database
- /dev/.. device name files.

In these three areas, each of which may be restricted independent of the others, permissions may be restricted at the user level by making a binary program set-uid to a restricted user ID and then restricting read-write access on all the program's datafiles to that user ID. In the three areas described below, you may prefer to not use a user set-uid approach, but to restrict all described files and binaries by only allowing user and group level access to files and binaries, and preventing others from accessing them (e.g. instead of setting 'chmod 4755' on a user set-uid binary, you would set 'chmod 750' on the program and the 'chmod 660' on the program's data files.

But be aware that controlling access to binaries and files on a group-level basis is less secure than locking everything down to one user.

TopCat security

To restrict access to the TOPCAT_HOME TopCat Catalog directory via a set-uid approach, the sys admin should set up a restricted tape user ID account, tighten permissions on the critical data files within the TopCat directory, and make the 'topcat' program binary set-uid to that user. For example, if the restricted user ID was "promax_tape", the following commands will only allow users to modify the TopCat Catalog files by running the 'topcat' program (i.e. from within ProMAX or the ShowCat Catalog UI...

```
chown -R promax_tape $TOPCAT_HOME
chmod 700 $TOPCAT_HOME
```

```
cd $TOPCAT_HOME
chmod 600       catadmin* *.catdata *.catindx
chown promax_tape $PROMAX_HOME/sys/bin/topcat
chmod 4755       $PROMAX_HOME/sys/bin/topcat
```

(to allow anyone to run the ‘topcat‘ program)

or

```
chmod 4750       $PROMAX_HOME/sys/bin/topcat
```

(to only allow users within a group to run the ‘topcat‘ program)

In addition, it is recommended that you back up the TopCat directory either to nightly tape backup and/or to a mirrored directory on a different disk in case of accidental deletion or disk failure.

OpCon Operator Console and PMX Resource Manager database restriction

To restrict who can use the Operator Console and who can access the PMX Resource Manager database via a set-uid approach, the sys admin should set up a restricted tape user ID account (may be the same account used to own the TopCat files described above), tighten permissions on the critical data files within the PMX directory (PROMAX_ETC_RDB_HOME), and make the necessary pmx programs set-uid to that user. For example, if the restricted Operator Console ID was "oper", the following commands will only allow users to modify the PMX resource database files by running the PROMAX_HOME/sys/bin/pmx... programs.

```
cd $PROMAX_HOME/etc/rdb  (or wherever
    $PROMAX_ETC_RDB_HOME is)
chown -R oper .
chmod 600 dynamic static config_file
chown oper $PROMAX_HOME/sys/bin/pmx*
chmod 4755 $PROMAX_HOME/sys/bin/pmx*
```

(to allow anyone to run ProMAX tape jobs interacting with OpCon)

or

```
chmod 4750 $PROMAX_HOME/sys/bin/pmx*
```

(to only allow users within a group to run tape jobs with the OpCon)

```
chown oper $PROMAX_HOME/port/bin/opcon
chmod 700 $PROMAX_HOME/etc/rdb
chmod 700 $PROMAX_HOME/port/bin/opcon
```

(to allow only one user to run the ‘opcon‘ program)

or

```
chmod 770 $PROMAX_HOME/etc/rdb
chmod 750 $PROMAX_HOME/port/bin/opcon
```

(to allow only users within a group to run the ‘opcon‘ program)

Restricting /dev/.. device access

For the ProMAX PMX Resource Manager routines to keep accurate track of which drives are reserved or available, users should not be able to arbitrarily loaded tapes into those drives for non-ProMAX jobs (e.g. performing ‘tar‘ tape copies).

To restrict who can access specific drives, either for ProMAX jobs or to keep others from using the drives for non-ProMAX jobs, the sys admin should set up a restricted tape user ID account, change ownership and permissions on the /dev/... device files, and make the $PROMAX_HOME/sys/exe/proclient binary set-uid to that user. For example, if the restricted user ID was "promax_tape", the following commands will allow users to only run ProMAX tape jobs on specific /dev/... devices (i.e. from within ProMAX):

```
chown promax_tape /dev/rmt/tps... /dev/rmt..
chmod 600       /dev/rmt/tps... /dev/rmt..
```

(change the ownership and permission for all devices on all hosts which are to be under exclusive ProMAX control)

```
chown promax_tape $PROMAX_HOME/sys/exe/proclient
chmod 4755      $PROMAX_HOME/sys/exe/proclient
```

(to allow anyone to run the ‘proclient‘ program)

or

```
chmod 4750      $PROMAX_HOME/sys/bin/proclient
```

(to only allow users within a group to run the ‘proclient‘ program)

My POPULATE commands are not initializing volumes in the Tape Catalog.

Check to make sure that there are not VSNs in POPULATE’s "POPVRNG" which coincide with VSNs currently in the Catalog. The "POPMPOOL" must also match a valid "POOLNAME:" defined in a CATPOOL stanza of the TopCat "catadmin" file.

My Tape Catalog jobs abort with core file dumps or memory faults, or ShowCat does not show any Catalog results.

While there may be other reasons this is happening, one thing to check is to make sure that your user name and group name are being resolved from NIS and the /etc/passwd and /etc/group files. This sometimes is not fully set up on new machines. If you enter the UNIX command ‘/bin/id‘, you should see something like:

```
uid=5321(prouser) gid=100(seisproc)
```

where both the uid/gid User and Group IDs are numeric values followed by their actual names in parentheses. If the "uid=" value is just numeric without any name association, you should add something like:

```
+::0:0:::
```

to your /etc/passwd file, which tells the system to incorporate entries from the NIS. If the "gid=" value is just numeric without any name association, you should add:

```
+:
```

to your /etc/group file, which tells the system to incorporate entries from the NIS group database. The ‘topcat‘ program compares your user and group names for access control functions such as the CATADMIN entries in the ’catadmin’ file, and ‘topcat‘ will abort if the user and group names are not available.

Other Docs Known Problems

ShowCat

My secondary Catalogs do not show up in the 'Select...' option of the [Catalog] menu; only the primary Catalog is listed.

The "catadmin" file in the TopCat directory is a reserved filename which defines the main Catalog. To add secondary catalogs, this "catadmin" file must have CATINCLUDE stanzas at the bottom of the file, after the CATALOG/CATEND stanza pair. Each CATINCLUDE must point to a valid "catadmin" of each secondary catalog (e.g. CATINCLUDE:catadmin.tcat).

I execute an 'Edit Query', but nothing shows up.

This may be due to a number of reasons: 1) there are actually no VSNs in the Catalog to match your query; 2) if you are querying a secondary Catalog, make sure that you 'Select' it first from the Catalog menu of ShowCat; 3) you have not sufficiently parameterized the [Search] parameters of the Catalog Query; 4) you have not selected any fields to display in the [Display] of the Catalog Query; 5) there may be some bugs in ShowCat which prevent the query from either executing or being displayed.

I cannot directly edit Catalog fields from ShowCat.

In order to directly edit ShowCat fields, your user ID must be included in the TOPCAT_HOME/catadmin file's "CATADMIN:" stanza. Furthermore, if your user ID is not listed in the "CATADMIN:" stanza, and some Catalog fields have been defined as "READ" read-only status in the 'CATRESTRICT' stanza of "catadmin", you will not be able to write to (modify) these fields.

I was editing Catalog fields from ShowCat, but made some major editing errors.

All edits done from ShowCat instantaneously update the TopCat catalog. But using the Showcat [Edit] menu's 'Undo' option, you can keep undoing each of your previous edits until you've reached the previously correct state again. Each time you click 'Undo', it will tell you what the last edit was

Other Docs Known Problems

that will be undone. This is one advantage of using ShowCat to perform Catalog edits, as opposed to running topcat

VEDITs from the command line. If you have a lot of tapes in the Catalog, the ShowCat 'Undo' function may currently take awhile to complete while it undoes the previous edit. Near the top of the ShowCat UI, there is the "Cell edits apply to:" option. This defaults to 'Edited row only', but during the course of editing, if you set it to 'All rows' for one edit, you should take note of whether that is what you still want to do for the next edit.

All of the Catalog fields are not being updated for my datasets' VSNs.

This is a known problem. TopCat currently only updates a portion of the 85 possible fields.

OpCon

My OpCon does not get any mount requests or notification messages, or has stopped getting them.

To check if this problem exists from within ProMAX, try communicating to the OpCon by sending messages to it from the command line. To send a tape mount request to the top [Mount Request] pane of the OpCon, enter something like:

```
PROMAX_HOME/port/bin/TAPE_REQUEST host_name mount1 2 3 4
```

where "host_name" is the hostname where the ‘rmiregistry‘ and ‘opcon‘ are running.

To send a tape notification message to the [Operator Notification] pane of the OpCon, enter:

```
PROMAX_HOME/port/bin/TAPE_NOTIFICATION host_name test
    "hello world"
```

Both the ‘TAPE_REQUEST‘ and ‘TAPE_NOTIFICATION‘ command line execution should send something to the OpCon.

The "Device Manager" pane is not working or is showing wrong devices/pools when I click [Inquire].

Check the: export **PROMAX_ETC_RDB_HOME=/network/galaxy/data7/PetroBras/70/etc/rdb** in the PROMAX_HOME/port/bin/opcon script. The information from all three tabs of the "Device Manager" pane should also match a command line execution of PROMAX/sys/bin/pmxDev.

Other Docs

Known Problems

PMX Resource Manager

My ProMAX job is not getting the correct device or device pool.

Check the PROMAX_HOME/etc/rdb/config_file settings, syntax, and UNIX device names. The device pool name (the second entry in the device.txt file) must match one of the "pool:" stanzas in the rdb config_file, and also must match the " DEVPOOL: " entry in one of the CATPOOL stanzas defined in the TOPCAT_HOME/catadmin file. The DEVPOOL which is used by the ProMAX job then maps to a media pool as defined by the CATPOOL's " POOLNAME: " value. And this POOLNAME must match the MPOOLNAM of VSNs in the TopCat Catalog.

Other Docs Known Problems

ProMAX Tape I/O

My job aborts during initialization phase.

This could be due to a number of reasons... First of all, some messages like:

```
server: accept(): Timed Out.

... or ...

ERROR: UNABLE TO CONNECT TO REMOTE TAPE SYSTEM.

Possible start-up script error client program: ...ProClient was not found

... or ...

Unix error - no more processes
```

tend to be spurious and may occur with various aborts. The actual cause of the error may lie elsewhere in the job.output. Other causes of initialization phase aborts are listed as other topics below.

If this is the first time running tape jobs, one of the most common reasons for aborts is script variables which have not been correctly updated. Refer to the Configure all TTS-related scripts. section for another review.

During initialization phase, the script

```
PROMAX_HOME/sys/exe/ProClient
```

is executed. This script has two levels of debug options to help determine what is going on. You may uncomment the line within the ProClient script:

```
export DEBUG_PROCLIENTSCRIPT=yes && set -x
```

to turn on debugging within the script itself. This will output all script executions to the job.output and may show if a problem is occurring during script execution. You can also uncomment the ProClient line:

```
[ -z "$prodebug" ] &&
   prodebug="DEBUG=DECO:SVOL:CONS:CAT:CMDL:STK:VERI:I
   O"
```

to turn on full ‘proclient‘ debugging. The PROMAX_HOME/sys/exe/proclient binary is executed near the bottom of the ‘ProClient‘ script. The "prodebug" options will not print debug messages to the job.output until the

ProClient script has executed the ‘proclient‘ binary. The "prodebug" options are a colon-delimited list of debug options which print out diagnostics relating to different aspects of ‘proclient‘ operation; they are described in the ProClient Command Guide . But if you are not sure just what is going wrong, you may turn on all debug options either by setting "prodebug" to ’ALL’, or listing each option delimited by a colon.

My job aborts during initialization phase with message: catEnv.n: not found

If you see a message like:

```
PROMAX_HOME/sys/exe/ProClient[75]:
/promax/data/home/area_name/line_name/flow_name/nnnn/catEnv.1:
not found
```

you probably have not added the necessary entries to PROMAX_HOME/etc/config_file.

My job aborts during initialization phase with message: Unable to write lock, check executable and catalog permissions

If you see a message like:

```
Unable to write lock, check executable and catalog permissions
```

check the TOPCAT_HOME directory to make sure that 1) the directory has proper read, write, and execute permissions; 2) the *.catdata and *.catindx files within the directory have proper write permissions; and 3) no stale *.lock files are left in the directory (stale means that the lock files should normally not linger for more than a few seconds).

My job aborts during initialization phase with message: Error 11 occurred opening device "device_pool_name"

If you see a message like:

```
Error 11 occurred opening device tape_8mm

...

Error opening tape dataset
```

check the PROMAX_HOME/etc/rdb/config_file settings, syntax, and UNIX device names. The device pool name (the second entry in the device.txt file) must match one of the "pool:" stanzas in the rdb config_file, and also must match the " DEVPOOL: " entry in one of the CATPOOL stanzas defined in the TOPCAT_HOME/catadmin file. The DEVPOOL which is used by the ProMAX job then maps to a media pool as defined by the CATPOOL's " POOLNAME: " value. And this POOLNAME must match the MPOOLNAM of VSNs in the TopCat Catalog

My job aborts during initialization phase with message: Permission denied

If you see a message like:

```
Permission denied
```

this is usually a error message coming from UNIX. Three areas to check UNIX permissions for proper tape I/O operation are: 1) the TopCat directory; 2) the PMX Resource Database directory; and 3) the ACSLS server if STK robotics are involved.

The TopCat directory, defined by TOPCAT_HOME or defaulted to " PROMAX_HOME/TopCat ", should have read-write-execute permissions for users. Users should be able to read from and write to the *.catdata Catalog files within this directory.

The PMX Resource Database directory, defined by PROMAX_ETC_RDB_HOME or defaulted to " PROMAX_HOME/etc/rdb ", should have read-write-execute permissions for users. Users should be able to read from and write to the "static", "dynamic", and "*.state" files within this directory.

If you are using STK robotics, the ProMAX tape systems issues ACSLS commands via ‘rsh‘ to the ACSLS host. Users must be able to freely ‘rsh‘ over to the ACSLS host; for example, a simple:

```
rsh acsls_host date
```

from the ProMAX host over to the ACSLS host should promptly return the date. If you get a "Permission denied" message with this or if you get prompted for a password, then you may need to add user id and authorization information to the .netrc and .rhosts files.

My job hangs during initialization phase for no apparent reason.

There may be various reasons why your job does not proceed beyond initialization phase. Checking the possible causes listed below may help you resolve the problem, or can provide better information for ProMAX support personnel to find the cause.

One of the more common reasons for the job hang is that a TopCat lock file has been left in the TOPCAT_HOME directory. Do an ‘ls -l‘ in the TOPCAT_HOME directory, and look for any "*.lock" files older than a few minutes. These leftover lock files may be caused by a number of reasons, but old lock files may usually be removed. Also check the ‘ps‘ listings for any leftover ‘topcat‘ processes that may need to be cleaned up.

Check the ‘ps‘ listings to see if there are any ‘pmxRsv‘ processes running. If all devices in the device pool you are requesting in your job are in use, the job will hang and not start until a device becomes available. This is normal. However, if the OpCon or a ‘pmxDev‘ output shows that there are available devices, there should be no lingering ‘pmxRsv‘ processes trying to reserve for that device pool.

Another cause for hangs may be due to network problems. If any of your TOPCAT_HOME, PROMAX_ETC_RDB_HOME, ROMAX_HOME, PROMAX_SCRATCH_HOME, or PROMAX_DATA_HOME directories are NFS-mounted, check for the response and accessibility of these directories.

If you are using STK robotics, check the accessibility of the ACSLS host by executing ‘rsh acsls_host date‘ from the ProMAX host.

Try killing the job from the ProMAX UI and re-running the job. But before you re-execute the job again, check to make sure the previously mentioned problems are not an issue; i.e. remove TopCat lock file if present and kill all leftover ’topcat’ or ’pmxRsv’ processes. Then uncomment the two debug options in the PROMAX_HOME/sys/exe/ProClient script and re-run the job. The output from the ProClient script debug messages may help you or ProMAX support personnel determine if the job is hanging within the script itself, or after the ‘proclient‘ binary program has started.

My "system generated" or "user tape filename" input dataset is not listed from the ProMAX UI, even though I can see it in the Catalog.

Either 1) the device pool which is selected is wrong for that input dataset; or 2) for sites with secondary catalogs, the 'Available Tape Catalogs' menu parameter is set wrong; or 3) the dataset that you output is of a different format than the menu you are using (e.g. native ProMAX, SEG-Y, Western4, etc).

The "user tape filename" option of input menus executes ‘ PROMAX_HOME/port/bin/PromaxDatasets ‘ to list out all catalogued datasets in pop-up list_select window that match three criteria: 1) the device pool selected by the user (which it uses to derive a MPOOLNAM media pool), 2) the catalog selected by the user (in the case of multiple available catalogs), and 3) the format of the data, asderived by the menu you are parameterizing.

I have secondary catalogs defined, but I do not seem to be able to access them from the ProMAX menus.

For sites with secondary catalogs (likely reasons would be to have read-only catalogs of old tcat catalogs or catalogs of field data), the TOPCAT_HOME/catadmin file must have at the bottom of the file (after the last " CATEND " line) a:

```
CATINCLUDE:secondary_catalog_catadmin_filename
```

entry for each secondary catalog to be included. The ProMAX tape menus then execute ‘PROMAX_HOME/port/bin/PromaxCatalogs count ‘ to generate a list of all defined catalogs. If the number of catalogs is greater than 1 (1 being the default main catalog), then the ProMAX menus will display a 'Available Tape Catalogs' menu parameter for selecting alternate catalogs. If you do not see this parameter, try executing:

```
topcat 'CATINFO[INFCATNAME]'
```

on the command line to see if you get more than one catalog listing. If you do not, then you have a problem with the "catadmin" file of either your primary catalog or CATINCLUDEd secondary catalog.

Robotics

I get a mount error message on the OpCon: "StorageTek Robotic Interface ERROR: Illegal or unrecognised robotic device specification"

If you get a mount error message on the OpCon and clicking MB2 on the mount message shows a message like:

```
Illegal or unrecognised robotic device specification.Verify configuration
before retry.
```

Check the PROMAX_HOME/etc/rdb/config_file (or PROMAX_ETC_RDB_HOME/config_file) entries to make sure that each defined robotics drive’s " device: " entry matches its ’ DEVINFO = ACS,LSM,Panel,Unit ’ parameter.

I get a mount error message on the OpCon: "StorageTek Robotic Interface ERROR: unable to find and mount the requested VSN"

If you get a mount error message on the OpCon and clicking MB2 on the mount message shows a message like:

```
ACSSA control was unable to find and mount the requested VSN. You
may correct this problem and retry.
```

ProMAX issued this error due to an error returned by ACSLS in trying to query a volume. To ascertain this, look in the job.output to find what VSN is being mounted, log onto the ACSLS host, bring up the ‘ ~acsss/bin/cmd_proc ‘ interface and issue the query command on that same VSN. If you get a "Volume identifier ...... not found" error:

```
ACSSA> query volume 734041

09-04-98 11:06:34             Volume Status

Identifier  Status        Current Location       Type

Volume identifier 734041 not found.
```

either the VSN being requested is not in the robot silo, or the ACSLS database does not match the volumes actually located inside the silo.

The latter situation may occur if volumes have been added, switched, or removed. You can reconcile the silo contents with the ACSLS database by issueing the ‘audit‘ command from the ‘cmd_proc‘ interface.

I get a mount error message on the OpCon: "StorageTek Robotic Interface ERROR: command has failed"

If you get a mount error message on the OpCon and clicking MB2 on the mount message shows a message like:

```
StorageTek Robotic Interface ERROR

One of these four robot commands has failed.

QUERY DRIVE  x,x,x,x

QUERY VOLUME xxxxxx

MOUNT VOLUME xxxxxx x,x,x,x

DISMOUNT xxxxxx x,x,x,x force
```

You may re-execute the failed STK command at the cmd_proc interface to determine its cause, fix, and retry.

ProMAX issued this error due to a problem interacting with the ACSLS host (from within the ‘proclient‘ program). To troubleshoot this, you can trace the command sequence manually to find out where the robotic operation is failing.

During STK robotics operation, ‘proclient‘ builds a series of ACSLS commands in files found in PROMAX_DATA_HOME/area/line/flow/NNNN.PT_STKCMD. NNN (where "NNN" are numbers). These ACSLS command files are then executed on the ACSLS host using the script PROMAX_HOME/sys/bin/stkrsh_cmdproc. For starters, see if you can execute ‘ rsh acsls_host date ‘ from the ProMAX host over to the ACSLS host. If you get a " Permission denied " error or are prompted for a password, you have to add user id and authorization information to the user’s .netrc and .rhosts files.

Go into the PROMAX_HOME/sys/bin/stkrsh_cmdproc script and find the line:

```
rsh acsls_host /export/home/ACSSS/bin/cmd_proc <${NAME} &&
    sleep 2 && /bin/rm ${NAME}
```

Check to make sure that the "acsls_host" and path to the ACSLS ‘cmd_proc‘ program are correct for your installation. Duplicate this line within the script, comment one copy, and edit the other copy so it looks like:

```
rsh acsls_host /export/home/ACSSS/bin/cmd_proc <${NAME}
```

This will cause the script to no longer clean up the "NNNN.PT_STKCMD.NNN" files that the jobs generate in the ProMAX flow directory.

Now re-run the job that failed, and when you get the same mount error on the OpCon again, cancel the mount. The last "NNNN.PT_STKCMD.NNN" ACSLS command file which the job attempted to execute is now leftover in the job's flow directory. Manually execute this failed operation from the command line by entering:

```
rsh acsls_host /export/home/ACSSS/bin/cmd_proc <
$PROMAX_HOME/area/line/flow/NNNN.PT_STKCMD.NNN
```

where "NNNN.PT_STKCMD.NNN" is the most recent ACSLS command file found in the flow directory. Does this work? If you got some error when trying this, check the "DEVINFO" setting of the device's entry in the PROMAX_ETC_RDB_HOME 'config_file', the configuration file used to initialize the PMX Resource Manager. The DEVINFO should be set to the correct 'ACS,LSM,Panel,Unit' numbers. If you get library errors, check the user's ".cshrc" file to make sure that LIBPATH/LD_LIBRARY_PATH are not being set there which may prevent proper execution of the 'cmd_proc' program on the ACSLS host.

To further check for proper robotics operation, try various 'cmd_proc' commands completely independent of ProMAX. Log onto the ACSLS host. Execute the ~acsss/bin/cmd_proc program. Issue the command:

```
ACSSA> query drive n,n,n,n
```

where "n,n,n,n" are the 'ACS,LSM,Panel,Unit' numbers of each of the drives defined for the PMX Resource Manager. These should all have a State/Status of "online/available". Enter:

```
ACSSA> query volume vvvvvv
```

where "vvvvvv" was the VSN number that 'ProClient' tried to access in the failed job. You can turn on the debug options within the 'ProClient' script to find out which VSNs are being accessed. Enter:

```
ACSSA> mount vvvvvv n,n,n,n
```

to manually mount the volume which the ProMAX job requested on the drive which was involved in the failed job. Finally, enter:

```
ACSSA> dismount vvvvvv n,n,n,n force
```

to dismount the volume. If either the 'query drive', 'query volume', 'mount', or 'dismount' had problems, you may have hardware or ACSLS problems. Also try checking all SCSI cabling, IPL-ing the robot silo, and executing a:

```
ACSSA> audit
```

and then re-running either the ProMAX job or the previously described troubleshooting steps.