ConvexOS Tape System Manager's Guide

First Edition

CONVEX COMPUTER CORPORATION

CONVEX Computer Corporation
3000 Waterview Parkway
P.O. Box 833851
Richardson, TX 75083-3851
United States of America
(214)497-4000

ConvexOS Tape System Manager's Guide

Order No. DSW-398

First Edition
July 1992

CONVEX Press
Richardson, Texas
United States of America

ConvexOS Tape System Manager's Guide

Order No. DSW-398

Printed in the United States of America

Revision information for

ConvexOS Tape System Manager's Guide

Edition	Document No.	Description
First	710-020430-001	Initial release, July 1992, with ConvexOS and Utilities V10.1.

Contents

Examples

11 Operating the ACS silo

Tables

Using this guide

The *ConvexOS Tape System Manager's Guide* is a guide for configuring and maintaining the ConvexOS Tape System. Most of the commands described in this guide require either superuser authorization or membership of the group *tapeadm*.

Notational conventions

This section describes notational conventions used in this guide. These conventions include:

- Typographical conventions, or what various typefaces mean
- Command syntax and how to distinguish requirements for commands and examples
- Difference between "entering" and "typing" commands
- Identifying cautions

Typographical conventions

The following typefaces have special meaning and are used in this guide:

`Bold courier`	Identifies user input in examples.
`Courier`	Identifies input and output, including: • Command names • System calls • Data structures and types • Error messages
Italic	Identifies: • User-supplied variables in a command-line example • New and important terms • Titles of documents
KEYCAP	Indicates keyboard keys to be pressed. For example, RETURN refers to the carriage return key. Two KEYCAP terms separated by a hyphen indicate two keys that you must press simultaneously. For example, CTRL-d indicates that you must press the d key while holding down the CTRL key.

Entering versus typing commands

In this guide there is a distinction between entering a command and typing a command. This distinction is:

type	Type in the command at the keyboard without pressing the **RETURN** key.
enter	Type in the command at the keyboard and follow by pressing the **RETURN** key.

Command syntax and example conventions

In the following example:

```
command   [options...]   {a | b}      filename[...]  \
   ①          ②           ③             ④      ⑤
   ⑥ ⋮
```

① `command` must be typed as it appears.

② Brackets indicate a part of a command that is optional. A horizontal ellipsis indicates that the preceding term can be repeated. The brackets and ellipsis are not typed.

③ Braces indicate a part of a command that is necessary, but that there may be a choice in what is supplied. In this example, values for either variable *a* or *b* must be supplied, but not both. The braces and vertical line are not typed.

④ In this example, *filename* is a variable that must be supplied. The ellipsis enclosed in brackets indicates additional arguments of the same type can be supplied, but are not necessary. Again, neither the brackets nor the ellipsis is typed.

⑤ A backslash (\) at the end of a line indicates that an example continues on the next line of text. If the example is to be entered, do not press **RETURN** until the entire example is typed.

⑥ Vertical ellipsis indicates that part of an example has been omitted. The vertical ellipsis is not typed.

Identifying cautions

"Caution" labels are used frequently throughout this guide. They are of this format and have the following meaning:

Caution

A "Caution" label and this typeface highlight procedures or information necessary to avoid damage to equipment, software, or data.

Accessing associated documentation

Using this software may require information not specific to the tasks described in this document. This section describes where to find additional information and assistance.

Accessing man pages

For more information on ConvexOS, use the online man pages. To view a man page enter:

`man` *command*

where *command* is any valid ConvexOS command.

References to man pages throughout this document are in the form:

cat(1)

where the man page's section number, enclosed in parentheses, follows.

Accessing paper documentation

For more in-depth information on ConvexOS, you can order these books from CONVEX Computer Corporation:

- *ConvexOS Man Pages for System Managers* (DSW-333), *ConvexOS Man Pages for Programmers* (DSW-332), and *ConvexOS Man Pages for Users* (DSW-331), hard copy versions of the ConvexOS man pages
- *ConvexOS Primer* (DSW-133), an introduction to ConvexOS for new users
- *ConvexOS Tape System Operator's Guide* (DSW-397), a guide and reference for ConvexOS Tape System operators
- *ConvexOS Tape System Quick Reference* (DSW-391), a quick reference for the ConvexOS Tape System
- *ConvexOS Tape System User's Guide* (DSW-395), a guide and reference for the ConvexOS Tape System
- *ConvexOS* `dump` *and* `restore` *Quick Reference* (DSW-392), a quick reference for dumping and restoring file systems
- *Managing ConvexOS: Configuration Guide* (DSW-030), a guide for configuring ConvexOS
- *Managing ConvexOS: Operations Guide* (DSW-031), a guide for ConvexOS maintenance and operations

Ordering additional paper documentation

To order the current edition of this or any other CONVEX document, send requests to:

CONVEX Computer Corporation
Customer Service
P.O. Box 833851
Richardson, TX 75083-3851 USA

If possible, please include the order number (DSW number) or the exact title of the document you are ordering.

Acquiring technical assistance

If you have questions that are not answered in this book, contact the CONVEX Technical Assistance Center (TAC) at the following locations:

- Within the continental U.S., call: 1 (800) 952-0379.
- From Canada, call: 1 (800) 345-2384.
- All other locations, contact the local CONVEX office.

You can also use the `contact` utility to report any problems you may have with ConvexOS or its associated documentation. For more information refer to the contact(1) man page in *ConvexOS Man Pages for Users*, or the appendix "Using `contact`" in the *ConvexOS Primer* or *Managing ConvexOS: Operations Guide*.

Tape system overview

1

This chapter describes the basic structure of the ConvexOS Tape System.

To use tapes locally on your system, your system must have at least one connected physical tape drive. A tape drive is the mechanical device on which an actual tape is placed.

Tape drive access

To access a tape drive, you must create at least one tape device for each tape drive. A tape device is a special file that represents an I/O device and contains characteristics of the I/O device. Physical devices are accessed through these special files in the same way an ordinary file is accessed. For information on the location and naming convention for device files, refer to Chapter 4, "Adding tape drives and devices," on page 29.

To access a tape drive through the tape system, you must logically configure the tape drive using the `tpconfig` utility. This logical configuration specifies whether or not the tape drive is under the control of the tape system and the timeout factor for the tape drive.

You must also specify the characteristics by which the tape drive is accessed, such as tape speed, tape density, whether or not to use buffering (applies to 3480 and DAT drives only), and whether or not to rewind when the file is closed. Because access to drives is through special files called *tape devices*, characteristics are defined for a tape device (called a node at this point) associated with a physical tape drive.

Node configurations

Multiple nodes can be configured for each tape drive, with each node having a different characteristic set. This allows you to define multiple characteristic sets for a single tape drive. In this way, you can access the same tape drive under several sets of criteria. For example, you can specify the following nodes for a single tape drive:

/dev/mt8	1600 bpi density, block access, automatic rewind
/dev/mt12	1600 bpi density, block access, no automatic rewind
/dev/mt16	6250 bpi density, block access, automatic rewind
/dev/mt20	6250 bpi density, block access, no automatic rewind
/dev/rmt8	1600 bpi density, character access, automatic rewind
/dev/rmt12	1600 bpi density, character access, no automatic rewind
/dev/rmt16	6250 bpi density, character access, automatic rewind
/dev/rmt20	6250 bpi density, character access, no automatic rewind

Generally, each drive has eight nodes associated with it.

Users can allocate tape drives using the `tpmount` command. If they specify a tape density, tape speed, access mode, or rewind capability, the tape system selects an appropriate node and tape drive based on the characteristics of the associated nodes. If users do not specify values for one or more of these criteria, default values for unspecified criteria are used. Default values are set by the system manager using `tpconfig`.

Tape system requirements

The tape system does not function unless the following daemons exist on the system:

- `inetd`
- `portmap`
- `opreq_daemon`
- `tpdaemon`

The features of the tape system are provided through these daemons. Even though you can access tape drives without the daemons (for instance, using the `tar` command), the features of the tape system will not be available if these daemons do not exist on the system. For more information refer to Chapter 2, "Keeping the tape system up and running," on page 7.

You cannot use labeled tapes if the following daemons do not exist on the system:

- `ansidaemon`
- `ibmdaemon`

Other programs needed by the tape system are located in /usr/convex; this directory must be in the search path of all tape system users.

tpdaemon and opreq_daemon

The tape system is not active unless the tape system daemon, `tpdaemon`, is running. This process is started when the system boots and should always be running. If `opreq_daemon` is not running, `tpdaemon` must be restarted *after* restarting `opreq_daemon` in the background.

Caution

If `tpdaemon` is not running, devices normally under control of the tape system are not protected, and any user who has permission to access the special device files and who has physical access to the tape drives can use them.

`tpdaemon` uses the characteristics defined for each tape device in order to select a tape device for a tape system user's mount request.

When `tpdaemon` starts up, it examines the tape system configuration database file and opens one device for every controlled logical tape drive. This prevents users, including root, from gaining access to the tape drive outside of the tape system because it is already opened by a process.

For more information on how `tpdaemon` functions and on basic tape system management, refer to the chapter, "Managing the Tape System," in the book *Managing ConvexOS: Operations Guide.*

The Operator Request Manager daemon

The `opreq_daemon` manages the Operator Request Manager (`opreq`) and `opreq` queueing. The `opreq` utility manages tape requests queued by `opreq_daemon`. `opreq` queueing allows tape mount requests to "wait in line" until the necessary resources are available. The tape operator uses `opreq` to manage tape requests.

`opreq_daemon` is started *before* `tpdaemon`. Consequently, `tpdaemon` must be killed and restarted if `opreq_daemon` should happen to die.

Tape label daemons

A label daemon, either `ansidaemon` or `ibmdaemon`, is forked by `tpdaemon` whenever a request to mount an ANSI- or IBM-labeled tape is processed. These daemons are filters between the user and the tape device. They handle tape label operations, such as recognition, creation, and deletion of labels. They only run when label processing is in use. One daemon is started for each mounted tape device through which ANSI- or IBM-labeled tape I/O is done. Thus, there may be more than one of each of these processes at any one time. Each daemon exits when its device is unmounted.

Caution

It is a security risk to use unlabeled tapes, because it permits the user to ignore access restrictions imposed by the tape labels.

Label verification on VMEbus drives is done each time the tape is at beginning-of-tape and a request to access it (read or write) occurs. This prevents users from switching to a tape with restrictive access rights after mounting a tape with unrestrictive access rights.

Label verification on Multibus drives is performed only when the drive is first mounted. Consequently, the system might not recognize an attempt to breach security by tape switching if you use Multibus drives. The operator should not change the tape until the Multibus drive is dismounted.

Keeping the tape system up and running

2

This chapter describes the daemons that are required for the tape system and how to restart the tape system in the event there is a problem.

The /etc/rc.std file and the tape system

Check the /etc/rc.std file to ensure that `tpdaemon` and `opreq_daemon` are automatically started on reboot.

`tpdaemon` and `opreq_daemon` are started by entries in the /etc/rc.std file, which is executed by ConvexOS when it boots. The lines shown in Example 1 must appear in the /etc/rc.std file.

Example 1 Tape daemon entries in the /etc/rc.std file

```
if [ -f /usr/lib/opreq/opreq_daemon ]; then
   /usr/lib/opreq/opreq_daemon & echo -n ' opreq_daemon'
fi

if [-f /usr/lib/tape/tpdaemon]; then
   /usr/lib/tape/tpdaemon & echo -n ' tpdaemon'
fi
```

If you are using tape system operators and `opreq`, `opreq_daemon` must be started before `tpdaemon`. For reference information about `ansidaemon`, `tpdaemon`, and `opreq_daemon`, see the ansidaemon(8), ibmdaemon (8), opreq_daemon(8) and tpdaemon(8) man pages.

Checking on daemons necessary for the tape system

For the tape system to stay up and running, the following four daemons must be running:

- `portmap`
- `inetd`
- `opreq_daemon`
- `tpdaemon`

To check on these daemons, enter the command at your system prompt as shown in Example 2.

Example 2 Checking on daemons necessary for the tape system

```
% ps aux | egrep "PID|portmap|inetd|opreq_daemon|tpdaemon"
USER PID %CPU %MEM  SZ  RSS  TT  STAT TIME COMMAND
root 217  0.0  0.2 276  112  ?   S    0:07 inetd
root 109  0.0  0.1 228   48  ?   S    0:27 /etc/portmap
root 229  0.1  0.3 700  148  ?   S    0:02 /usr/lib/tape/tpdaemon
root 228  0.0  0.1 392   28  ?   S    0:00 /usr/lib/opreq/opreq_daemon
% ▮
```

Not only must `portmap`, `inetd`, `opreq_daemon`, and `tpdaemon` be present for the tape system to remain functional, but in *most* cases their process IDs (PIDs) should adhere to the following criteria:

- The PID of `portmap` should be less than that of `inetd`.
- The PID of `inetd` should be less than that of `opreq_daemon`.
- The PID of `opreq_daemon` should be less than that of `tpdaemon`.

If you are having problems with the tape system, sometimes it is best to bring it down and restart it.

Basic tape system configuration

3

Many variables that control the way in which the tape system performs can be set with `tpconfig`, the tape system configuration utility. Configuring the tape subsystem entails:

- Setting parameters that control the functioning of the tape system, such as:
  - Enabling/disabling `opreq` queueing
  - Defining tape label types
  - Setting defaults for `tpmount`, `tplabel`, and `tpattr`
- Defining characteristics of tape drives, such as:
  - Drive type
  - Control status
  - Timeout
  - Bypass labeling permission
  - Allocate drive permission
  - Access drive permission
- Defining characteristics of nodes (nodes are devices associated with one of the drives defined with `tpconfig`). The following parameters can be set individually for each node associated with a drive:
  - Buffered status
  - Speed
  - Density
  - Rewind status

This chapter describes the basics of the `tpconfig` utility and how it affects entries in the tape system configuration database, /usr/lib/tape/config.db.

Tape system variables and their defaults

The tape system is shipped with the following default configuration:

- Queuing with the `opreq` utility is disabled.
- No users or groups are allowed to bypass label processing.
- All users have allocate tape drive permission.
- Default density is 6250 (for `tpmount`).
- No default speed (for `tpmount`).
- One controlled drive (mt:0) is defined, along with associated nodes.
- No-rewind character special device is the default for `tpmount`.
- ANSI-standard is the default label type.
- Labeled tape restriction is disabled.
- Tape volume and file restrictions are disabled.

Tape system configuration commands

The `tpconfig` utility is used to configure the tape system. `tpconfig` modifies the tape system configuration file called /usr/lib/tape/config.db.

Caution

Do not attempt to alter the /usr/lib/tape/config.db file with a text editor; configuration parameters can be changed only with the `tpconfig` utility.

With the exception of setting the `opreq` queueing status, all changes made via `tpconfig` take effect within a few seconds after exiting `tpconfig`. If the queueing status is changed, the change does not go into effect until there are no active `tpmount` requests. `tpmount` requests are active until canceled by operator intervention, by the system, or by a corresponding `tpunmount` command.

Only one `tpconfig` can be running on a system at any one time; the tape system database is locked while `tpconfig` is active. If a second instance of `tpconfig` is started, its access to the configuration database is read-only.

You must be root or be a member of the group *tapeadm* to change the tape system configuration; if you are not root, `tpconfig` will run but will not make changes to the tape system configuration file. Under ConvexOS/Secure, you must also have tapeadm authorization.

Command format notation

`tpconfig` commands are not case sensitive. Also, there are valid abbreviations of each command. Because there are a number of different ways you can enter each command, the notational conventions described in this book are as follows:

- The full command is shown in its syntax description.
- The shortest, unambiguous abbreviation is shown in an accompanying example.

For instance, the complete syntax of the `set control` command is shown as:

`tpconfig set control {on|off}` *type:unit*

However, in its usage example, it is written in its abbreviated form, which is:

```
# tpconfig se c on mt:4
```

When a command is discussed in text, the unabbreviated, lowercase form is used and is shown in `monospace` type.

The three modes of `tpconfig`

You can use `tpconfig` in:

- Single command mode
- Interactive mode
- Batch mode

Each method is described in the following sections of this chapter.

Using `tpconfig` in single command mode

To use `tpconfig` as a single command, enter the `tpconfig` command followed by one of the commands recognized by `tpconfig`. For example:

```
# tpconfig add drive tc:1
```

This command adds a tape drive called tc:1 to the tape system. For a complete list of these commands, refer to the section, "Listing tpconfig commands," on page 19 or the tpconfig(8) man page.

Using `tpconfig` interactively

To use `tpconfig` interactively, enter:

```
# tpconfig
```

The shell prompt is replaced by the `tpconfig` prompt:

```
Tpconfig>
```

You can enter any of the commands available with `tpconfig`. For a complete list of these commands, refer to the section, "Listing tpconfig commands," on page 19 or the tpconfig(8) man page. When a command executes, the `Tpconfig>` prompt reappears, and you can enter additional commands. To exit `tpconfig`, press **CTRL-d**.

Example 3 illustrates a sample interactive session.

Example 3 An interactive `tpconfig` session

```
# tpconfig
Tpconfig> add drive mt:1
Tpconfig> add node /dev/rmt9 den=1600 rew mt:1
Tpconfig> add node /dev/rmt13 den=1600 mt:1
Tpconfig> add node /dev/rmt17 den=6250 rew mt:1
Tpconfig> add node /dev/rmt21 den=6250 mt:1
Tpconfig> add node /dev/mt9 den=1600 rew mt:1
Tpconfig> add node /dev/mt13 den=1600 mt:1
Tpconfig> add node /dev/mt17 den=6250 rew mt:1
Tpconfig> add node /dev/mt21 den=6250 mt:1
Tpconfig> ^D
# ▮
```

Using `tpconfig` in batch mode

To use `tpconfig` in batch mode, specify an input file containing the `tpconfig` commands you want executed. When you invoke `tpconfig` in the following way:

`tpconfig <` *commands.file*

the commands in *commands.file* are executed.

The most common way to use batch mode is when restoring the tape system to a previous configuration using a configuration snapshot in place of *command.file*. For more information on taking a snapshot of your tape system configuration, refer to the section, "Saving an image of your tape system configuration," on page 18.

Saving an image of your tape system configuration

Use the `tpconfig snapshot` command to save the current configuration of your tape system. If you must restore the tape system configuration at a later time, you can use this file to produce the tape system configuration that was in effect when you took the snapshot.

The syntax of the `tpconfig snapshot` command is

`tpconfig snapshot >` *file*

where *file* is the output file to which you want your current configuration saved. Unless you specify a full path name, *file* is saved in the directory you are in at the time the snapshot is taken. If you do not specify an output file, output is sent to standard output.

You can edit the output file, *file*, with a standard text editor; however, the `tpconfig` database itself cannot be edited with a text editor.

For more information on how to use a snapshot to restore your tape system configuration, refer to the section, "Using tpconfig in batch mode," on page 17 or to Chapter 4, the section "Matching the config.db file to the /ioconfig file when adding or deleting drives," on page 41.

Listing `tpconfig` commands

Use the `help` command to view a complete list of commands available with `tpconfig`. The output from the `help` command is shown in Example 4.

Example 4 Using the `help` command

```
% tpconfig help
Database access is read-only.
Add Drive <type:unit> [Nocontrol] [Timeout=<N>]
Del Drive <type:unit>
Add Node <path> [Speed=<string>] [Density=<string>] [Rewind]
[Buffered] <type:unit>
Del Node <path>
Add Label <label_type> <daemon>
Del Label <label_type>
Del Silohost <type:unit>
Add STACKER <path> <type:unit>
Del STACKER <type:unit>
Add STACKERDaemon <path>
Del STACKERDaemon
Add/Del/SEt ACess_drive User_set <user_list...> <type_unit>
Add/Del/SEt ACess_drive Group_set <user_list...> <type_unit>
Add/Del/SEt ALlocate_drive User_set <user_list...> <type_unit>
Add/Del/SEt ALlocate_drive Group_set <user_list...> <type_unit>
Add/Del/SEt Bypass_labels User_set <user_list...> <type_unit>
Add/Del/SEt Bypass_labels Group_set <user_list...> <type_unit>
SEt Control ON|OFf <type:unit>
SEt Default DRive <type>
SEt Default DEnsity <density>
SEt No_default Density
SEt Default File-access [none|r|w|rw]
SEt Default Flags [Rewind|Norewind] [Character|Block|Labeled]
SEt Default Speed <speed>
SEt Labeled_mode_only Enabled|Disabled
SEt No_default Speed
SEt Silohost <hostname> <drive_string> <type:unit>
SEt Timeout <N> <type:unit>
SEt Queueing Enabled|Disabled
SEt Restrict Vol_headers Enabled|Disabled
SHow All
SHow DEfaults
SHow DRive [<type:unit>]
SHow Labels
SHow Node [<path>]
SHow Stackerdaemon
SNapshot [<file>]
% ▮
```

Displaying tape system configuration information

Although the `tpconfig` command is usually used by system managers to configure the tape system, you can use `tpconfig` to display read-only information about tape system configuration.

The information displayed by `tpconfig show` commands can be very useful in helping you select a tape device. If you have more than one tape drive and want to use multiple tapes simultaneously, you must specify devices with your mount request. Use `tpconfig show` commands to view the devices available.

Syntax of the `tpconfig show` commands

The syntax for the `tpconfig show` command is described in Table 1.

Table 1 `tpconfig show` commands

Command	Description
`tpconfig show all`	Show all tape system configuration values.
`tpconfig show de`	Show default values for mounting tapes.
`tpconfig show dr [drive]`	Show information about a drive, where *drive* is the name of the drive you specify. For example: **`tpconfig sh dr mt:0`** displays information about the tape drive mt:0. If you do not specify a tape drive, you receive information about all tape drives.
`tpconfig show labels`	Show defined label types (IBM and ANSI standards, and non-labeled).
`tpconfig show node [node]`	Show information about a specific node (tape device), where *node* is the path name of the node you specify. For example: **`tpconfig sh n /dev/rmt20`** shows information about the /dev/rmt20 device. If you do not specify a node, you receive information about all nodes.

Examples of displaying information on tape system configurations

The following sections contain several examples of how to use each `tpconfig show` command, and what the output represents.

Displaying all tape system configuration information

Example 5 shows how to display all tape system configurations using the `tpconfig show all` command:

① Tape drive type mt and unit 0.

② Number of minutes that can elapse before the idle drive is declared unused. After this interval elapses, the tape system automatically unmounts your tape.

③ This drive, mt:0, is controlled by the tape system. If the drive is uncontrolled, the control status is shown as "Not controlled."

④ All users and groups can use this drive.

⑤ All users and groups can mount tapes on this drive without specifying a volume serial number (VSN).

⑥ No users or groups can bypass tape-label processing. (root is an exception. If you have root privileges, you can always bypass label processing.)

⑦ Defined devices associated with the drive. Each of these devices is called a *node* of the drive.

⑧ IBM and ANSI-standard labels, and non-labeled type for unlabeled tape sets are recognized.

⑨ `opreq` queueing is disabled. Tapes do not need to be mounted in labeled mode only. Labeled tape file headers need not be access restricted.

⑩ The default settings:

- Density is 6250 bpi.
- No default speed has been defined.
- Drive type is mt.
- Mount mode is character, no automatic rewind.
- Files created on a tape are read-only by anyone other than the owner.

Example 5 `tpconfig show all` command output

```
% tpconfig sh a
Database access is read-only.
Drives:
①  mt:0     ② timeout: 60③ Controlled
④          Access:
             Users:     All
             Groups:    All
⑤          Alloc access:
             Users:     All
             Groups:    All
⑥          Bypass access:
             Users:     None
             Groups:    None
⑦          Nodes:
          /dev/rmt8       speed:        density: 1600    Rewind      Char
          /dev/rmt12      speed:        density: 1600    No rewind   Char
          /dev/rmt16      speed:        density: 6250    Rewind      Char
          /dev/rmt20      speed:        density: 6250    No rewind   Char
          /dev/mt8        speed:        density: 1600    Rewind      Block
          /dev/mt12       speed:        density: 1600    No rewind   Block
          /dev/mt20       speed:        density: 6250    No rewind   Block

⑧  Labels:
     nl daemon: /usr/lib/tape/nldaemon
     ibm daemon: /usr/lib/tape/ibmdaemon
     ansi daemon: /usr/lib/tape/ansidaemon

Stacker Daemon: /usr/lib/tape/stkdaemon

⑨  Queueing is: Disabled
   Labeled-mode-only: Disabled
   Restricted-Volume-Headers is: Disabled

⑩  Defaults:
     Density: 6250
     Speed: No default
     Drive type: mt
     Mount flags: Character special, No-rewind
     Default labeled tape access: r
%
```

Displaying tape system defaults

Example 6 exhibits how to show all configured defaults by using the `tpconfig show de` command:

① You can read the tape configuration database, but cannot change it unless you have root privileges on your system.

② `opreq` queueing is disabled. Tape operators will not enqueue your mount requests.

③ Default tape density is 6250 bpi.

④ No default tape speed has been specified.

⑤ Tape drive is of type mt.

⑥ Access mode is character, and there is no automatic rewind on close.

⑦ Files created on a tape are read-only to anyone other than the owner.

Example 6 `tpconfig show de` command output

```
% tpconfig sh de
① Database access is read-only.
② Queueing is: Disabled

Defaults:
③     Density: 6250
④     Speed: No default
⑤     Drive type: mt
⑥     Mount flags: Character special, No-rewind
⑦    Default labeled tape access: r
%
```

Displaying valid label types

Example 7 exhibits how to show all valid label types by using the `tpconfig show labels` command:

① You can read the tape configuration database, but cannot change it unless you have root privileges on your system.

② Only IBM and ANSI-standard labels, and non-labeled type (for unlabeled tape sets), are recognized.

③ Complete path names for the label daemons are displayed.

Example 7 `tpconfig show labels` command output

```
% tpconfig sh l
Database access is read-only. ①
Labels:
② nl daemon: /usr/lib/tape/nldaemon ③
   ibm daemon: /usr/lib/tape/ibmdaemon
   ansi daemon: /usr/lib/tape/ansidaemon
%
```

Displaying information on all tape drive configurations

Example 8 exhibits how to show all configured drives by using the `tpconfig show dr` command:

① Type **`tpconfig show dr`** and specify the drive type and unit number of the tape drive. If you do not specify a drive type and unit number, information about all the tape drives is displayed.

② You can read the tape configuration database, but cannot change it unless you have root privileges on your system.

③ Drive type and unit number. In this example, the drive is of type mt, and the unit number is 0.

④ Number of minutes that can elapse before the idle drive is declared unused. After this interval elapses, the tape system automatically unmounts the tape drive.

⑤ Access to the tape drive is controlled by the tape system.

⑥ Any user can mount drives without specifying a tape VSN.

⑦ No users or groups can bypass label processing.

⑧ Tape devices that are associated with the tape drive. Each of these tape devices is called a *node* of the drive.

Example 8 `tpconfig show dr` command output

```
① % tpconfig sh dr mt:0
② Database access is read-only.
Drives:
  ③ mt:0        ④ timeout: 60    ④ Controlled
⑥         Alloc access:
            Users:     All
            Groups:    All
⑦         Bypass access:
            Users:     None
            Groups:    None
⑧         Nodes:
        /dev/rmt8       speed:        density: 1600  Rewind     Char
        /dev/rmt12      speed:        density: 1600  No rewind  Char
        /dev/rmt16      speed:        density: 6250  Rewind     Char
        /dev/rmt20      speed:        density: 6250  No rewind  Char
        /dev/mt8        speed:        density: 1600  Rewind     Block
        /dev/mt12       speed:        density: 1600  No rewind  Block
        /dev/mt16       speed:        density: 6250  Rewind     Block
        /dev/mt20       speed:        density: 6250  No rewind  Block
%
```

Displaying information on all defined nodes

Example 9 exhibits how to show all defined node configurations by using the `tpconfig show node` command:

① The `tpconfig show node` command displays information about all defined tape devices for all drives, unless a node is specified (refer to ⑨).

② You can read the tape configuration database, but cannot change it unless you have root privileges on your system.

③ The tape drive to which each tape device (node) is associated

④ Tape devices (nodes) associated with the tape drive

⑤ Tape speed for each tape device

⑥ Tape density requirements, in bpi, for each tape device

⑦ Whether the tape device automatically rewinds on close

⑧ Whether the tape device requires block or character mode input

⑨ A `tpconfig show node` command specifying a tape device (node)

Example 9 `tpconfig show node` command output

```
① % tpconfig sh n
② Database access is read-only.
Nodes:③      ④              ⑤              ⑥          ⑦          ⑧
    mt:0  /dev/rmt8      speed:       density: 1600  Rewind     Char
    mt:0  /dev/rmt12     speed:       density: 1600  No rewind  Char
    mt:0  /dev/rmt16     speed:       density: 6250  Rewind     Char
    mt:0  /dev/rmt20     speed:       density: 6250  No rewind  Char
    mt:0  /dev/mt12      speed:       density: 1600  No rewind  Block
    mt:0  /dev/mt8       speed:       density: 1600  Rewind     Block
    mt:0  /dev/mt16      speed:       density: 6250  Rewind     Block
    mt:0  /dev/mt20      speed:       density: 6250  No rewind  Block
⑨ % tpconfig sh n /dev/rmt20
Database access is read-only.
    mt:0  /dev/rmt20     speed:       density: 6250  No rewind  Char
%
```

Adding tape drives and devices

4

This chapter describes

- How to add previously configured tape drives and devices to the tape system
- How to delete previously configured tape drives and devices from the tape system
- How to ensure that your `tpconfig` database matches your /ioconfig file on the SPU

which involves altering the `tpconfig` database.

The `tpconfig` database can only be altered by the superuser or a member of the group *tapeadm*, and, in ConvexOS/Secure, by those who have tapeadm authorization.

Tape drives and devices are configured when the appropriate hardware is added to your system. This chapter does not address this type of configuration.

Adding tape drives and nodes

This section describes how to add a tape drive to your tape system and how to add nodes to a tape drive.

Adding 9-track, cartridge, and DAT drives

To add tape drives to your tape system, use the following procedure:

Step 1 Follow the installation procedures for that are supplied with the drive you are adding. Part of this procedure involves creating an entry in the /ioconfig file on the SPU disk for the tape drive and creating tape devices for the new tape drive with the `MAKEDEV` utility. This has probably already been done for you by a CONVEX field engineer.

Step 2 Ensure that the proper drive entries exist in the /ioconfig file on the SPU disk. You can do this with the following command:

`spu -r /ioconfig | grep` *MTD_number*

MTD_number is any one of the MTD numbers in Table 2.

In the following example, an entry for a 9-track drive with a VMEbus controller exists, but an entry for a DAT drive does not.

Example 10 Checking for proper /ioconfig entries

```
% spu -r /ioconfig | grep MTD-204
      unit 0 type MTD-204
% spu -r /ioconfig | grep MTD-208
%
```

If you find you do not have the proper entry in the /ioconfig file on the SPU disk, you must create the entry and reboot. For more information, refer to the installation procedures for the device you are installing.

Table 2 MTD numbers and their associated drive types

Drive MTD number	Type of drive
MTD-001	9-track drive with Multibus controller
MTD-002	
MTD-003	
MTD-201	9-track drive with VMEbus controller
MTD-202	
MTD-203	
MTD-204	
MTD-207	3480 cartridge drive
MTD-217	3480 cartridge drive with stacker/loader
MTD-227	3480 cartridge drive with data compression
MTD-237	3480 cartridge drive with stacker/loader and data compression
MTD-301	TLI 3480 cartridge drive
MTD-208	DAT drive
MTD-218	DAT drive with stacker/loader
MTD-228	DAT drive with data compression
MTD-238	DAT drive with stacker/loader and data compression

Step 3 Add a configuration entry for each new tape drive to the `tpconfig` database.

All drives that are to be used through the tape system must be defined in the following manner. Failure to do this results in making drives inaccessible to users through tape system control. Drives that have not been defined through `tpconfig` cannot be accessed or queried by tape system commands.

Generally, drive names follow the naming convention

type:*unit*

where *type* is one of the following drive types:

`dat`	DAT drives
`mt`	9-track drives
`tc`	3480 cartridge drives

unit is the unit number of the drive. If you only have one drive of a particular type, its unit number is 0 (zero). If you have two drives of a particular type, their numbers are 0 and 1, and so on.

To add tape drives to the tape system, use the following command syntax:

`tpconfig add drive` *type*`:`*unit* `[nocontrol] [timeout=`*n*`]`

The options for `tpconfig add drive` are described in Table 3.

Table 3 `tpconfig add drive` options

Option	Description
type:unit	Drive type and unit number of the tape drive being added
`nocontrol`	Whether the tape system will control the tape drive. If you specify `nocontrol`, the drive is not accessible through the tape system.
`timeout=`*n*	How long the drive can be idle before it is timed out, where *n* is an integer specifying the number of idle minutes. When the tape drive is timed out, the tape that is on it is unmounted, and tape drive reservation is relinquished to the next user. A timeout limit must be set. Timeout limits cannot be disabled.

In the following example, a 3480 cartridge drive called tc:1 is added to the tape system. It is controlled and times out after 30 minutes. This command must be entered as superuser at the root prompt:

```
# tpconfig a d tc:1 t=30
```

Adding nodes to a defined tape drive

After a tape drive is defined by `tpconfig` (refer to "Adding 9-track, cartridge, and DAT drives" on page 30), define nodes for it. *Nodes* are tape devices that are associated with a drive. (These devices should have been created when the tape drive was installed.)

If you do not define nodes for a drive through the tape system, the tape drive cannot be accessed through the tape system.

To add nodes to a tape drive, use the following procedure:

Step 1 Ensure that devices were created for the drive you are going to add to your system with the following command:

`ls /dev/`*`type`*`*`

type is the drive type of the drive that has been installed.

In Example 11, mt drives have been properly installed, but DAT drives have not.

Example 11 Ensuring tape devices have been created

```
# ls /dev/tc*
/dev/tc0   /dev/tc16   /dev/tc20
/dev/tc12  /dev/tc4    /dev/tc8
# ls /dev/dat*
/dev/dat* not found
#
```

Table 4 lists recommended tape device names for associated tape drives of various types. Look in the /dev directory to see if you have the recommended tape devices for your tape drive.

Table 4 Names of tape drives and recommended names and attributes of associated nodes

Tape drive and associated node names						Node attributes			
dat:0†	mt:0	tc:0†	dat:1†	mt:1	tc:1†	Density	Rewind Status	Mode	Buffered?
	mt8			mt9		1600	Rewind	Block	
	mt12			mt13		1600	No rewind	Block	
dat0u	mt16	tc0u	dat1u	mt17	tc1u	6250	Rewind	Block	
dat0nu	mt20	tc0nu	dat1nu	mt21	tc1nu	6250	No rewind	Block	
dat0		tc0	dat1		tc1	6250	Rewind	Block	Buffered
dat0n		tc0n	dat1n		tc1n	6250	No rewind	Block	Buffered
	rmt8			rmt9		1600	Rewind	Char	
	rmt12			rmt13		1600	No rewind	Char	
rdat0u	rmt16	rtc0u	rdat1u	rmt17	rtc1u	6250	Rewind	Char	
rdat0nu	rmt20	rtc0nu	rdat1nu	rmt21	rtc1nu	6250	No rewind	Char	
rdat0		rtc0	rdat1		rtc1	6250	Rewind	Char	Buffered
rdat0n		rtc0n	rdat1n		rtc1n	6250	No rewind	Char	Buffered

† When an "i" is appended to any of the node names in these columns, this indicates that the node offers data compression.

Step 2 If you do not have the recommended tape devices as listed in Table 4, you should create them with the `MAKEDEV` command as follows:

`MAKEDEV {ta|tc|dat}`*unit*

`MAKEDEV` has the options:

`{ta|tc|dat}`	Specify one of the following:	
	`ta`	To create 9-track devices
	`tc`	To create 3480 cartridge devices
	`dat`	To create DAT devices
unit	The unit number of the tape drive for which you are creating devices.	

In the following example, the tape devices for dat:1, mt:1, and tc:1 are created, respectively:

```
# MAKEDEV dat1
# MAKEDEV ta1
# MAKEDEV tc1
```

Step 3 After ensuring that you have the proper devices, you need to add them to the tape system. Tell the tape system the attributes of each tape device and which tape drive it is associated with by using the following command syntax:

`tpconfig add node` *path* `[speed=`*speed*`] [density=`*density*`] [rewind] [buffered]` *type*`:`*unit*

The options for `add node` are described in Table 3.

Table 5 `tpconfig add node` options

Option	Description
path	Full path name of the tape device being defined as a node. For example: /dev/rmt8.
`speed=`*n*	Speed of the device, where *n* is an integer representing inches-per-second (ips).
`density=`*n*	Density of the tape device in bytes-per-inch, where *n* should be either `1600` (for mt devices only) or `6250` (for mt, tc, and DAT devices). With 3480 cartridge and DAT drives the actual tape density is not 6250, even though this is the recommended setting for their devices. Because these drives have a singular density, it is best to set the densities on their devices so that users are not required to explicitly state the density on the command line. Refer to the mtio(4) man page for more information about valid tape densities.
`rewind`	Automatic rewind after closing a tape file. A file on tape is opened when it is read from or written to, and closed when the process is completed. If rewind is set, the tape is automatically rewound when any file on tape is closed. If rewind is not specified, the tape device is "no rewind."
`buffered`	Whether or not internal buffering is enabled. Some 3480 and DAT tape drives have internal hardware buffers. If buffered is set, then internal buffering is enabled through the tape device. If buffered is not specified, the tape device does not allow internal buffering.
type:*unit*	The drive type and unit number of the associated tape drive.

The easiest way to define a number of drives is to create a batch file with the `add node` commands in it, then run the batch file through `tpconfig`. Example 12 shows a file named drive_config that is composed of commands to add tc:1 devices to the tape system. The commands are shown in their most abbreviated form. Example 12 also shows how to run the file through `tpconfig`.

Example 12 Adding many nodes to a tape drive

```
# cat drive_config
# /dev/tc1u d=6250 r tc:1
# /dev/tc1nu d=6250 tc:1
# /dev/tc1 d=6250 r b tc:1
# /dev/tc1n d=6250 b tc:1
# /dev/rtc1u d=6250 r tc:1
# /dev/rtc1nu d=6250 tc:1
# /dev/rtc1 d=6250 r b tc:1
# /dev/rtc1n d=6250 b tc:1
# tpconfig < drive_config
#
```

After a tape device is associated with a tape drive, it is a node of that drive. All nodes added to a tape drive share its control status and timeout value, as well as its label bypass and allocation permissions.

The `add drive` command, as all `tpconfig` commands, can also be run in either command or interactive mode.

Step 4 Use the following `tpconfig show drive` command syntax to verify that all nodes and drives were correctly added:

`tpconfig show drive` *type*`:`*unit*

type is the drive type, and *unit* is the unit number of the drive you want to check.

In Example 13 the tc:1 drive and its nodes are shown.

Example 13 Showing nodes associated with a drive

```
% tpconfig sh dr tc:1
Database is read-only.
    tc:1     timeout: 30     Controlled
         Access:
              Users:    All
              Groups:   All
         Alloc access:
              Users:    All
              Groups:   All
         Bypass access:
              Users:    None
              Groups:   None
         Nodes:
         /dev/rtc1    speed:       density: 6250   Rewind     Char   Buffered
         /dev/rtc1u   speed:       density: 6250   Rewind     Char
         /dev/rtc1n   speed:       density: 6250   No rewind  Char   Buffered
         /dev/rtc1nu  speed:       density: 6250   No rewind  Char
         /dev/tc1     speed:       density: 6250   Rewind     Block  Buffered
         /dev/tc1u    speed:       density: 6250   Rewind     Block
         /dev/tc1n    speed:       density: 6250   No rewind  Block  Buffered
         /dev/tc1nu   speed:       density: 6250   No rewind  Block
# ▮
```

Deleting tape drives and nodes from the tape system

This section describes how to remove tape drives from the tape system, and how to remove nodes from a tape drive.

Deleting tape drives

When a tape drive is deleted, it is removed from the tape system. The removed tape drive can no longer be accessed through tape system controls, nor can any information be obtained about it through tape system commands.

You can remove a tape drive, along with its nodes, from the tape system using the command syntax:

`tpconfig delete drive` *type:unit*

type:*unit* is the drive type and unit number, respectively, of the tape drive you are deleting.

For example, to delete drive mt:1 from the tape system, enter at the root prompt:

```
# tpconfig d d mt:1
```

Deleting nodes

When a node is deleted, it is removed from the tape system and is no longer associated with a tape drive. The removed node can no longer be used to access a tape drive through the tape system.

You can remove nodes from a tape drive using the command syntax:

`tpconfig delete node` *path*

path is the path name of the node you are deleting.

For example, to delete node rmt9 from the tape system, enter at the root prompt:

```
# tpconfig d n /dev/rmt9
```

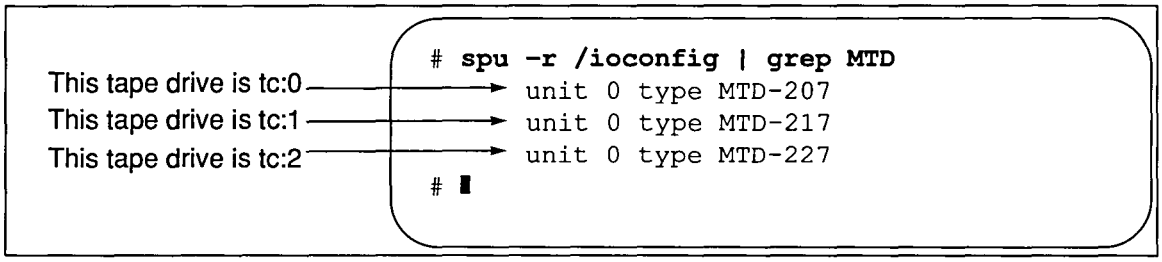
Matching the config.db file to the /ioconfig file when adding or deleting drives

After adding or removing a tape drive to your system and modifying the appropriate entry in the /ioconfig file on the SPU disk, you may need to reconfigure your `tpconfig` database. If you have tape drives of identical types (dat, mt, tc), but have different MTD numbers in the /ioconfig file, you may need to reconfigure your `tpconfig` database.

When rebooting the system after adding an entry to the /ioconfig file, the tape drive unit numbers are reassigned; however, the `tpconfig` database is not automatically updated. This may cause tape devices defined in the `tpconfig` database to be mismatched with their associated tape drives.

The following example shows a list of 3480 cartridge tape drives that have different MTD numbers in the /ioconfig file:

Example 14 /ioconfig entries that show drives of identical types and different MTD numbers



If the tc:1 drive and its entry in the /ioconfig file were removed, the tape drive that was previously named tc:2 would be renamed tc:1 when the system is rebooted. In this case, the following situations must be corrected:

- The tape drive and devices that are currently defined in the tape system for tc:1 and tc:2 must be removed.
- The tape drive and devices for tc:1 must be redefined.

The following procedure describes the easiest way update your `tpconfig` database after adding or deleting a tape drive from a list of identically typed but differently configured tape drives from the /ioconfig file.

Step 1 If adding a tape drive, follow the appropriate installation procedures, which include editing the /ioconfig file on the SPU disk, creating the appropriate devices with MAKEDEV, and rebooting your system.

If removing a tape drive, remove its corresponding entry from the /ioconfig file on the SPU disk and reboot your system.

Step 2 Take a snapshot of your current tape system configuration by using the command syntax:

`tpconfig snapshot` *filename*

filename is the name you want to give your configuration snapshot file.

In the following example a snapshot named new.config is created:

```
# tpconfig sn > new.config
```

Step 3 Edit the snapshot file so that it has the correct entries for your new tape system configuration.

Step 4 Remove the current `tpconfig` database by entering the following as superuser at the root prompt:

```
# rm /usr/lib/tape/config.db
```

Step 5 Run the edited and updated version of your `tpconfig` snapshot through `tpconfig` as follows:

`tpconfig <` *filename*

filename is the name of your edited and updated `tpconfig` snapshot.

In the following example the edited `tpconfig` snapshot, new.config, is used to update the tape system according to the new configuration:

```
# tpconfig < new.config
```

It is not necessary to restart `opreq_daemon` or `tpdaemon`. The changes that were made are effective immediately except for queuing status, which is activated the next instance when the tape system is idle.

Resetting timeout limit and control status of a tape drive

Timeout limit and control status are set when a tape drive is defined in the tape system with the `add drive` command. The commands in this section change these settings after the tape drive has been defined.

Resetting timeout limit

The timeout limit is the number of minutes a tape drive can stand idle before its tape is unmounted and its reservation relinquished. Change the timeout limit using the following command syntax:

`tpconfig set timeout` *n type:unit*

This requires you to supply the following information:

n — An integer representing the number of minutes a tape drive can be idle before it is timed out.

type:unit — Drive type and unit number of the tape drive whose timeout limit you are changing.

For example, to set a timeout of 120 minutes for mt:1, enter as root:

```
# tpconfig s t 120 mt:1
```

The `tpconfig set timeout` command cannot be used to shut off a drive's timeout limit. A timeout limit must be set.

Resetting control status

The control status of a tape drive is whether or not it is controlled by the tape system. Change a tape drive's control status with the command syntax

`tpconfig set control {on|off}` *type:unit*

which requires you to supply the following information:

`on|off` — Switch tape system control of the drive on or off.

type:unit — Drive type and unit number of the tape drive whose control status you are changing.

For example, to remove the tape drive mt:1 from tape system control, enter as root:

```
# tpconfig s c of mt:1
```

Setting tape system defaults 5

This chapter describes how to set defaults in the `tpconfig` database for mounting tapes through the tape system. Tape system mounting defaults are:

- Access mode (flag)
- Automatic rewind on close (flag)
- Drive type selection
- File access permissions
- Tape density
- Tape speed

The `tpconfig` database can only be altered by the superuser or a member of the group *tapeadm*, and, in ConvexOS/Secure, by those who have tapeadm authorization.

Setting default drive type

You can set the default drive type for any mount request that does not specify a specific tape drive or drive type.

The command to set a default drive type is

`tpconfig set default drive` *type*

where *type* is the drive type.

Drive type can be any one of the following:

`dat`	For DAT drives
`mt`	For 9-track drives
`tc`	For 3480 cartridge drives

For example, to set the default drive type to mt, enter as superuser at the root prompt:

```
# tpconfig se d dr mt
```

Setting and unsetting default tape density

You can set the default tape density for any mount request that does not specify a density.

The command to set a default tape density for mount requests is

`tpconfig set default density` *density*

where *density* is the default density in bytes-per-inch (bpi), and must match at least one of the defined nodes. Tape density is usually either 1600 or 6250 bpi.

For example, to specify a default density of 1600 bpi, enter as superuser at the root prompt:

```
# tpconfig s d de 1600
```

This indicates to the tape system to select a node with a tape density of 1600 bpi if the mount request does not specify otherwise.

To unset a previously set default density so that there is no default density, use the command:

`tpconfig set no_default density`

You can also enter as superuser at the root prompt:

```
# tpconfig se n d
```

After unsetting the default tape density, if a tape system user does not specify a tape density with their mount request, density is not a criterion the tape system uses in selecting a node. Under this circumstance, a node of any density can be selected by the tape system in response to the mount request.

Setting default flags for rewind status and access mode

You can set the default flags for rewind status and access mode for any mount request that does not specify otherwise.

The command to set the flags indicating automatic rewind on close and access mode is

```
set default flags [rewind|norewind] [character|block|labeled]
```

which has the following options:

`[rewind|norewind]`	Whether or not the tape is automatically rewound immediately after a file is closed after being read from or written to.
`[character|block|labeled]`	Access mode.

For example, to set no automatic rewind and labeled access as defaults, enter as superuser at the root prompt:

```
# se d fl n l
```

Setting and unsetting default tape speed

You can set the default tape speed for any mount request that does not specify a speed.

The command to set a default tape speed for mount requests is

`set default speed` *speed*

where *speed* is the default tape speed in inches-per-second (ips). This default tape speed, if set, must match one of the defined nodes. Tape speed is dependent upon the tape drive hardware.

For example, to set the default tape speed to 50 ips, enter as superuser at the root prompt:

```
# tpconfig se d s 50
```

This indicates to the tape system to select a node with a tape speed of 50 ips if the mount request does not specify otherwise.

To unset a previously set default tape speed so that there is no default tape speed, use the command:

```
tpconfig set no_default speed
```

You can also enter as superuser at the root prompt:

```
# tpconfig se n s
```

After unsetting the default tape speed, if a tape system user does not specify a tape speed with their mount request, tape speed is not a criterion the tape system uses in selecting a node. Under this circumstance, a node of any speed can be selected by the tape system in response to the mount request.

Setting default labeled-tape file access permissions

You can set default tape file access for labeled-tape files. Unless otherwise specified by the user through the setting of tape attributes with the `tpattr` command and *before* writing the file to the tape, the tape system creates all files on a tape with the default file access permissions.

The command to set default tape file access for labeled-tape files is

```
tpconfig set default file_access {r|w|rw|none}
```

where any one of the following file access permissions must be specified:

`r`	Read only
`w`	Write only
`rw`	Read and write
`none`	No access permissions for anyone but the owner and the superuser

For example, to set default tape file access so that no one but the owner and the superuser can access newly created files (unless otherwise set by the user before creating the file), enter the following as superuser at the root prompt:

```
# tpconfig se d fi n
```

Adding and deleting label types

6

This chapter describes how to

- Add a new label type
- Change the name of a label type
- Delete a label type
- Show currently defined label types

which involves altering the `tpconfig` database.

The `tpconfig` database can only be altered by the superuser or a member of the group *tapeadm*, and, in ConvexOS/Secure, by those who have tapeadm authorization.

Viewing current label types

The command to view current label types is:

```
tpconfig show labels
```

Example 15 shows the (recommended) defined label types and daemons on a tape system.

Example 15 `tpconfig show labels` command output

```
% tpconfig sh l
Database is read-only.
Labels:
   nl daemon: /usr/lib/tape/nldaemon
   ibm daemon: /usr/lib/tape/ibmdaemon
   ansi daemon: /usr/lib/tape/ansidaemon
% ▮
```

Adding a new label type

The command syntax to add a label type to the tape system is

`tpconfig add label` *label_type label_daemon*

which has the following options:

label_type One of the following (recommended) label types for the specified label daemon:

- `ansi` For `ansidaemon`, which services ANSI-labeled tapes
- `ibm` For `ibmdaemon`, which services IBM-labeled tapes
- `nl` For `nldaemon`, which services unlabeled tape sets

label_daemon Path name of the label daemon whose type is being defined. Currently, there are three label daemons:

- /usr/lib/tape/ansidaemon for servicing ANSI-labeled tapes
- /usr/lib/tape/ibmdaemon for servicing IBM-labeled tapes
- /usr/lib/tape/nldaemon for servicing unlabeled tape sets

In the following example, the label type "ibm" is added to the tape system and is assigned to /usr/lib/tape/ibmdaemon. This must be done as superuser at the root prompt.

```
# tpconfig a l ibm /usr/lib/tape/ibmdaemon
```

Deleting a label type

The command syntax to delete a label type from the tape system is

`tpconfig delete label` *label_type*

where *label_type* is the label type you want to delete. These three label types are available:

`ansi` For `ansidaemon`, which services ANSI-labeled tapes

`ibm` For `ibmdaemon`, which services IBM-labeled tapes

`nl` For `nldaemon`, which services unlabeled tape sets

In the following example, the label type "ibm" is deleted from the tape system. This must be done as superuser at the root prompt.

```
# tpconfig d l ibm
```

Changing the name of a currently defined label type

To change the name of a label type, first delete the current label type with the `tpconfig delete label` command (refer to the section, "Deleting a label type," on page 54), and then add it to the tape system as the new type with the `tpconfig add label` command (refer to the section, "Adding a new label type," on page 53).

Setting tape system access 7

This chapter describes how to

- Configure access permissions to tape drives
- Indicate to the tape system those groups and users who can allocate a tape drive without specifying a symbolic link name and volume serial number (VSN)
- Indicate to the tape system those groups and users who can bypass tape label permissions
- Configure the tape system so that no one can mount unlabeled tapes
- Configure the tape system so that no one can create an unrestricted volume header

which involves altering the `tpconfig` database.

The `tpconfig` database can only be altered by the superuser or a member of the group *tapeadm*, and, in ConvexOS/Secure, by those who have tapeadm authorization.

Specifying who can access a tape drive

With the `tpconfig` utility you can indicate individual users and groups who can access a tape drive. This is done through three variations of the `tpconfig` utility's `access_drive` command:

- `tpconfig add access_drive`
- `tpconfig set access_drive`
- `tpconfig delete access_drive`

The `access_drive` command creates or alters an *access drive list*, which is consulted by the tape system before tape drives can be reserved. The default is that no one can access a drive.

The `tpconfig` database can only be altered by the superuser or a member of the group *tapeadm*, and, in ConvexOS/Secure, by those who have tapeadm authorization.

Adding to or creating an access drive list

The command to create or add entries to an access drive list is:

```
tpconfig add access_drive
```

Its syntax depends on whether you are adding users or groups to an access drive list. You cannot add both users and groups to the list in a single invocation. If you have a list of groups and a list of users to add to an access drive list, you must do so in two separate invocations.

Adding groups to an access drive list

To add groups to an access drive list, use the command syntax

```
tpconfig add access_drive group_set group_list type:unit
```

which requires that you supply the following information:

group_list A list of group IDs (GIDs) and valid group names, or an asterisk. The list can be a combination of both GIDs and group names.

Each GID must be enclosed in brackets. Separate all items in the list with spaces. To specify access for all groups, use an asterisk enclosed in double quotes ("*") in place of *group_list*.

If a group name does not exist in the /etc/group file, it is not used in the replacement, and an error message is displayed.

If a GID does not exist in the /etc/group file, it is accepted without an error and appears untranslated in the access drive list.

type:unit The drive type and unit number of the tape drive whose access drive list you are adding to.

The following example shows how a list of groups is added to the access drive list for the tc:1 drive.

```
# tpconfig a ac g [01] [50] tapeop tc:1
```

Adding users to an access drive list

The command syntax to add entries for individual users to an access drive list is

`tpconfig add access_drive user_set` *user_list type*:*unit*

which requires that you supply the following information:

user_list — A list of user IDs (UIDs) and valid user names, or an asterisk. The list can be a combination of both UIDs and user names.

Each UID must be enclosed in brackets. Separate all items in the list with spaces. To specify access for all users, use an asterisk enclosed in double quotes ("*") in place of *user_list*.

If a user name does not exist in the /etc/passwd file, it is not used in the replacement, and an error message is displayed.

If a UID does not exist in the /etc/passwd file, it is accepted without an error and appears untranslated in the access drive list.

type:*unit* — The drive type and unit number of the tape drive whose access drive list you are adding to.

The following example shows how a list of users is added to the access drive list for the tc:1 drive.

```
# tpconfig a ac u [2889] [1670] chris pat tc:1
```

Replacing an access drive list

The command syntax to replace entries in an existing access drive list is:

```
tpconfig set access_drive
```

Its syntax depends on whether you are replacing users or groups in an access drive list. You cannot replace both users and groups in an access drive list in a single invocation. If you have lists of groups and users to replace the current entries in an access drive list, you must do so in two separate invocations.

Replacing all groups in an access drive list

The command to replace entries for groups who are permitted to access a drive is

`tpconfig set access_drive group_set` *group_list type:unit*

which requires that you supply the following information:

group_list A list of group IDs (GIDs) and valid group names, or an asterisk. The list can be a combination of both GIDs and group names.

Each GID must be enclosed in brackets. Separate all items in the list with spaces. To specify access for all groups, use an asterisk enclosed in double quotes ("*") in place of *group_list*.

If a group name does not exist in the /etc/group file, it is not used in the replacement, and an error message is displayed.

If a GID does not exist in the /etc/group file, it is accepted without an error, but is not converted in the `tpconfig` database to a corresponding group name.

type:unit The drive type and unit number of the tape drive whose access you are setting.

The following example shows how to replace the current list of groups in an existing access drive list for the tc:1 drive with a new group list.

```
# tpconfig se ac g [01] [50] tapeop tc:1
```

Replacing all individual users in an access drive list

The command syntax to replace entries for individual users in an access drive list is:

`tpconfig set access_drive user_set` *user_list type:unit*

which requires that you supply the following information:

user_list — A list of user IDs (UIDs) and valid user names, or an asterisk. The list can be a combination of both UIDs and user names.

Each UID must be enclosed in brackets. Separate all items in the list with spaces. To specify access for all users, use an asterisk enclosed in double quotes (`"*"`) in place of *user_list*.

If a user name does not exist in the /etc/passwd file, it is not used in the replacement, and an error message is displayed.

If a UID does not exist in the /etc/passwd file, it is accepted without an error, but is not converted in the `tpconfig` database to a corresponding user name.

type:unit — The drive type and unit number of the tape drive whose access you are setting.

The following example shows how to replace the current list of users in an existing access drive list for the tc:1 drive with a new user list.

```
# tpconfig se ac u [2889] [1670] chris pat tc:1
```

Deleting from an access drive list

The command syntax to delete entries from an access drive list is:

```
tpconfig delete access_drive
```

Its syntax depends on whether you are deleting users or groups from an access drive list. You cannot delete entries for both users and groups from the list in a single invocation. If you have a list of groups and a list of users to delete from an access drive list, you must do so in two separate invocations.

Deleting groups from an access drive list

To delete entries for groups from an access drive list, use the command syntax:

```
tpconfig delete access_drive group_set group_list type:unit
```

which requires that you supply the following information:

group_list A list of group IDs (GIDs) and valid group names, or an asterisk. The list can be a combination of both GIDs and group names.

Each GID must be enclosed in brackets. Separate all items in the list with spaces. To delete access for all groups, use an asterisk enclosed in double quotes ("*") in place of *group_list*.

If a group name does not exist in the /etc/group file, an error message is displayed.

If a GID does not exist in the /etc/group file, it is accepted without an error and appears untranslated in the access drive list.

type:unit The drive type and unit number of the tape drive whose access drive list you are deleting from.

The following example shows how a list of groups is deleted from the access drive list for the tc:1 drive.

```
# tpconfig d ac g [01] [50] tapeop tc:1
```

Deleting users from an access drive list

The command to delete individual users from an access drive list is

`tpconfig delete access_drive user_set` *user_list type:unit*

which requires that you supply the following information:

user_list A list of user IDs (UIDs) and valid user names, or an asterisk. The list can be a combination of both UIDs and user names.

Each UID must be enclosed in brackets. Separate all items in the list with spaces. To specify access for all users, use an asterisk enclosed in double quotes (`"*"`) in place of *user_list*.

If a user name does not exist in the /etc/passwd file, an error message is displayed.

If a UID does not exist in the /etc/passwd file, there is no notification of an error.

type:unit The drive type and unit number of the tape drive whose access drive list you are deleting from.

The following example shows how a list of users is deleted from the access drive list for the tc:1 drive.

```
tpconfig d ac u [2889] [1670] chris pat tc:1
```

Specifying who can allocate a tape drive without specifying a symbolic link name and a VSN

With the `tpconfig` utility you can indicate individual users and groups who can access a tape drive without specifying a symbolic link name and a volume serial number (VSN). This is done through three variations of the `tpconfig` utility's `allocate_drive` command:

- `tpconfig add allocate_drive`
- `tpconfig set allocate_drive`
- `tpconfig delete allocate_drive`

The `allocate_drive` command creates or alters an *allocate drive list*, which is consulted by the tape system before tape drives can be reserved. The default is that no one can allocate a drive without specifying a symbolic link name and a VSN.

Adding to or creating an allocate drive list

The command to create or add to an allocate drive list is:

```
tpconfig add allocate_drive
```

Its syntax depends on whether you are adding users or groups to an allocate drive list. You cannot add both users and groups to the list in a single invocation. If you have a list of groups and a list of users to add to an allocate drive list, you must do so in two separate invocations.

Adding groups to an allocate drive list

To add groups to an allocate drive list, use the command

`tpconfig add allocate_drive group_set` *group_list type:unit*

which requires that you supply the following information:

group_list A list of group IDs (GIDs) and valid group names, or an asterisk. The list can be a combination of both GIDs and group names.

Each GID must be enclosed in brackets. Separate all items in the list with spaces. To specify unrestricted allocation for all groups, use an asterisk enclosed in double quotes ("*") in place of *group_list*.

If a group name does not exist in the /etc/group file, it is not used in the replacement, and an error message is displayed.

If a GID does not exist in the /etc/group file, it is accepted without an error and appears untranslated in the allocate drive list.

type:unit The drive type and unit number of the tape drive whose allocate drive list you are adding to.

The following example shows how a list of groups is added to the allocate drive list for the tc:1 drive.

```
# tpconfig a al g [01] [50] tapeop tc:1
```

Adding users to an allocate drive list

The command syntax to add individual users to an allocate drive list is

`tpconfig add allocate_drive user_set` *user_list type:unit*

which requires that you supply the following information:

user_list A list of user IDs (UIDs) and valid user names, or an asterisk. The list can be a combination of both UIDs and user names.

Each UID must be enclosed in brackets. Separate all items in the list with spaces. To specify unrestricted allocation for all users, use an asterisk enclosed in double quotes ("*") in place of *user_list*.

If a user name does not exist in the /etc/passwd file, it is not used in the replacement, and an error message is displayed.

If a UID does not exist in the /etc/passwd file, it is accepted without an error and appears untranslated in the allocate drive list.

type:unit The drive type and unit number of the tape drive whose allocate drive list you are adding to.

The following example shows how a list of users is added to the allocate drive list for the tc:1 drive.

```
# tpconfig a al u [2889] [1670] chris pat tc:1
```

Replacing an allocate drive list

The command to replace the entries in an existing allocate drive list is:

```
tpconfig set allocate_drive
```

Its syntax depends on whether you are replacing users or groups in an allocate drive list. You cannot replace both users and groups in an allocate drive list in a single invocation. If you have lists of groups and users to replace the current entries in an allocate drive list, you must do so in two separate invocations.

Replacing all groups in an allocate drive list

The command syntax to replace groups who are permitted to allocate a drive is

`tpconfig set allocate_drive group_set` *group_list type:unit*

which requires that you supply the following information:

group_list A list of group IDs (GIDs) and valid group names, or an asterisk. The list can be a combination of both GIDs and group names.

Each GID must be enclosed in brackets. Separate all items in the list with spaces. To specify unrestricted allocation for all groups, use an asterisk enclosed in double quotes (`"*"`) in place of *group_list*.

If a group name does not exist in the /etc/group file, it is not used in the replacement, and an error message is displayed.

If a GID does not exist in the /etc/group file, it is accepted without an error, but is not converted in the `tpconfig` database to a corresponding group name.

type:unit The drive type and unit number of the tape drive whose allocate drive list you are setting.

The following example shows how to replace the current list of groups in an existing allocate drive list for the tc:1 drive with a new group list. This command must be entered as superuser at the root prompt.

```
# tpconfig se al g [01] [50] tapeop tc:1
```

Replacing all individual users in an allocate drive list

The command syntax to replace entries for individual users in an allocate drive list is

`tpconfig set allocate_drive user_set` *user_list type:unit*

which requires that you supply the following information:

user_list A list of user IDs (UIDs) and valid user names, or an asterisk. The list can be a combination of both UIDs and user names.

Each UID must be enclosed in brackets. Separate all items in the list with spaces. To specify unrestricted allocation for all users, use an asterisk enclosed in double quotes (`"*"`) in place of *user_list*.

If a user name does not exist in the /etc/passwd file, it is not used in the replacement, and an error message is displayed.

If a UID does not exist in the /etc/passwd file, it is accepted without an error, but is not converted in the `tpconfig` database to a corresponding user name.

type:unit The drive type and unit number of the tape drive whose allocate drive list you are setting.

The following example shows how to replace the current list of users in an existing allocate drive list for the tc:1 drive with a new user list. This command must be entered as superuser at the root prompt.

```
# tpconfig se al u [2889] [1670] chris pat tc:1
```

Deleting from an allocate drive list

The command to delete entries from an allocate drive list is:

```
tpconfig delete allocate_drive
```

Its syntax depends on whether you are deleting users or groups from an allocate drive list. You cannot delete entries for both users and groups from the list in a single invocation. If you have a list of groups and a list of users to delete from an allocate drive list, you must do so in two separate invocations.

Deleting groups from an allocate drive list

To delete entries for groups from an allocate drive list, use the command syntax

`tpconfig delete allocate_drive group_set` *group_list type:unit*

which requires that you supply the following information:

group_list A list of group IDs (GIDs) and valid group names, or an asterisk. The list can be a combination of both GIDs and group names.

Each GID must be enclosed in brackets. Separate all items in the list with spaces. To delete unrestricted allocation for all groups, use an asterisk enclosed in double quotes (`"*"`) in place of *group_list*.

If a group name does not exist in the /etc/group file, an error message is displayed.

If a GID does not exist in the /etc/group file, notification of an error is not displayed.

type:unit The drive type and unit number of the tape drive whose allocate drive list you are deleting from.

The following example shows how a list of groups is deleted from the allocate drive list for the tc:1 drive.

```
# tpconfig d al g [01] [50] tapeop tc:1
```

Deleting users from an allocate drive list

The command syntax to delete entries for individual users from an allocate drive list is

`tpconfig delete allocate_drive user_set` *user_list type:unit*

which requires that you supply the following information:

user_list A list of user IDs (UIDs) and valid user names, or an asterisk. The list can be a combination of both UIDs and user names.

Each UID must be enclosed in brackets. Separate all items in the list with spaces. To delete unrestricted allocation for all users, use an asterisk enclosed in double quotes (`"*"`) in place of *user_list*.

If a user name does not exist in the /etc/passwd file, an error message is displayed.

If a UID does not exist in the /etc/passwd file, notification of an error is not displayed.

type:unit The drive type and unit number of the tape drive whose allocate drive list you are deleting from.

The following example shows how a list of users is deleted from the allocate drive list for the tc:1 drive.

```
# tpconfig d al u [2889] [1670] chris pat tc:1
```

Specifying who can bypass label permissions

With the `tpconfig` utility you can indicate individual users and groups who can bypass restrictions on labeled tapes. This is done through three variations of the `tpconfig` utility's `bypass_labels` command:

- `tpconfig add bypass_labels`
- `tpconfig set bypass_labels`
- `tpconfig delete bypass_labels`

The `bypass_labels` command creates or alters a *bypass labels list*, which is consulted by the tape system before tape label permissions can be bypassed. The default is that no one can bypass label permissions except for the superuser.

Caution

It is a security risk to allow labels to be bypassed because it permits the user to ignore access restrictions imposed by the tape labels. This could result in the accidental or deliberate destruction or alteration of data. Under ConvexOS/Secure, only trusted users should have bypass access.

Adding to or creating a bypass labels list

The command to create or add to a bypass labels list is:

```
tpconfig add bypass_labels
```

Its syntax depends on whether you are adding users or groups to a bypass labels list. You cannot add both users and groups to the list in a single invocation. If you have a list of groups and a list of users to add to a bypass labels list, you must do so in two separate invocations.

Adding groups to a bypass labels list

To add groups to a bypass labels list, use the command

`tpconfig add bypass_labels group_set` *group_list type:unit*

which requires that you supply the following information:

group_list A list of group IDs (GIDs) and valid group names, or an asterisk. The list can be a combination of both GIDs and group names.

Each GID must be enclosed in brackets. Separate all items in the list with spaces. To specify label bypass permission for all groups, use an asterisk enclosed in double quotes (`"*"`) in place of *group_list*.

If a group name does not exist in the /etc/group file, it is not used in the replacement, and an error message is displayed.

If a GID does not exist in the /etc/group file, it is accepted without an error and appears untranslated in the bypass labels list.

type:unit The drive type and unit number of the tape drive whose bypass labels list you are adding to.

The following example shows how a list of groups is added to the bypass labels list for the tc:1 drive.

```
# tpconfig a b g [01] [50] tapeop tc:1
```

Adding users to a bypass labels list

The command to add individual users to a bypass labels list is

`tpconfig add bypass_labels user_set` *user_list type:unit*

which requires that you supply the following information:

user_list A list of user IDs (UIDs) and valid user names, or an asterisk. The list can be a combination of both UIDs and user names.

Each UID must be enclosed in brackets. Separate all items in the list with spaces. To specify label bypass permission for all users, use an asterisk enclosed in double quotes ("*") in place of *user_list*.

If a user name does not exist in the /etc/passwd file, it is not used in the replacement, and an error message is displayed.

If a UID does not exist in the /etc/passwd file, it is accepted without an error and appears untranslated in the bypass labels list.

type:unit The drive type and unit number of the tape drive whose bypass labels list you are adding to.

The following example shows how a list of users is added to the bypass labels list for the tc:1 drive.

```
# tpconfig a b u [2889] [1670] chris pat tc:1
```

Replacing a bypass labels list

The command to replace the entries in an existing bypass labels list is:

```
tpconfig set bypass_labels
```

Its syntax depends on whether you are replacing users or groups in a bypass labels list. You cannot replace both users and groups in a bypass labels list in a single invocation. If you have lists of groups and users to replace the current entries in a bypass labels list, you must do so in two separate invocations.

Replacing all groups in a bypass labels list

The command syntax to replace entries for groups who are permitted to allocate a drive is

```
tpconfig set bypass_labels group_set group_list type:unit
```

which requires that you supply the following information:

group_list A list of group IDs (GIDs) and valid group names, or an asterisk. The list can be a combination of both GIDs and group names.

Each GID must be enclosed in brackets. Separate all items in the list with spaces. To specify label bypass permission for all groups, use an asterisk enclosed in double quotes ("*") in place of *group_list*.

If a group name does not exist in the /etc/group file, it is not used in the replacement, and an error message is displayed.

If a GID does not exist in the /etc/group file, it is accepted without an error, but is not converted in the `tpconfig` database to a corresponding group name.

type:unit The drive type and unit number of the tape drive whose bypass labels list you are setting.

The following example shows how to replace the current list of groups in an existing bypass labels list for the tc:1 drive with a new group list. This command must be entered as superuser at the root prompt.

```
# tpconfig se b g [01] [50] tapeop tc:1
```

Replacing all individual users in a bypass labels list

The command syntax to replace all entries for individual users in a bypass labels list is

`tpconfig set bypass_labels user_set` *user_list type:unit*

which requires that you supply the following information:

user_list A list of user IDs (UIDs) and valid user names, or an asterisk. The list can be a combination of both UIDs and user names.

Each UID must be enclosed in brackets. Separate all items in the list with spaces. To specify label bypass permission for all users, use an asterisk enclosed in double quotes (`"*"`) in place of *user_list*.

If a user name does not exist in the /etc/passwd file, it is not used in the replacement, and an error message is displayed.

If a UID does not exist in the /etc/passwd file, it is accepted without an error, but is not converted in the `tpconfig` database to a corresponding user name.

type:unit The drive type and unit number of the tape drive whose bypass labels list you are setting.

The following example shows how to replace the current list of users in an existing bypass labels list for the tc:1 drive with a new user list. This command must be entered as superuser at the root prompt.

```
# tpconfig se b u [2889] [1670] chris pat tc:1
```

Deleting from a bypass labels list

The command to delete entries from a bypass labels list is:

```
tpconfig delete bypass_labels
```

Its syntax depends on whether you are deleting users or groups from a bypass labels list. You cannot delete both users and groups from the list in a single invocation. If you have a list of groups and a list of users to delete from a bypass labels list, you must do so in two separate invocations.

Deleting groups from a bypass labels list

To delete groups from a bypass labels list, use the command syntax

```
tpconfig delete bypass_labels group_set group_list type:unit
```

which requires that you supply the following information:

group_list A list of group IDs (GIDs) and valid group names, or an asterisk. The list can be a combination of both GIDs and group names.

Each GID must be enclosed in brackets. Separate all items in the list with spaces. To delete label bypass permission for all groups, use an asterisk enclosed in double quotes ("*") in place of *group_list*.

If a group name does not exist in the /etc/group file, an error message is displayed.

If a GID does not exist in the /etc/group file, notification of an error is not displayed.

type:unit The drive type and unit number of the tape drive whose bypass labels list you are deleting from.

The following example shows how a list of groups is deleted from the bypass labels list for the tc:1 drive.

```
# tpconfig d b g [01] [50] tapeop tc:1
```

Deleting users from a bypass labels list

The command syntax to delete entries for individual users from a bypass labels list is

`tpconfig delete bypass_labels user_set` *user_list type:unit*

which requires that you supply the following information:

user_list A list of user IDs (UIDs) and valid user names, or an asterisk. The list can be a combination of both UIDs and user names.

Each UID must be enclosed in brackets. Separate all items in the list with spaces. To delete label bypass permission for all users, use an asterisk enclosed in double quotes (`"*"`) in place of *user_list*.

If a user name does not exist in the /etc/passwd file, an error message is displayed.

If a UID does not exist in the /etc/passwd file, notification of an error is not displayed.

type:unit The drive type and unit number of the tape drive whose bypass labels list you are deleting from.

The following example shows how a list of users is deleted from the bypass labels list for the tc:1 drive.

```
# tpconfig d b u [2889] [1670] chris pat tc:1
```

Adding extra security measures

It may be necessary to add an extra measure of security to the tape system. You can do this at any time using tape system configuration commands. The procedures in this section ensure that only labeled tapes can be used with the tape system, and only the owners of tapes can mount and access the files on them.

These procedures alter the `tpconfig` database. The `tpconfig` database can only be altered by the superuser or a member of the group *tapeadm*, and, in ConvexOS/Secure, by those who have tapeadm authorization.

Restricting use to labeled tapes only

The tape system is delivered so that a user can mount a device for use with either a labeled or unlabeled tape.

The command syntax to restrict users to labeled tape is:

```
set labled_mode_only {enabled|disabled}
```

This command configures the tape system so that all users must use labeled tapes. For example, to restrict users to labeled tapes, enter:

```
# tpconfig se l e
```

Restricting use to secure volume headers only

The tape system is delivered so that any user can label a tape with either restricted access, in which case only the owner or root can mount the tape, or with world-accessibility, in which case anyone can mount the tape.

The command syntax to restrict users to restricted volume headers on labeled tapes is:

```
set restrict vol_headers {enabled|disabled}
```

This command configures the tape system so that all users must label their tapes with restricted access.

For example, to force users to restrict tape access enter:

```
# tpconfig se r v e
```

You can also set the tape system's file access default so that all files written to a labeled tape have restricted access, unless explicitly overridden by the user. For more information on this default, refer to Chapter 5, the section, "Setting default labeled-tape file access permissions," on page 50.

Configuring your system for tape operators

8

This chapter describes how to configure your system for use with tape operators, which includes:

- How to establish what tasks the tape operators are authorized to do
- What the default Operator Request management (`opreq`) screen configuration us and how to change it
- How to add users to the tape operator's group, *tapeop*
- How to enable `opreq` tape system queuing

For further information on tape system operational procedures, refer to the *ConvexOS Tape System Operator's Guide* (DSW-397).

opreq terminal and screen configurations

opreq configures its display to fit the characteristics of a wide variety of terminals and operator preferences. When opreq starts up, it does the following:

- Determines the operator's terminal type by examining the TERM environment variable.
- Determines the characteristics of the operator's terminal type by examining the TERMCAP environment variable. These are the possibilities:
  - If TERMCAP is empty, opreq uses the characteristics specified for the terminal type (obtained from TERM) in the /etc/termcap file.
  - If TERMCAP contains the path name of a file, opreq uses the terminal characteristics specified in that file.
  - If TERMCAP contains a list of terminal characteristics (instead of a file name), opreq uses the information from the variable that pertains to the terminal type.
- Determines how to configure the opreq screen by checking for a .opreqrc file in the operator's current directory. If one does not exist, opreq checks for a .opreqrc file in the operator's home directory. If one is not found there, opreq uses the /usr/lib/opreq/.opreqrc file to configure the operator's screen.

Setting TERM and TERMCAP environment variables for opreq

To change the display width of the opreq window, change the number of columns indicated in /etc/termcap file for the operator's terminal type and change the number of terminal columns with the stty command.

Refer to the stty(1) man page for more information about this command.

Default opreq screen configuration

Default values that control the window configuration for the entire tape system are stored in the file /usr/lib/opreq/.opreqrc. When opreq starts, it first looks in the operator's current directory, then in the operator's home directory for a .opreqrc file. If this file doesn't exist, opreq uses the system's default file /usr/lib/opreq/.opreqrc, which is maintained by the system manager.

opreq screen defaults that are delivered with the tape system

Example 17 shows the /usr/lib/opreq/.opreqrc file that is delivered with the tape system.

Example 16 /usr/lib/opreq/.opreqrc file that is delivered with the tape system

```
# The following is the default configuration for the opreq program.
# To change this default copy this file into you current directory
# or into your home directory, then change to the configuration you desire.
# The opreq program will first look in you current working directory
# (ie. "./.opreq"), then in your home directory (ie. "~/.opreq"), and
# finally, it will use this file.

#
# open a window at 0 0 (x, y) with a size of at most 200 200 (width, length)
#
window-open 0 0 200 200

#
# configure status of the last window open to display "ready" messages
#
configure-status ready remote

#
# configure type of the last window opened to display mount-tape,
# unmount-tape, and replace-tape messages
#
configure-type mount-tape unmount-tape replace-tape silo-enter

#
# configure title of the last window opened to display the uid, status,
# type, vsn, ring, drive, time-in, and comment fields.
#
configure-title uid status type vsn ring drive time-in comment
```

Changing `opreq` screen defaults

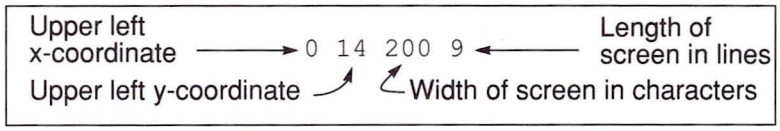
Example 17 shows a .opreqrc file that configures a two-screen display.

Example 17 .opreqrc file configuring a two-screen display

```
① window-open 0 14 200 9
② configure-status ready waiting
③ configure-type info auto-vol-rec
④ configure-title uid type vsn ring drive comment
⑤ window-open 0 0 200 14
⑥ configure-status ready waiting
⑦ configure-type mount-tape unmount-tape replace-tape silo-enter
```

In Example 17, the following lines of the .opreqrc file are:

① `window-open` opens an `opreq` window. The coordinates have the following significance:



② `configure-status` tells `opreq` to display only messages of a specified status.

In this example, `configure-status` tells `opreq` to only show those messages with a status of `ready` or `waiting`.

This command only affects the window opened in line ①.

③ `configure-type` tells `opreq` to display only messages of a specified type.

In this example, `configure-status` tells `opreq` to only show those messages of type `info` or `auto-vol-rec`, which do not require operator action.

This command only affects the window opened in line ①.

④ `configure-title` tells `opreq` to display certain titles in the title bar. Titles label columns that sort information about each message.

In this example, `configure-titles` tells `opreq` to show the following information about each message: `uid`, `type`, `vsn`, `ring`, `drive`, and `comment`.

This command only affects the window opened in line ①.

⑤ `window-open` opens a second `opreq` window.

⑥ `configure-status` tells `opreq` to display only messages of a specified status.

In this example, `configure-status` tells `opreq` to only show those messages with a status of `ready` or `waiting`.

This command only affects the window opened in line ⑤.

⑦ `configure-type` tells `opreq` to display only messages of a specified type.

In this example, `configure-type` tells `opreq` to only show those messages of types that require some type of operator action.

This command only affects the window opened in line ⑤.

Example 18 shows how the configuration from Example 17 looks when `opreq` is started.

Example 18 A two-screen display

```
+ UID-----STATUS--TYPE--------VSN------RING-DRIVE------------COMMENT---------- 2+
|                                                                              |
|          This window shows messages that are ready for service or            |
|          are waiting on resources. This window is in reverse video,          |
|          indicating that the command line only affects this window.          |
|                                                                              |
|          Lines ⑤ through ⑦ in Example 10 configure this window.              |
|                                                                              |
|          Titles in the title bar have not been explicitly set. The titles shown |
|          are ones that are set in the system default file /usr/lib/opreq/.opreqrc. |
|                                                                              |
+------------------------------------------------------------------------------+
+UID--TYPE---------VSN--------RING-DRIVE-----------COMMENT---------------- 1+
|          This window shows messages that give information only. The         |
|          messages in this window require no operator action. This window    |
|          is not highlighted, indicating that the command line does not      |
|          affect it.                                                         |
|                                                                             |
|          Lines ① through ④ in Example 10 configure this window.             |
+-----------------------------------------------------------------------------+
Command:█
```

Adding users to the tapeop group

The first step in establishing tape operators for your system is to add operators' login IDs to the tapeop group.

The group tapeop can have many of the same authorizations as superuser, depending on which operations you need them to do. At a minimum, users who are members of group tapeop can perform some restricted `opreq` commands, which are:

`select-cancel`	Cancel a mount request and, if necessary, send a message to the user
`select-done`	Complete a mount request
`select-work`	Assign a mount request to oneself, if multiple tape operators are working simultaneously

To add users to the tapeop group, modify the /etc/group file. The names you add to this file must be valid login names from the /etc/passwd file.

Following is the line from an /etc/group file that defines who the members of group tapeop are:

```
tapeop:*:29:abrams,kim,nelson
```

By adding valid login IDs to this line, you add the users to the tapeop group. They can then start up an `opreq` session and complete `opreq` requests.

Enabling opreq queuing

If you want mount and unmount requests to be handled by a tape operator through opreq, you must first enable queueing in the tape system. The command syntax to enable (or disable) opreq queuing is:

```
tpconfig set queuing {enable|disable}
```

To execute this command you must be superuser or be a member of the group *tapeadm*. Under ConvexOS/Secure, you must also have tapeadm authorization.

To enable queuing, enter the following:

```
# tpconfig se q e
```

The status of opreq queueing does change until there are no active tpmount requests on the system. tpmount requests are active until they are canceled either by operator intervention, by the system, or by a corresponding tpunmount command. Use the tpqueue command to show active tpmount requests.

After opreq queueing is enabled and the tape system reaches an idle state, subsequent user requests to mount, change, and unmount tapes are passed on to opreq. The tape operator then completes the request. Users cannot use the tpmount -q command to queue a tape mount request once tape system queuing is enabled.

Supporting an automatic cartridge loader (ACL)

9

This chapter describes how to:

- Add ACL drives and the ACL (stacker) daemon to the tape system
- Delete ACL drives and the ACL (stacker) daemon from the tape system

The procedures in this chapter require altering the `tpconfig` database. To alter the `tpconfig` database you must be superuser or be a member of the group *tapeadm*. Under ConvexOS/Secure, you must also have tapeadm authorization.

Adding an ACL to your system

There are two steps to adding an ACL to your tape system:

- Associating the ACL tape devices as nodes to the ACL tape drive
- Adding the ACL (stacker) daemon to the tape system

To add an ACL drive to your tape system, follow this procedure:

Step 1 Ensure that your ACL drive has been added to the tape system. Refer to Chapter 4, the section, "Adding 9-track, cartridge, and DAT drives," on page 30.

Step 2 Associate ACL tape devices as nodes to the ACL tape drive. This enables tape system users to access the ACL through the tape system, which controls the tape devices once they become nodes.

The command syntax to associate the ACL tape devices as nodes of the ACL drive is

`tpconfig add stacker` *path type*:*unit*

which requires you to supply the following information:

path	The full path name of the tape device being assigned as a node to the ACL drive
type:*unit*	The drive type and unit number of the ACL drive

This command must be executed for all ACL devices, if you have more than one.

The following example shows the tape device /dev/tc2stk being assigned as a node to the tc:2 drive, which in this case is an ACL drive:

```
# tpconfig a stacker /dev/tc2stk tc:2
```

Step 3 Add the ACL (stacker) daemon to the tape system. The command to do this is

`tpconfig add stackerdaemon` *path*

where *path* is the full path name to the ACL (stacker) daemon.

In the following example, /usr/lib/tape/stkdaemon is defined as the ACL (stacker) daemon in the tape system:

```
# tpconfig a stackerd /usr/lib/tape/stkdaemon
```

Showing ACL (stacker) daemon information

After the ACL (stacker) daemon has been added to the tape system, its location can be displayed. The command to show which ACL (stacker) daemon is currently defined in the tape system is

```
tpconfig show stackerdaemon
```

In Example 19, /usr/lib/tape/stkdaemon is shown as being the current ACL (stacker) daemon.

Example 19 Showing currently defined ACL (stacker) daemon location

```
% tpconfig sh stackerd
Stacker Daemon: /usr/lib/tape/stkdaemon
% ▮
```

Deleting an ACL from your system

There are two steps to deleting an ACL from your tape system:

- Deleting the entries in the `tpconfig` database for the ACL drives
- Deleting the ACL (stacker) daemon from the tape system

To delete an ACL drive from your tape system, follow this procedure:

Step 1 Delete the entries for the ACL drive from the `tpconfig` database. This prohibits tape system users from accessing the ACL through the tape system.

The command to delete an ACL drive from the tape system is

`tpconfig del stacker` *type*`:`*unit*

where *type*`:`*unit* is the drive type and unit number of the ACL drive you are deleting.

The following example shows an ACL drive named tc:2 being deleted from the tape system:

```
# tpconfig d stacker tc:2
```

Step 2 If you are deleting all ACL drives from your tape system, you should delete the ACL (stacker) daemon as well. The command to delete the ACL (stacker) daemon from your tape system is:

`tpconfig del stackerd`

In the following example, the ACL (stacker) daemon is deleted from the tape system:

```
# tpconfig d stackerd
```

Configuring CONVEX ACS 10

This chapter tells you how to configure your system for use with the StorageTek 4400 Automated Cartridge System (ACS). It addresses the following requirements:

- Setting your `SILOHOST` environment variable in order to use ACS commands
- Adding TLI drives to the ACS
- Mapping CONVEX tape devices to TLI drives

Setting your environment variable, SILOHOST

To use the following ACS commands:

- `silodismount`
- `siloeject`
- `siloenter`
- `silomount`
- `siloquery`

you must set your environment variable, `SILOHOST`, to the host name of the Sun workstation that controls ACS on your CONVEX system.

For example, if you use `csh`, and the host name of the Sun CPU that controls the ACS is "jupiter," enter

```
% setenv SILOHOST jupiter
```

If you use `sh` or `ksh`, enter

```
$ SILOHOST=jupiter; export SILOHOST
```

You should add the appropriate command to your .login or .profile file, depending on the shell you use.

Configuring the CONVEX ACS

This section describes how to prepare the CONVEX ACS for use. All drives and devices must first be defined in the tape system as explained in Chapter 4, "Adding tape drives and devices," on page 29.

ACS drive coordinates

ACS drive coordinates tell an ACS mechanical tape operator the position occupied by the tape drive in ACS silo. This enables the ACS to mount and unmount cartridges on the tape drive, as well as report its status. The ACS drive coordinates are four integers separated by commas as shown:

① ② ③ ④
`0,0,0,0`

Each integer in an ACS coordinate indicates:

① StorageTek unit number, if units are separated (ACS number)

② StorageTek unit number, if units are connected (LSM number)

③ StorageTek panel number (inside the unit)

④ StorageTek drive position number (on the panel)

Adding TLI drives to the ACS

To add a TLI drive to the ACS, you must specify that the TLI drive is associated with the ACS host. The command to associate a TLI drive with a ACS silo host is

`tpconfig set silohost` *hostname drive_string type:unit*

which requires that you supply the following information:

hostname The host name of the Sun CPU that controlling the ACS silo

drive_string The ACS drive coordinates of the TLI drive you are adding

type:unit The drive type and unit number of the TLI drive you are adding

This command involves altering the `tpconfig` database. To alter the `tpconfig` database you must be superuser or be a member of the group *tapeadm*. Under ConvexOS/Secure, you must also have tapeadm authorization.

An example of how to add the drives tc:0 and tc:1 to an ACS that is hosted by a Sun workstation named "sunburst" follows:

```
# tpconfig se silohost sunburst 0,0,10,0 tc:0
# tpconfig se silohost sunburst 0,0,10,2 tc:1
```

Mapping tape devices to ACS tape drives

The file /usr/lib/tape/silodrivelist maps a CONVEX tape device to an ACS drive coordinate, which allows the tape device to be used to access the associated tape drive. The silodrivelist file is a simple ASCII file you edit manually and must create before using ACS tape commands.

The format of the /usr/lib/tape/silodrivelist file is:

- Only one entry per line is allowed.
- Each entry is a CONVEX tape drive or device name followed by the ACS drive coordinate. A name and a coordinate are separated by spaces or tabs.
- No comments are allowed.

An example of a /usr/lib/tape/silodrivelist file is as follows:

```
tc:0                0,0,10,0
tc:1                0,0,10,2
/dev/rtc0           0,0,10,0
/dev/rtc0n          0,0,10,0
/dev/rtc0nu         0,0,10,0
/dev/rtc0u          0,0,10,0
/dev/rtc1           0,0,10,2
/dev/rtc1n          0,0,10,2
/dev/rtc1nu         0,0,10,2
/dev/rtc1u          0,0,10,2
/dev/tc0            0,0,10,0
/dev/tc0n           0,0,10,0
/dev/tc0nu          0,0,10,0
/dev/tc0u           0,0,10,0
/dev/tc1            0,0,10,2
/dev/tc1n           0,0,10,2
/dev/tc1nu          0,0,10,2
/dev/tc1u           0,0,10,2
```

Operating the ACS silo

11

The StorageTek Automated Cartridge System (ACS) silo is a cluster of one to sixteen Library Storage Modules (LSMs), each of which can contain nearly six thousand 3480 tape cartridges; two to 16 StorageTek 4480 tape cartridge drives; and a robot arm for loading and unloading tapes in the system, and mounting and unmounting cartridges on the tape drives.

This chapter describes how to do the following tasks with through the ACS:

- Enter a tape into the silo
- Eject a tape from the silo
- Manually mount a tape cartridge on a silo drive
- Manually dismount a tape cartridge from a silo drive
- Report the status of a silo component

For more information on these utilities, refer to the silomount(8), silodismount(8), siloenter(8), siloeject(8), and siloquery(8) man pages. These man pages are also included in the book *CONVEX Man Pages for System Managers* (DSW-333).

Reporting ACS component statuses

You can determine the status of any ACS component by using the `siloquery` utility. The statuses of the following ACS components can be queried:

- ACS
- Controlled Access Ports (CAPs)
- Library Management Unit (LMU) ports
- Library Storage Modules (LSMs)
- Mounts for specific cartridges
- Requests
- Server
- Specific cartridges that are stored
- Tape drives

The syntax for the `siloquery` utility is

`siloquery` *command*

where *command* can be any of the commands described in Table 6.

Table 6 `siloquery` utility and its commands

Command	Description
`acs` [*acsID...*]	Report the status of all ACS silos controlled by the server, unless one or more ACS IDs are specified. *acsID* is the ID of a specific ACS silo whose status is desired. If one or more ACS IDs are specified, only information about those ACS silos are reported.
`cap` [*capID...*]	Report the status of all CAPs controlled by the server, unless one or more CAP IDs are specified. *capID* is the ID of a specific CAP whose status is desired. If one or more CAP IDs are specified, only information about those CAPs are reported.
`drive` [*driveID...*]	Report the status of all tape drives in the ACS silo controlled by the server, unless one or more tape drive IDs are specified. *driveID* is the ID of a specific tape drive whose status is desired. If one or more drive IDs are specified, only information about those tape drives are reported.

Table 6 `siloquery` utility and its commands

Command	Description
`lsm [`*lsmID...*`]`	Report the status of all LSMs controlled by the server, unless one or more LSM IDs are specified. *lsmID* is the ID of a specific LSM whose status is desired. If one or more LSM IDs are specified, only information about those LSMs are reported.
`mount` *VSN* `[`*VSN...*`]`	Report the status of all tape drives and cartridges controlled by the server, unless one or more VSNs are specified. *VSN* is the VSN of a specific cartridge whose status is desired. If one or more VSNs are specified, the statuses of all drives are still reported, but only the statuses of the specified cartridges are reported.
`port [`*portID...*`]`	Report the status of all LMU ports controlled by the server, unless one or more port IDs are specified. *portID* is the ID of a specific LMU port, which is a serial communication line from the server to an LMU. If one or more port IDs are specified, only statuses about those ports are reported.
`request [`*requestID...*`]`	Report the status of all server requests currently being executed, unless one or more request IDs are specified. *requestID* is the ID number of a server request whose status is desired. If one or more request ID are specified, only information about those requests are reported.
`server`	Report the status of the StorageTek Sun server.
`volume [`*VSN...*`]`	Report the VSN and location of every cartridge contained in the ACS silo, unless one or more VSNs are specified. *VSN* is the VSN of a specific cartridge whose status is desired. If one or more VSNs are specified, only information about those VSNs are reported.

Reporting ACS status

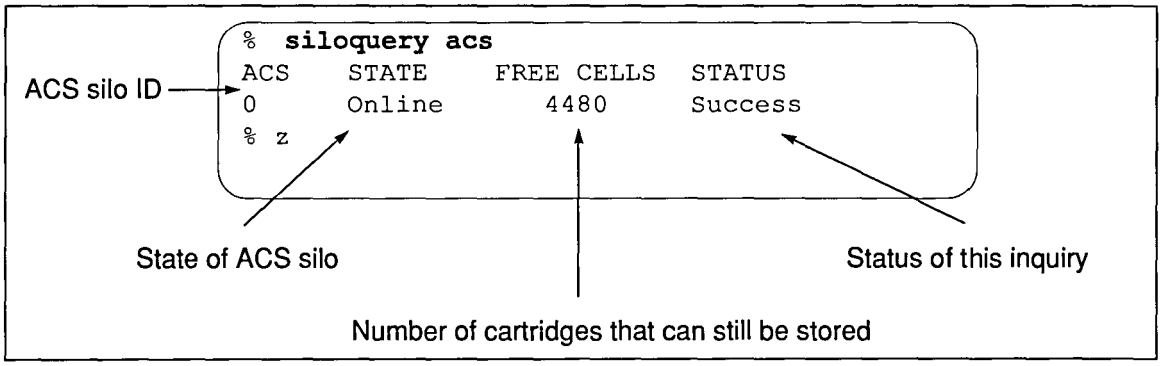
To report status information about an ACS silo, use the `siloquery acs` command. `siloquery acs` has the following syntax:

`siloquery acs [`*acsID*`...]`

acsID is an integer representing the ACS silo's identification number. The first ACS silo that is installed is typically numbered "0," the second "1," the third "2," and so on. If you do not specify an ACS ID, `siloquery acs` reports the status of all ACS silos in your system.

For an example of how `siloquery acs` reports statuses, refer to Example 20.

Example 20 Reporting general ACS silo status



Reporting CAP status

To report status information about an ACS silo CAP (Control Access Port), use the `siloquery cap` command.

`siloquery cap` has the following syntax:

`siloquery cap [`*capID*`...]`

capID is an the ACS silo's CAP ID. The first CAP in the first ACS silo is generally numbered "0,0", the second CAP in the first ACS silo is numbered "0,1", the first CAP in the second ACS silo (if there is one) is numbered "1,0", and so on. If you do not specify an CAP ID, `siloquery cap` reports the status of all CAPs in your system.

For an example of how `siloquery cap` reports statuses, refer to Example 21.

Example 21 Reporting ACS silo CAP status

```
% siloquery cap
CAP     STATUS
0,0     CAP available
%
```

CAP ID

Whether the CAP is available or unavailable for entering and ejecting cartridges

Reporting tape drive status

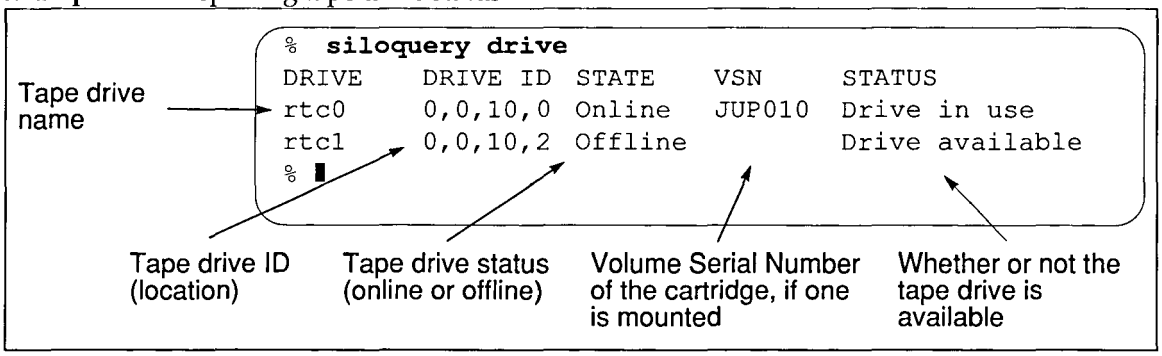
To report status information about an ACS tape drive, use the `siloquery drive` command. `siloquery drive` has the following syntax:

`siloquery drive [`*`driveID...`*`]`

driveID is the tape drive's ID as specified in the ioconfig file. If you do not specify a tape drive ID, `siloquery drive` reports the status of all tape drives in the ACS. If you need more information about finding out drive names or displaying information about drives, refer to Chapter 4, "Adding tape drives and devices," on page 29.

For an example of how `siloquery drive` reports the status of tape drives, refer to Example 22.

Example 22 Reporting tape drive status



Reporting Library Storage Module status

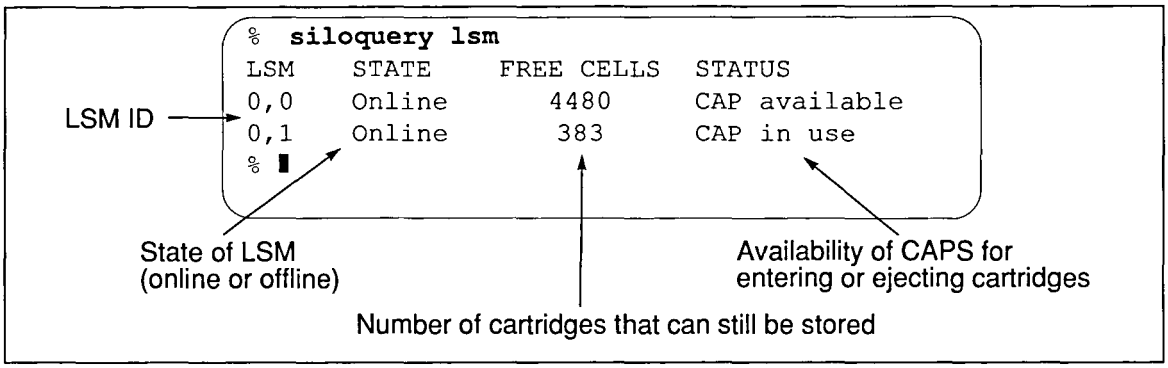
To report status information about a Library Storage Module (LSM), use the `siloquery lsm` command. `siloquery lsm` has the following syntax:

`siloquery lsm [`*lsmID...*`]`

lsmID is one or more LSM IDs. The first LSM in the first ACS is generally numbered "0,0", the second LSM in the first ACS is numbered "0,1", the first LSM in the second ACS (if there is one) is numbered "1,0", and so on. If you do not specify an LSM ID, `siloquery lsm` reports the status of all LSMs in your system.

For an example of how `siloquery lsm` reports the status of LSMs, refer to Example 23.

Example 23 Reporting LSM status



Reporting mount and tape drive status

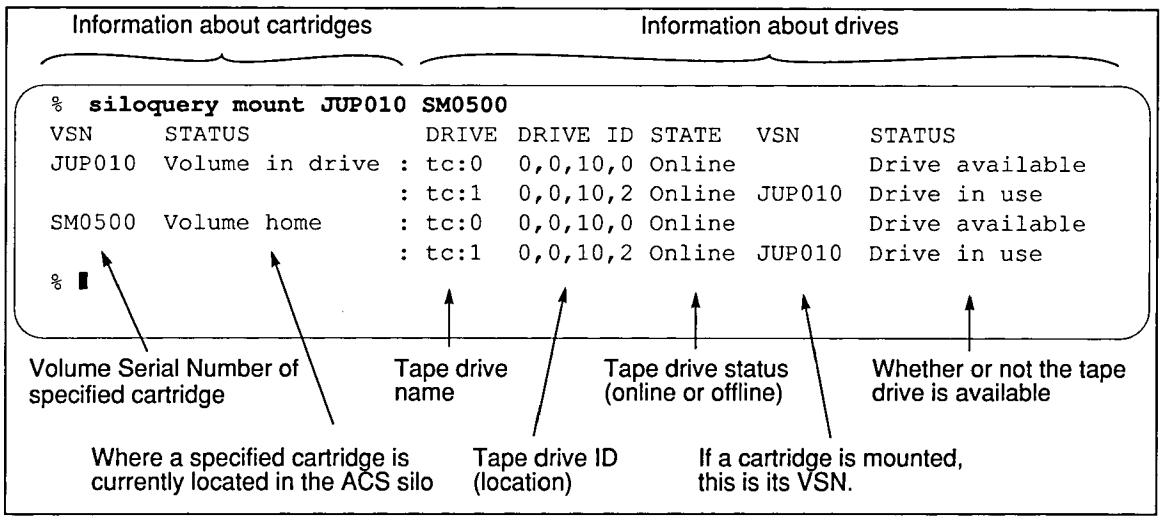
To report status information about a cartridge that is mounted and all the tape drives, use the `siloquery mount` command. `siloquery mount` has the following syntax:

`siloquery mount` *VSN* `[`*VSN...*`]`

VSN is the Volume Serial Number of a cartridge you want queried. You can specify one or more VSNs, but you must specify at least one.

For an example of how `siloquery mount` reports mount and tape drive status, refer to Example 24.

Example 24 Reporting mount and tape drive status



Reporting port status

A port is a serial connection line from the ACS server to a Library Management Unit (LMU).

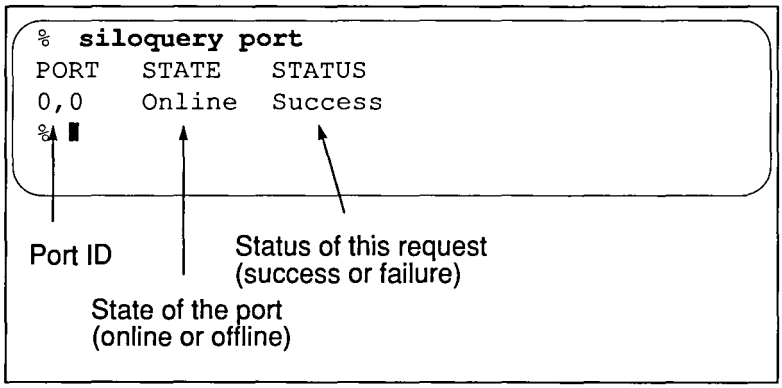
To report status information about an ACS port, use the `siloquery port` command. `siloquery drive` has the following syntax:

`siloquery port [`*portID*`...]`

portID is the location (port ID) of the port about which you are inquiring. The first port in the first ACS is generally numbered "0,0", the second port in the first ACS is numbered "0,1", the first port in the second ACS (if there is one) is numbered "1,0", and so on. If you do not specify a port ID, `siloquery port` reports the status of all ports in your system.

For an example of how `siloquery port` reports a port's status, refer to Example 25.

Example 25 Reporting port status



Reporting request status

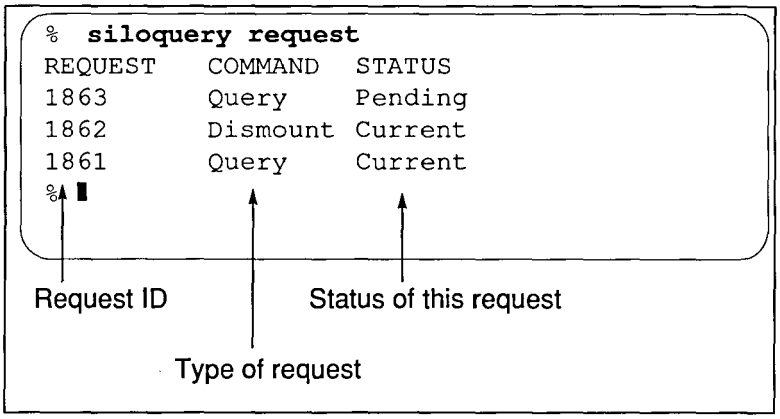
To report status information about an ACS request, use the `siloquery request` command. `siloquery request` has the following syntax:

`siloquery request [`*requestID*`...]`

requestID is the number assigned to a request when it is submitted. If you do not specify one or more request IDs, `siloquery request` reports information about all current requests.

For an example of how `siloquery request` reports the status of an ACS request, refer to Example 26.

Example 26 Reporting request status



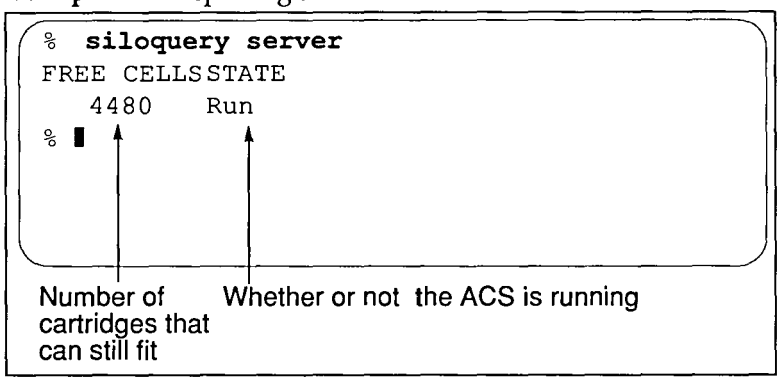
Reporting server status

To report status information about the ACS server, use the `siloquery server` command:

```
siloquery server
```

For an example of how `siloquery server` reports the status of the ACS server, refer to Example 27.

Example 27 Reporting server status



Reporting volume status

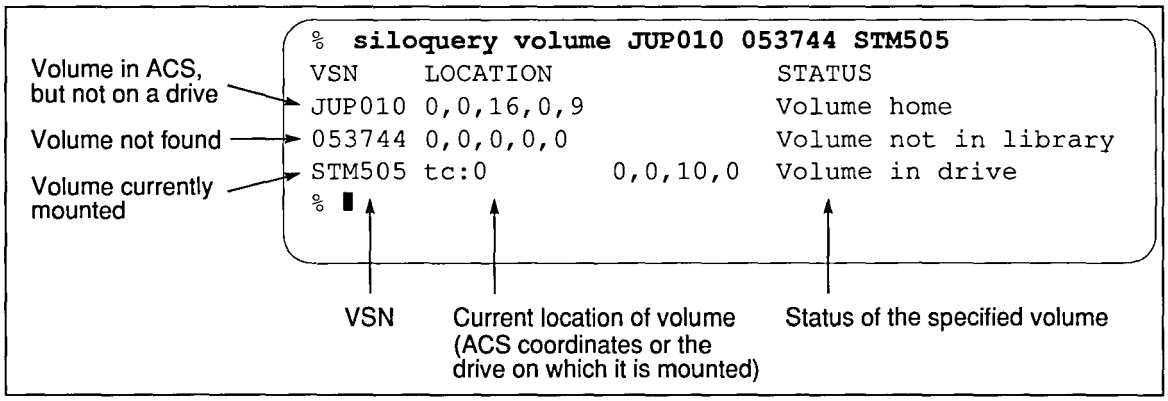
To report volume status information about any or all cartridges in the ACS silo, use the `siloquery volume` command. `siloquery volume` has the following syntax:

`siloquery volume [`*VSN*`...]`

VSN is the Volume Serial Number of a specific cartridge. If you do not specify one or more VSNs, `siloquery volume` reports the status of all cartridges in the ACS.

For an example of how `siloquery volume` reports the status of a cartridge, refer to Example 28.

Example 28 Reporting status of a cartridge volume

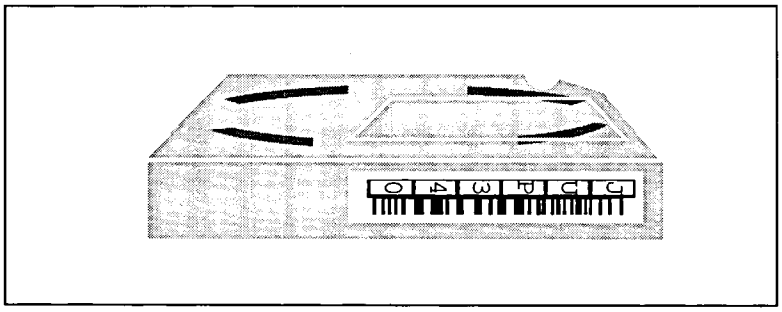


Entering cartridges into the ACS

To enter up to twenty-one 3480 tape cartridges into the ACS, use the following procedure:

Step 1 Ensure that each cartridge has a bar-code label on the outside as shown in Example 29. Each label should have a unique Volume Serial Number (VSN) and should be readable by the robot camera that is inside the ACS silo.

Example 29 Valid bar code label



Any cartridge that does not have a valid label or cannot be read by the robot camera is rejected by the ACS silo during the entering process.

Step 2 Use the following command to inform the ACS that you are entering some new cartridges and identifying the Controlled Access Port (CAP) that you are using:

`siloenter` *CAPid*

A sample of how this command is used is shown in Example 30.

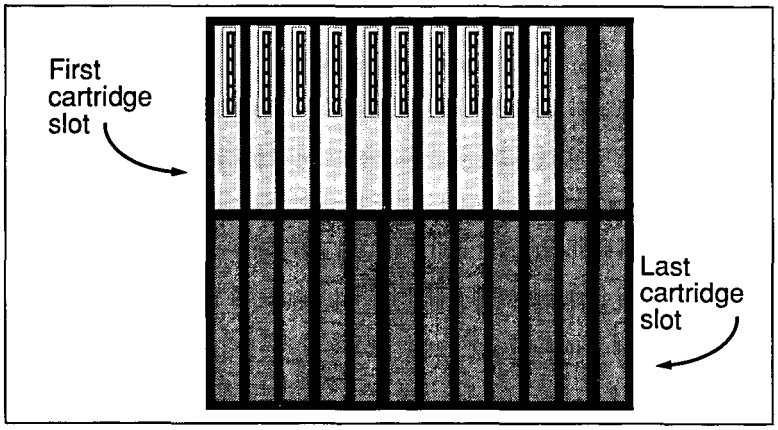
Example 30 Entering cartridges into the ACS

```
% siloenter 0,0
% ▮
```

This command unlocks the CAP door that corresponds to the *CAPid* you have specified. The "CAP enter" light above the CAP is illuminated. The tape system waits until the next two steps are completed before continuing.

Step 3 Open the CAP door and insert the cartridges into the slots from left to right starting with the top row. Do not leave any slots empty between the first and last cartridges. See Example 31.

Example 31 Loading cartridges into the ACS silo's CAP



Step 4 Close the CAP door.

The ACS silo robot camera verifies that each tape cartridge has a readable label and a unique VSN, and the robot arm stores the tape cartridges into the LSMs.

Any tape cartridges whose labels are not readable or have invalid VSNs are returned to the CAP door.

If the "CAP locked" light above the CAP is once again illuminated, the entering process is finished. A list of the VSNs of the newly entered cartridges and their statuses is displayed on the screen. See Example 30.

Example 32 Completion of cartridge entry into the ACS

```
% siloenter 0,0
VOLUME STATUS
JUP010 Success
JUP200 Success
MEL001 Success
% ▮
```

Step 5 If any cartridges have been rejected, the "CAP eject" light above the CAP is illuminated. After the "CAP eject" light is illuminated, remove any rejected cartridges. Do not leave any cartridges in the door.

Step 6 Close the CAP door after all the cartridges have been removed.

The robot arm verifies that the top left-hand slot of the CAP door is empty. When the "CAP locked" light above the CAP is once again illuminated, the entering process is finished. A list of the VSNs of the newly entered cartridges and their statuses is displayed on the screen. See Example 30 on page 109.

Ejecting cartridges from the ACS

To remove up to twenty-one 3480 tape cartridges from the ACS, use the following procedure:

Step 1 Use the following command to inform the ACS that you are ejecting some cartridges and identifying the Controlled Access Port (CAP) that you are using:

`siloeject` *CAPid VSN* `[`*VSN...*`]`

CAPid corresponds to the CAP ID of the CAP you are using, and *VSN* is the Volume Serial Number (VSN) of the cartridge you are ejecting.

Example 33 Ejecting cartridges from an ACS silo

```
% siloeject 0,0  000100 000200 000300
% ▮
```

This command unlocks the CAP door that corresponds to the *CAPid* you have specified. The "CAP eject" light above the CAP is illuminated. The tape system waits until the next two steps are completed before continuing.

Step 2 When the "CAP eject" light above the CAP is illuminated, open the CAP door and remove all the ejected cartridges. Do not leave any cartridges in the door.

Step 3 Close the CAP door after all the cartridges have been removed.

The robot arm verifies that the top left-hand slot of the CAP door is empty. When the "CAP locked" light above the CAP door is once again illuminated, the ejection process is finished.

A list of the VSNs of the newly ejected cartridges and their statuses is displayed on the screen as shown in Example 30.

Example 34 Completion of cartridge ejection from the ACS

```
% siloeject 0,0  JUP010 JUP200 MEL001
VOLUME STATUS
JUP010 Success
JUP200 Success
MEL001 Success
% ▮
```

Bypassing the ConvexOS tape system and mounting a cartridge with the ACS

The `silomount` command bypasses the ConvexOS tape system to mount a cartridge with the ACS. Normally, `tpmount` is used.

The `silomount` command informs the ACS of the VSN of the cartridge you want mounted and the tape device you want to use.

The `silomount` command has the following syntax:

`silomount` *VSN tapedevice*

VSN is the VSN of the cartridge you want mounted, and *tapedevice* is the name of the tape device you want the ACS to use.

Example 30 shows how to bypass the tape system when mounting a cartridge with the ACS.

Example 35 Bypassing the ConvexOS tape system and mounting a cartridge with the ACS

```
% silomount JUP340 /dev/rtc0
Volume JUP340 mounted on drive /dev/rtc0 (0,0,10,0)
% ▮
```

Bypassing the ConvexOS tape system and dismounting a cartridge with the ACS

The `silodismount` command bypasses the ConvexOS tape system to dismount a cartridge with the ACS. Normally, `tpunmount` is used.

The `silodismount` command informs the ACS of the Volume Serial Number (VSN) of the cartridge you want unmounted and the tape device it is currently using.

The `silodismount` command has the syntax

`silodismount [-force]` *VSN tapedevice*

which has the following components:

`-force`	Execute the dismount immediately regardless of the state of the tape drive.
VSN	VSN of the cartridge you want dismounted
tapedevice	Name of the tape device the cartridge currently is using

Example 36 Bypassing the ConvexOS tape system and dismounting a cartridge from the ACS

```
% silodismount JUP340 /dev/rtc0
Volume JUP340 dismounted from drive /dev/rtc0 (0,0,10,0)
% ▮
```

Tape system error messages

This appendix contains a description of error messages that can be returned by the tape system. Because it is impossible to describe all combinations of circumstances that can result in error messages, this list is not exhaustive.

Error messages in this appendix are organized as follows:

- General tape system error messages
- `tpconfig` error messages
- `opreq` error messages

These error codes are either displayed or sent to the error log file (usually /usr/adm/log/tapelog), or both.

Tape system error message labels in the tapelog file

Messages recorded in the /usr/adm/log/tapelog file will usually be labeled to indicate their severity. The following labels may be used:

EMERG	System is unusable.
ALERT	Action must be taken immediately.
CRIT	Critical conditions
ERR	Error conditions
WARNING	Warning conditions
NOTICE	Normal but significant condition
INFO	Informational; no action required.
DEBUG	For debugging only; normally should not be seen.

General tape system error messages

This section alphabetically lists and describes some of the error messages you may receive when encountering problems while using the tape system. Error messages may be sent by the tape system to your terminal screen or to a log file in the form:

command: *error_message*

command is the command you were using when the error occurred, and *error_message* is the reason the error occurred.

Each error message entry suggests a possible solution. This solution is provided for the most common occurrences and may not be valid in all situations.

`ambiguous request`
The tape system cannot distinguish the tape and drive on which to take action. Specify the symbolic-link name of the device with the appropriate option.

`can't create symbolic link`
You may not have write permissions for the directory from which you have invoked `tpmount`, or a file exists with same name as the requested symbolic link to the tape device.

You must change to a directory that you can write to or change the permissions of the current working directory, or specify a unique symbolic-link name with the `tpmount -s` option.

`can't create symbolic link to stacker`
The symbolic link to the stacker device could not be created. The reason for the failure is appended. For example:

```
tpmount: can't create symbolic link to \
stacker /mnt/guest/TAPE.STK: File exists
```

In this case, a file called "TAPE.STK" already exists; you should use `-s` with `tpmount` to specify a different link name.

`can't establish connection with label daemon`
The label daemon for the label type you have requested doesn't exist.

If the label type is a valid label type that is supported by CONVEX, check if it has been defined with `tpconfig`:

```
% tpconfig show labels
```

If the label type is not defined, refer to Chapter 6, "Adding and deleting label types," on page 51.

Another option is to bypass label restrictions and copy the contents of the tape to another tape that has a proper label. Refer to the *ConvexOS Tape System Operator's Guide* for information on bypassing label restrictions.

`can't get current working directory`
The directory you are currently in has been deleted. Change to another directory to issue your mount request.

`device and mode requested do not match`
You have specified character or label access mode to `tpmount` and the specified device access mode is block.

`device is opened by another process`
The device you have attempted to deallocate is currently in use by a process. You must wait until the process closes the tape device to deallocate it.

`empty request`
You have not specified a tape in your `tpmount` library call.

`EOT seen`
The utility you are using has encountered the end of the tape. If there is still more tape, this indicates there is a bad spot on it.

`file exists`
The symbolic link displayed with this error message exists, so it cannot be created as you have specified in your mount request.

Specify a unique symbolic-link name with the `tpmount -s` option.

`internal tape system error`
There is currently an internal problem with the tape system. Refer to Chapter 2, "Keeping the tape system up and running," on page 7. It may be necessary to kill and restart the tape system, and, possibly, to shut down ConvexOS and reboot.

`invalid drive name specified`
You have specified a drive name in your `tpmount` request that is not defined for your system.

Use the `tpconfig show drives` command to list defined drive information.

`invalid drive type`
You have specified a drive type in your `tpmount` request that is not defined for your system.

To display information about defined drives, enter:

```
% tpconfig show drives
```

`invalid label type`
You have specified a label in your `tpmount` request that is not defined for your system.

Use the `tpconfig show labels` command to list defined label types.

`invalid path to symbolic link`
The path you have specified to your symbolic link is not a valid path. One or more of the directories in the path do not exist.

Reissue your mount request after determining a proper path for the symbolic link.

`labeled tapes only`
`tpconfig labeled-mode-only` is enabled which requires tapes to be mounted in labeled mode. Only users with bypass access may mount tapes in character or block mode.

`magnetic vsn doesn't match visual vsn`
The VSN or VSNs you have specified in your mount request do not match the VSN or VSNs on the tape.

If you do not know the proper VSN or VSNs for the tape, and you are the owner or root, remount the tape in unlabeled mode and bypass the label with the `tpmount -b` option. View the first file on the tape, which contains a field listing the proper VSNs. Refer to the *ConvexOS Tape System Operator's Guide* for more information on bypassing label restrictions.

If you are using a tape set, you must list all the VSNs in the set with the `tpmount -s` option.

`mount not yet complete`
You have entered a command before your tape has been mounted.

If you placed your mount request in the background with the `tpmount -B` option, use the `tpwait` command to bring it to the foreground.

If you haven't placed your mount request in the background, wait until the mount is complete.

`mount request failed`
You have passed a null pointer as an option in your `tpmount` library request.

`mount request failed - tape is labeled`
You have probably tried to mount a labeled tape with a block device or tried to mount a labeled tape in character or block mode without bypassing the label.

To display character devices you should use with labeled tapes, enter:

```
% tpconfig show nodes | grep "rewind Char"
```

To bypass a tape label (if you are the owner of the tape or have superuser privileges only) mount the labeled tape with the `tpmount -b` option. Otherwise, mount the labeled tape in labeled mode with the `tpmount -m` option.

`no access`
You are not on the user or group access drive list for the specified drive.

`no alloc access`
You are not on the user or group allocate drive list for the specified drive. You cannot reserve the drive and must specify a particular tape to mount.

`no authorization`
The drive you have selected is not controlled, or you have attempted to unmount a tape without proper authorization. You must either select a controlled tape drive or unmount the tape as the appropriate user, respectively.

For help in selecting a proper drive that is controlled, enter:

```
% tpconfig show drives | grep Cont
```

To display a list of tape devices that have been allocated locally and by whom, enter:

```
% tpqueue -l
```

`no bypass access`
You specified `tpmount -b` to bypass label processing, but are not on the user or group bypass labels list for the drive.

`no drives meet the required speed density and type`
You have specified attributes that do not correspond to a defined device. For help in selecting a proper device enter:

```
% tpconfig show nodes
```

`no entry in password file`
Your ID is not in the password file /etc/passwd.

`no input`
You have attempted to use `tar -t` to list files in an archive on a tape that is at EOT.

Reposition the tape using `mt` commands.

`no matching drives are available at this time`
You have specified attributes that correspond to a defined device that is currently in use.

To display a list of tape devices that have been allocated locally and by whom, enter:

```
% tpqueue -l
```

For help in selecting a proper device, enter:

```
% tpconfig show nodes
```

`no such request`
You have issued a tape command that is invalid either for the type of tape you have mounted or for the current situation.

`no such file or directory`
The file you are trying to read from an archive doesn't exist.

Use a command to list the files on the labeled tape, if using a labeled tape, or use a command to list the files in the archive, if trying to access an archive. If you are trying to read from disk, list the directory you are trying to read from.

`no such tape`
You have requested a tape to be mounted that is not currently included in the tape library. Enter the proper tape.

`queueing is disabled - please specify device with tpmount -a option`
A device cannot be selected for you by the tape operator. You must select your own device with the `tpmount` option `-a`. For help in selecting a proper device enter:

```
% tpconfig show nodes
```

`Read error`
The utility you are using cannot read from the tape. This is probably because you do not have the proper device mounted, or you have not selected the proper tape attributes (if you are using a labeled tape).

`server didn't respond`
`tpdaemon` has not responded to your request due to network problems. This message is accompanied by an RPC (remote procedure call) error message.

`tape cannot be labeled`
You have attempted to label a tape not mounted in labeled mode.

First, be certain the correct tape is on the tape drive. If it is the correct tape, unmount it, remount it in labeled mode, and then label it. Any information that is on a tape at the time of labeling is no longer accessible after labeling.

`tape cannot be unlabeled`
You have attempted to unlabel a tape not mounted in labeled mode.

First, be certain the correct tape is on the tape drive. If it is the correct tape, unmount it, remount it in labeled mode, and then unlabel it. Any information that is on a tape at the time of labeling is no longer accessible after labeling.

`tape change in progress`
The tape operator is changing the tapes. Wait until the change operation is complete to continue.

`tape forward error`
When using the `tar -t` command to view files in an archive, this message means that the tape is at EOT. This happens when listing files in the last archive on a tape.

`tape device must be no-rewind for labeled tapes`
You have requested a rewind device. Rewind devices are not compatible with labeled tapes.

Select a device that is no-rewind. For help in selecting a proper no-rewind device, enter the following:

```
% tpconfig show nodes | grep No
```

`tape drive is not on-line or not ready`
The tape drive is probably not online yet; or, you have issued a mount request for a labeled tape without an operator, and the tape has not yet been placed on the drive.

For 9-track tape drives, the online button on the tape drive probably needs to be pressed. For cartridge tapes, ensure the tape has been inserted properly.

If you are using labeled tapes, you must place the tape on the drive and put the drive online before issuing a mount request.

`tape is already in use`
You have specified a tape that is already in use.

To display a list of tape devices that have been allocated locally and by whom, enter:

```
% tpqueue -l
```

`tape is already labeled`
You have attempted to label a tape that is already labeled.

First, be certain the correct tape is on the tape drive. If it is the correct tape, use `tpunlabel` to unlabel it and then relabel it. Any information that is on a tape at the time of labeling is no longer accessible after labeling.

`tape is already unlabeled`
You have attempted to unlabel a tape that is not labeled, but is mounted in labeled mode.

`tape is mounted read only`
The tape you have mounted as read-and-write is physically set for reading only.

The tape should be unmounted, taken from the drive, and have its write authorization affixed. You can then remount it with read and write permissions.

`tape is mounted read/write`
The tape you have mounted as read-only is physically set for reading and writing.

The tape should be unmounted, taken from the drive, and have its write authorization removed. You can then remount it as read-only.

`tape is not known to tape system`
You have requested a tape that is not currently in the tape library. Enter the proper tape.

`tape is not labeled; use tplabel to label tape`
You have mounted an unlabeled tape in labeled mode.

First, be certain you have mounted the correct tape. Once you label a tape, any information on it is lost. If you get this message, you must label the tape in order to use it further.

`tape mount canceled by operator`
There is a problem with your tape, tape drive, or request. This message may be followed by a comment from a tape operator. If not, contact them.

`tape must be mounted as labeled to use this command`
You have attempted to use labeled-tape utilities on an unlabeled tape.

First, be certain the correct tape is on the tape drive. If the correct tape is on the tape drive, you must unmount the tape, remount it in labeled mode, and label it before using labeled-tape utilities. Any information that is on a tape at the time of labeling is no longer accessible after labeling.

`unable to connect to server`
There is a problem connecting to your server.

`unable to set attributes`
You have attempted to set attributes on a tape that is not labeled. You must label the tape first with the `tplabel` command. Any information that is on an unlabeled tape at the time of its labeling is not accessible after it is labeled.

`Warning: tape is not labeled; use tplabel to label tape`

You have mounted an unlabeled tape in labeled mode.

First, be certain you have mounted the correct tape. Once you label a tape, any information on it is lost. If you get this message, you must label the tape in order to use it further.

`Write error`

The utility you are using cannot write to the tape. This is probably because you do not have the proper device mounted, or you have not selected the proper tape attributes (if you are using a labeled tape).

tpconfig error messages

The following error messages may appear on your screen while you are running `tpconfig`.

`Access is read-only.`
The configuration database is locked. You cannot edit the `tpconfig` database because you do not have authorization.

`Ambiguous command.`

`Ambiguous keyword match`
Not enough characters were given to distinguish the command or keyword from another command or keyword. Enter a more complete form of the command.

`Cannot create new database.`
The database /usr/lib/tape/config.db does not exist, and you do not have permission to create a new database. Make sure that you are logged on as root.

`Cannot open database: /usr/lib/tape/config.db`
The indicated database could not be opened. Check access permissions of the database with `ls`.

`Database access is read-only.`
You do not have permission to make changes to the tape system configuration database. (Only root has permission.) You may, however, read information from the database. Only the `show` commands work when the database is read-only.

`Database does not exist. Creating new database.`
Because /usr/lib/tape/config.db did not exist, a new database is being created.

`Database is readonly.`
The user issued a command to change the configuration but only has read-only access. (Only root has permission to change the database.)

`Error reading defaults`

`Error reading drives`

`Error reading labels`

`Error reading node`
The configuration database is corrupted. Check for other error messages. The problem can be fixed by removing the old configuration file and creating a new one.

`Error writing defaults`

`Error writing drives`

`Error writing labels`

`Error writing nodes`
A write error occurred while writing out a new configuration. These messages are always accompanied by a system error message.

`Interrupt.`
The keyboard interrupt character (usually **CTRL-C**) was typed. If you want to exit `tpconfig`, type the EOF character (usually **CTRL-D**).

`Missing quote.`
Somewhere on the command line there is a mismatched quote character.

`Node must be either block special or character special.`
Only character special files and block special files can be given as a node in the `add node` command.

`readdrives: malloc failed`

`readlabels: malloc failed`

`readnodes: malloc failed`
There is not enough memory available for `tpconfig`.

`readnodes: stat failed on /dev/rmt?`
One of the nodes defined in the database no longer exists. You should delete that node with the `delete node` command.

`The database is being modified by another person.`
The configuration database is locked. You cannot edit the `tpconfig` database because another user has it currently open.

`Unable to write out configuration.`
`tpconfig` could not write the configuration back to /usr/lib/tape/config.db. Verify that you still have write permission and that the disk is not full.

`Warning: Invalid daemon path.`
The daemon specified with `add label` does not exist.

opreq error messages

The following error messages may be displayed on your screen or sent to the log file to indicate error conditions while you are using `opreq`.

`At limit`
An attempt was made to move past the edge of the screen.

`Invalid command`
A command was issued that `opreq` does not recognize.

`Invalid drive`
The selected drive cannot be mounted.

`Invalid window`
An attempt was made to select a window that is not available.

`No current message`
A message operation has been attempted, but no message is currently selected. Move the selection cursor over the desired message.

`No current window`
An operation was attempted that requires a window to be active, but no window is active.

`No other window`
An attempt was made to move to another `opreq` window; no other window exists.

`Not implemented`
You have attempted to use an unimplemented command or feature.

`Not ready`
A `select-done` command was attempted on a message that is not in "Ready" status.

`Restricted command`
You do not have permission to carry out the command. To use its full capabilities, you must run `opreq` as root or be in the group tapeop.

`Tape server: opreply() failed`
Internal error: an attempt to send a message via `opreply()` has failed.

`Tape server: timed out`
The tape drive has not been used for longer than the defined idle time; it has been unmounted.

Index

A

B

C

D

E

F

H

I

L

U

V

W

Order Number
DSW-398

Document Number
710-020430-001