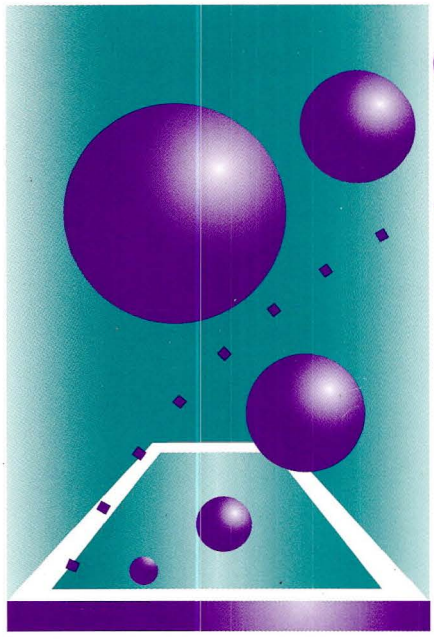


CONVEX

ConvexTMR Administrator's Guide

First Edition

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

ConvexTMR Administrator's Guide

Order No. DSW-480

First Edition

October 1994

CONVEX Press
Richardson, Texas
United States of America

ConvexTMR Administrator's Guide

Order No. DSW-480

Printed in the United States of America

Revision information for ConvexTMR Administrator's Guide

Edition	Document No.	Description
First	710-029630-000	Released with ConvexTMR V1.0, October 1994.

Contents

Figures

Tables

Preface

Purpose and audience

The *ConvexTMR Administrator's Guide* is for ConvexOS and SPP-UX system managers who configure and maintain ConvexTMR (Convex Tape Mount Request) software.

Using this guide

This book is organized into the following chapters:

- **Overview**—Introduces ConvexTMR and the fundamentals of tape management. This chapter defines tape library organization and operations, data access methods, tape life cycles, and tape objects. This chapter also outlines steps for getting started, lists administrator commands, and contains a step-by-step walk through of some basic commands.
- **Configuration**—Contains instructions for configuring administrator and operator groups, configuring the tape server, and defining operator domains. This chapter also explains how to define tape system defaults and describes user privileges.
- **Administration**—Explains how to enroll as a tape administrator and describes message options, accounting, and log files.
- **Glossary**—Defines terms used in this document.

Notational conventions

This section discusses notational conventions used in this book.

`Bold monospace` In command examples, text shown in **`bold monospace`** identifies user input that must be typed exactly as shown.

`Monospace` In paragraph text, `monospace` identifies:

- Command names
- System calls

- Data structures and types

In command examples, `monospace` identifies command output, including error messages.

In command syntax diagrams, text shown in `monospace` must be typed exactly as shown.

Italic

In paragraph text, *italic* identifies:

- New and important terms
- Titles of documents

In command syntax diagrams, *italic* identifies variables that must be supplied by the user.

{ }

In command syntax diagrams, text surrounded by curly brackets indicate a choice. The choices available are shown inside the curly brackets and separated by the pipe (|) sign.

The following command example indicates that you can enter either `a` or `b`:

```
command {a | b}
```

[ ]

In command syntax diagrams, square brackets indicate optional data.

The following command example indicates that definition of the variable *output_file* is optional:

`command` *input_file* [*output_file*]

...

In command syntax, horizontal ellipsis shows repetition of the preceding item(s).

The following command example indicates you can optionally specify more than one *input_file* on the command line.

```
command input_file [input_file ...]
```

KEYCAP

In paragraph text, text shown in **KEYCAP** indicates keyboard keys you must press to execute the command. 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.

Notes, cautions, and warnings

This document presents notes, cautions, and warnings in the following formats.

Note

A Note highlights supplemental information.

Caution

A Caution highlights information necessary to avoid damage to the system.

Warning

A warning highlights information necessary to avoid injury to personnel or loss of data.

Associated documents

Using this software may require information not specific to the tasks described in this document.

For more information about ConvexTMR, you can order these books from CONVEX Computer Corporation:

- *ConvexTMR Operator's Guide* (DSW-482). This book describes the Convex Tape Mount Request operator interface.
- *ConvexTMR User's Guide* (DSW-481). This book describes the Convex Tape Mount Request user interface.
- SPP-UX System Administration Guide (DSW-853). This book is the standard reference for the SPP-UX operating system.

Ordering documents

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

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

Please include the order number (DSW or DHW number) or the exact title of the document.

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, if you would like to report any problems you may have with ConvexTMR or its associated documentation. For more information refer to the contact(1) man page in *ConvexOS Man Pages for Users*, or the appendix "Reporting problems" in the *ConvexOS Primer* or *Managing ConvexOS: Operations Guide.*

Overview

1

Introduction

ConvexTMR is a tape management software system that integrates a full-featured tape manager with powerful cataloging abilities. ConvexTMR offers the following features and services:

- role-based interfaces
- extensive media, device, and recording format options
- interactive and batch tape session support
- data security control
- tape drive arbitration
- complete tape life-cycle management
- operational domain support
- unlimited vaulting and rotation options
- user pool creation options
- comprehensive logging and reporting facilities

The mount request system

This section describes the mount request system.

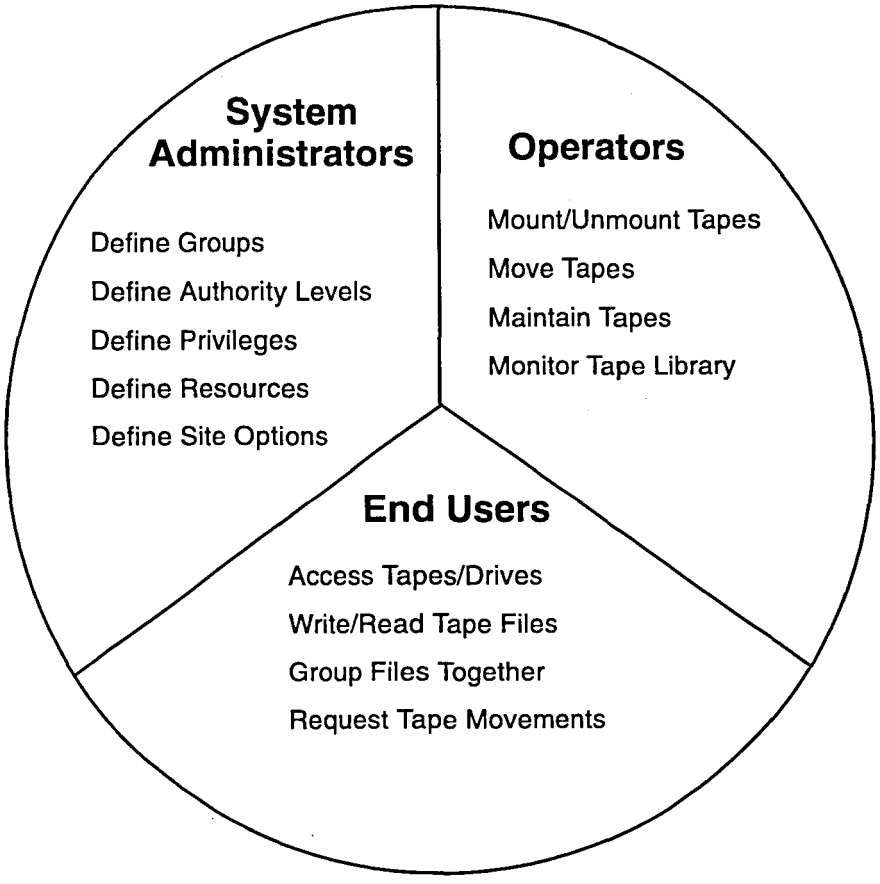
Role-based interfaces

ConvexTMR brings together three distinct role groups: system administrators, operators, and end users. ConvexTMR provides separate interfaces and programs for each of these user groups, each tailored to the specific needs of the particular group.

Figure 1 illustrates how these three groups perform their roles interdependently.

The following sections introduce the three ConvexTMR role groups and list the activities of each.

Figure 1 ConvexTMR role groups



Administrator activities

The ConvexTMR administrator configures the ConvexTMR software and supervises its operation. Administrator activities include defining

- administrator groups and authority levels
- operator groups and authority levels
- user and group privileges
- system resources
- site options

Operator activities

The ConvexTMR operator monitors the software operation and performs the day-to-day physical tape library tasks. All mounting, vaulting, and tape management requests are directed to the ConvexTMR operator via the Request Monitor.

Operator activities include

- mounting and unmounting tapes
- moving tapes to new locations
- performing tape maintenance:
  - cleaning tapes
  - erasing tapes
  - initializing tapes
  - identifying found tapes
  - monitoring and controlling the server programs

User activities

The ConvexTMR user stores and retrieves tape file data during tape sessions. Users may control the disposition and access rights to their off-line data via the ConvexTMR catalog. User activities include

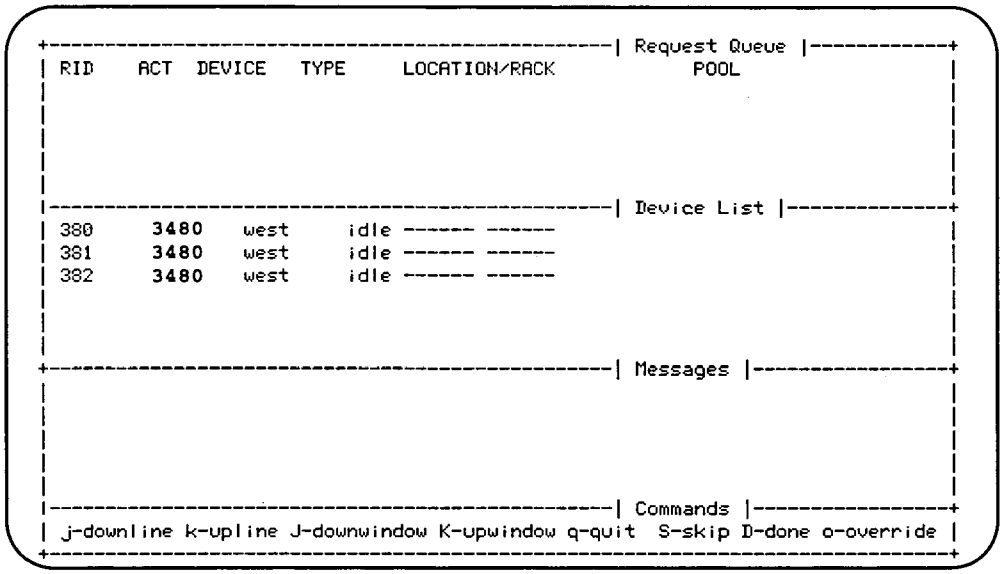
- accessing tape drives and tapes for tape sessions
- writing and reading tape file data
- grouping tape files together in logical ways
- requesting tape movement among library sites

Request monitor

The Request Monitor is the tape operations console for ConvexTMR. ConvexTMR operators receive all tape service requests via the Request Monitor. Operators can also view the current status of tape devices and receive electronic messages via the Request Monitor.

ConvexTMR provides both command-line and full-screen Request Monitor interfaces; both interfaces offer complete functionality. The full-screen Request Monitor is shown in Figure 2.

Figure 2 The full-screen request monitor



User tape sessions

ConvexTMR supports both interactive tape sessions and batch operations. It collects incoming requests in a prioritized queue, allocates the necessary resources, and directs the operator to perform the needed activities.

ConvexTMR offers two data access methods. In addition to traditional volume-level access, ConvexTMR also provides file-level access.

Volume-level access

In volume-level access, the entire tape volume is accessible to users. Each file access request may need to include instructions

to the tape device to wind or rewind the tape to a specific position.

File-level access

An alternative to volume-level access, file-level access allows the selection of a file by name, as if the tape file were a file on a disk. This frees the user from the tape navigation concerns inherent in volume-level access. With file-level access, it is also possible to make different files on the same volume accessible to discrete user groups.

The catalog

The ConvexTMR on-line catalog is a comprehensive database that tracks all the objects in the library. The catalog maintains records on many attributes of ConvexTMR objects. Some of these attributes are listed below.

- ownership
- contents
- association with other tapes
- pools
- vaulting
- status

Operating without a catalog

Some ConvexTMR sites choose not to order the additional catalog package. Parts of some commands and the entirety of others are inactive in no-catalog mode. These differences are noted throughout the documentation.

Tape naming and identification

ConvexTMR stores identification information for all cataloged volume objects. This allows users to reference volume objects in a variety of ways. Users may reference volume objects by

- relative or absolute path name (volume sets only)
- external label
- system-generated database key
- receipt number

In addition, ConvexTMR stores the following identifying information on each cataloged volume:

- volume internal label
- vault slot number
- storage container
- volume fingerprint

The volume fingerprint is an encryption of several key label fields to create a guaranteed unique volume identifier. This fingerprint is checked electronically at the start of the tape session to safeguard against unauthorized tape access.

Tape life cycle management

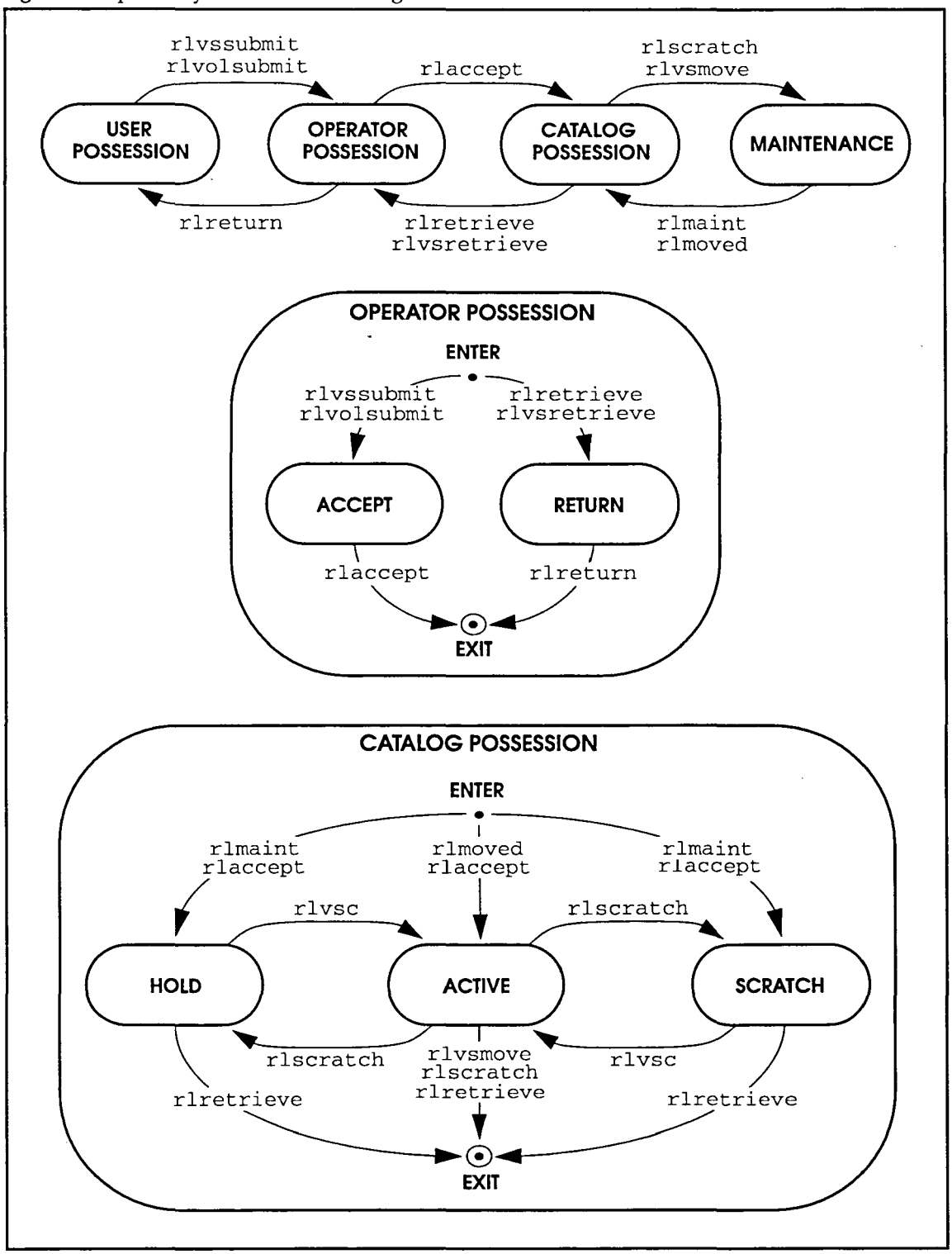
The ConvexTMR catalog provides complete tape life cycle management. The catalog tracks and controls the status of each tape in the library, controlling the transition of tapes between states, providing security, and scheduling and tracking maintenance tasks.

Each tape associated with the library is always in on one of seven life cycle states:

- user possession
- accept
- return
- scratch
- active
- maintenance
- hold

Figure 3 illustrates these life cycle states and the ConvexTMR commands used to move tapes between these various states. The sections following the illustration introduce the tape life cycle states with a brief definition for each.

Figure 3 Tape life cycle nested state diagram



User possession

A tape in user possession may or may not contain active data. The user cannot access data on this tape at a ConvexTMR site until the tape is submitted to the library.

Operator possession

Operator possession includes the accept and return states.

Accept

A tape is in the accept state after possession has been transferred from the user to the tape library. Once a tape is accepted, its catalog record may be appended or edited.

Return

A tape is in the return state after possession has been transferred from the library to the user. A tape in this state is not tracked by the ConvexTMR catalog.

Catalog possession

Catalog possession includes the hold, active, and scratch states.

Hold

A tape is in the hold state if the user wishes to scratch it and reserve it for future use. For example, a user may request that a multiple-volume volumeset be scratched, but that the individual volumes not be made available for other mount requests. Putting the scratched volumes in a hold state makes them unavailable until they are taken to an active state using the rlvsc command.

Active

An active tape stores wanted data that has not expired. These tapes may be involved in frequent tape sessions, or they may reside in the library's archive and remain unaccessed for long periods of time.

Scratch

A scratch tape is not in current use, and any data written to it has expired. Scratch tapes contain no wanted data and are available for assignment to a volumeset. You may think of a scratch tape as a blank tape.

Maintenance

Tape maintenance is tracked by the ConvexTMR software and performed by the tape operator. A tape is in this state whenever it is undergoing any maintenance task. These tasks include

- cleaning
- certification
- disposal
- erasure
- moving

Data security

ConvexTMR is designed to prevent unauthorized tape access. Ownership of data is confirmed via the catalog for each access request. Other security features include

- command authorization
- electronic fingerprinting
- passwords
- administrator-controlled user privileges
- Access Control Lists (ACLs) for catalog objects:
  - files
  - volumesets
  - pools
  - directories
  - rotations
  - devices

Vault management

ConvexTMR can accommodate multiple storage vaults and tape rotation schedules.

ConvexTMR vaults are defined by the system administrator based on the site's physical resources and organizational and security needs. Tapes are stored according to the vaulting conventions of the particular site.

Users move tapes among vaults in two ways: by making ad-hoc movement requests, and by assigning tapes to customized rotations. A rotation is a schedule of storage locations and

durations for each location. Users create ConvexTMR rotations as database objects, and then apply them to specific volumesets.

Tape pools

Tape pools partition the tape library into sub-libraries, and provide the library with a logical way to organize groups of tapes. All tapes in the library belong to one and only one pool. The characteristics of a particular pool, such as media type and recording format restrictions, access restrictions, and volume ownership within the pool, are determined by the pool's creator. Pools may be created by library administrators and by users.

Many tape libraries employ default pools. At these libraries, each volume is automatically submitted to a default pool unless another pool is specified. The library may also be configured to create implicit user pools. These are private tape pools automatically created for each library user. Tape libraries that do not employ a default pool or implicit user pools require users to specify a pool each time they submit a tape.

Like rotations, tape pools are database objects that volumes are assigned to. As such, pools represent logical, rather than physical, groups of tapes. The tapes that comprise a single pool are not required to reside in the same vault, and tapes from various pools may occupy a single storage rack.

Configuration

The ConvexTMR administrator must make several configuration decisions before the tape library is ready to process user requests. Some of the questions an administrator needs to address are listed below.

- What devices will be under ConvexTMR control?
- What tape labelling and storage conventions will the library enforce?
- What media types and recording formats will the library accommodate?
- How many domains are necessary, and how should each be configured?
- What administrator and operator groups should be created, and which powers should be extended each of these groups?
- What fees will be charged for tape mounts and drive allocation time?

- What user privileges will be extended to individuals and groups?
- Will ConvexTMR use an on-line catalog, or no catalog?

Device configuration

ConvexTMR device options are to some extent limited by the types and quantities of devices available to the library. But within these limitations, many configuration options exist.

Media type, recording format, and device type defaults

For example, when a user makes a request that specifies a device model but not a recording format or media type, what default values will the system employ? The ConvexTMR administrator determines these default values and stores them in the configuration files `mmr_map`, `mtype_map`, and `record_map`. Each time a user makes an access request, these files are examined to provide the request with compatible default values.

Default and custom domains

An operator domain is a set of tape drives and accompanying capabilities. A ConvexTMR site may have one or more domains. The default domain covers all devices and capabilities; other domains may be defined to limit the scope of the domain to specific devices and/or specific functions. For example, a domain that is serviced by a robot operator would necessarily have a limited set of devices and activities. Operator domains are defined by the library administrator and stored in the configuration file `dom_map`. The `dev_config` file determines which domain will be the system default; other domains are invoked via the `ods` command.

Site conventions

All tape libraries have conventions for tape labelling, storage, and other site options. For example:

- At some sites, the volume internal and external label always match, while at other sites these values may differ.
- Some sites restrict the set of acceptable characters or regular expression that may be used in an internal or external label, slot, or container name.
- A site may require unusually low or unusually high thresholds for maintenance tasks such as cleaning, replacement, and certification.

- Sites may require high or low intervals between operator reminder messages and user inactive warning messages.

The ConvexTMR administrator determines these and other site-specific conventions and records them in the `site_options` file via the `rlcop` command.

Administrator and operator work groups

To perform administrator or operator functions, all library personnel must be enrolled in one or more administrator or operator work groups. The ConvexTMR administrator creates and controls these groups.

All work groups are assigned an authority level. The authority level of a group defines the powers extended to the members enrolled in the group. Authority levels are hierarchical (level 1 being the highest); higher authority levels possess all the powers of lower levels, plus additional powers. This way, more senior and experienced personnel may perform a wider range of tasks than new and/or inexperienced staff members.

User privileges

ConvexTMR administrators control the privileges of ConvexTMR users and user groups. There is a restricted set of privileges that must be explicitly and individually extended to users and groups. These privileges are listed and explained in the rlauth(8) man page.

Getting started

ConvexTMR is a comprehensive tape management system and therefore there are many commands, options, and concepts to learn. It is best to read all of the documentation thoroughly before conducting actual tape sessions. But to familiarize yourself with the software and begin using it more quickly, follow these steps.

1. Install ConvexTMR according to the instructions in *ConvexTMR Installation Procedures*.
2. Read Chapter 2, "Configuration." Follow the instructions for configuring work groups and customizing the configuration files.
3. Follow the instructions for enrolling as an administrator provided in Chapter 3, "Administration."
4. Read the rlcop(8) man page; follow the instructions provided for determining site options.
5. Read Chapter 4, "The request monitor," of the ConvexTMR Operator's Guide.
6. Try the "Short walk through ConvexTMR" presented at the end of this chapter.
7. Read the examples of the `rlaccess` command found in Chapter 2, "Tape Sessions," of the *ConvexTMR User's Guide*.
8. Assemble your tapes. Be sure to distinguish scratch tapes from active data tapes.
9. Conduct tape sessions with the `rlaccess` command.

Command Summary

Table 1 summarizes the commands available to the ConvexTMR administrator.

Table 1 ConvexTMR administrator commands

Command	Function
`rladm`	enroll as a tape administrator
`rlagroup`	control administrator and operator groups
`rlauth`	grant user privileges
`rlcop`	configure site options
`rldbck`	check database and report or fix errors
`rlrebuild`	recover ConvexTMR catalog records
`rlupriv`	designate user privilege

A short walk through ConvexTMR

This section provides a step-by-step walk through the ConvexTMR system. It demonstrates a simple use of the product.

To follow the steps listed below you will need access to two separate terminal sessions. On one session you will be acting as a user, making user requests; on the other session you will be servicing those user requests, as an operator. You will need to verify that both */usr/convex* and */usr/lib/REEL* are in your default command path.

	Operator	User

Step 1	From the first (operator) terminal session, start the servers; log in as root and enter:
	
	# `convextmr start`
	
	The following message is displayed:
	
	`ConvexTMR version 1.1`
	
	`Starting Servers:`
	`rlsvc...complete`
	
	If you did not receive this response, then the software may not be installed correctly. Retry the install procedure outlined ein the ConvexTMR Installation Procedures before returning to this step.	

Step 2	If you have tpdaemon running, you will have to make the device "not controlled" for ConvexTMR to be able to access it. Enter **`tpconfig set con off`** *{tpdaemon device name}* example: **`tpconfig con off tc:10`**	

Operator	User
To configure the drives to be available, enter	
`rldev -s status=up` *{ConvexTMR device name}* example: **`rldev -s status=up`** tc10	
This command is required for rloreserve to work. The following message is displayed:	
`Device: tc10` **`Updating...Device request accepted`**	
Step 3 To reserve an operator device to write labels to a tape. Enter	
`rloreserve` {*device name*}	
The following status is displayed:	

`Waiting for request: 1024`
 	
`Device {`*device name*`}: Reservation complete`	

Step 4 To initialize the tape. Enter	
`rlinit -c UNCAT -l IBM -d {`*device name*`} 0001`	
The following message is displayed:	
`MOUNT volume '0001' on device '{`*device name*`}'. Press return when ready (or 's' to skip, 'q' to quit): Tape Mounted`	
Step 5 Mount the tape and press **Return**. The following message is displayed:	
Scratched 0001 IBM intlbl=0001 Ejecting Volume '0001' on device '{*device name*}...	

	Operator	User
Step 6	Unmount the tape. It is now labeled with IBM volume labels.	
Step 7	Free the device by entering **rlofree** {*device name*}	
Step 8		Create a directory for this walk through session. This step assumes that you are using the catalog. Enter **rlmkdir /walkthrough**
Step 9		Move into the newly-created directory. Enter **rlcd /walkthrough**
Step 9		Submit a volumeset with a single volume in it. Enter **rlvssubmit -V 0001 -l IBM test_vs**
		The following message is displayed: Volume 0001 : Receipt 'R76835'...Attached
Step 10	Accept the request. Enter **rlaccept -e0001 :rR76835**	

	Operator	User
Step 11		View the cataloged information about the volumeset. First verify your current ConvexTMR directory. Enter **rlpwd** The following is displayed: `/walkthrough`
Step 12		List the contents of the current ConvexTMR directory. Enter **rlls** The following is displayed: `test_vs`
Step 13		View a long list of the contents of the directory by entering **rlls -l** Something similar to the following is displayed:

```
Vrwxoe>----->----->  lfw stortek Aug 12 10:33 test_vs
```

Step 14		Request an extended report on the newly-created volumeset by entering **rlls -L test_vs**

Operator	User
	Refer to Chapter 7 of the *ConvexTMR User Guide* for a complete explanation of all the ConvexTMR reports. A report similar to the one shown below will appear on screen:

```
VOLSET                                                          VOLSET


              name: test_vs

             owner: stortek                    uperm: rwxoedmp---s
             group: lp                         gperm: ------------
                                               operm: ------------
                                         acl entries: 0

              pool: u_stortek                  vault: onsite
          rotation: NONE                     catalog: @_UsT_
        media type: 3480                  expiration: :S
     recording fmt: 3480                 disposition: scratch
         label fmt: IBM                         erase: FALSE
     file tracking: TRUE                     scratch: notscr
           comment:

Dates
       date expired: Dec 31 1969
           creation: Aug 12 10:33
        last access: Aug 12 10:33
  last modification: Aug 12 10:33

Keys
                key: 2e4b964e00000001Z
VOLSET                                                          VOLSET
```

Step 15	Request a list of all the volumes in the new volumeset by entering **`rlls -r vsvlist test_vs`**

Operator	User
	A report similar to the one shown below will appear on screen:

```
Volume Set Volume                          Fri Aug 12 10:36:08 1994

Volset: test_vs

     vsvn     extlbl          vault        status
     ----     ------          -----        ------
     1        0001            onsite       active
```

Step 16 Start the command line request monitor by entering

rldop

The following message will be displayed:

```
Command line request server started
```

Step 17 Access the volumeset to write the password and group files by entering

rlaccess -RW -V :Ntest_vs -F :Npasswd:A tlink

Using the :A sub-option makes writing implicit when the request is submitted. If you want to read the file following a write, then you must submit the request with the explicit -RW option.

Step 18 A mount request similar to the one below should appear on the operator terminal session:

Operator					User
RID	ACT	ADN	Type	Location/Rack	Pool
---	---	---	---	---	---
01048	WMNT	tc10	3480	0001	

Mount the requested volume (created in steps 4-6) on the specified device.

Step 19 Normally the system detects the on-line transition when a tape is mounted. If the mount request still appears on the operator screen, manual confirmation is required. If using the full-screen request monitor, press **D** to confirm. If using the command-line request monitor, enter **rldone** to confirm.

Step 20

A message similar to the following appears on the user screen:

```
Device 1: /mnt/stortek/tlink
```

This demonstrates that when `rlaccess` completes, `tlink` is created by ConvexTMR as a symbolic link to a tape device with the proper volume mounted.

Step 21

When `rlaccess` completes, the tape is mounted and positioned to append a file. Write the file out to the tape by entering

```
cat /etc/passwd > tlink
```

Operator	User
Step 22	Release the device. Enter **rlrelease**

The operator's terminal displays output similar to the following:
`RID ACT ADN Type Location/Rack Pool`
`--- --- ------- ------ --------------- --------------`
`01049 UMNT tc10 0001`	

Step 23	Verify that the file now exists in the current directory by requesting a long directory list. Enter **rlls -l** Output similar to the following appears on the screen
`Vrwxoe>---->----> stortek lp Aug 12 10:33 test_vs` `-rwxoes---------- stortek lp UNSET Aug 12 10:38 passwd`	
Step 24	View an extended object report for the newly-created file by entering **rlls -L passwd**

Operator	User
	The following report appears on the screen

```
FILE                                                              FILE


              name: passwd

             owner: eroberts              uperm: rwxoes
             group: lp                    gperm: ------
                                          operm: ------
                                  acl entries: 0
           comment:

Location
          on volset: 2e4b964e00000001Z   volume: 2e4b965000000001V
      volset file #: 1                vol file #: 1
     file seek addr: UNSET             sections: 1

Label
           File Set: 0001                    Generation: 0
            File ID:                            Version: 0

Statistics
       record format: u:B32768:R0
         block count: 2
           file size: UNSET
          expiration: :S
             scratch: notscr

Dates
           creation: Aug 12 10:38
        last access: Aug 12 10:38
  last modification: Aug 12 10:38

Keys
                key: 2e4b978300000001F
FILE                                                              FILE
```

Step 25	Access the file for reading by entering **`rlaccess -F passwd tlink`**

Operator	User
Output similar to the following appears on the operator's screen:	

```
  RID   ACT   ADN       Type      Location/Rack      Pool
  ---   ---   -------   ------    ----------------   ---------------
01049   UMNT  tc10                0001
01051   WMNT  tc10      3480      0001

  RID   ACT   ADN       Type      Location/Rack      Pool
  ---   ---   -------   ------    ----------------   ---------------
01051   WMNT  tc10      3480      0001
```

Step 26	Mount the requested tape on the specified device and confirm the request, if necessary.	A message similar to the following appears on the user screen:

```
Device 1: /mnt/stortek/tlink
```

Step 27		Read the file back. Enter **`cat tlink > passwd.copy`**
Step 28		Release the device. Enter **`rlrelease`**
	Output similar to the following appears on the operator's screen:	

```
  RID   ACT   ADN       Type      Location/Rack      Pool
  ---   ---   -------   ------    ----------------   ---------------
01052   UMNT  tc10                0001
```

	Operator	User
Step 29		Verify that the new file is identical to the original. Enter **`diff /etc/passwd passwd.copy`**
Step 30		Retrieve the volumeset and the file written to it from the catalog by entering **`rlvsretrieve -o -V test_vs`** Output showing a submission receipt number appears on the screen: `volume 0001  receipt=R43460`
Step 31	Return the volume to its owner by entering **`rlreturn :rR43460`** The following output is displayed `Volume  :rR43460 : Complete`	
Step 32		Verify that the catalog records have been removed. Enter **`rlls -l`**
Step 33		Change directories to the parent directory. Enter **`rlcd ..`**
Step 34		Remove the /walkthrough directory. Enter

Operator	User
	`rlrmdir /walkthrough`

Step 35 Terminate the command line request monitor. Enter

`rldop -x`

The following output appears on the operator's screen:

```
Operator request server going down
```

Configuration 2

Administrator and operator groups

ConvexTMR provides separate interfaces for administrators and operators. The right to use the administrator and operator interfaces is governed by security structures called administrator and operator groups.

Administrator groups

To act as an administrator, you must enroll as an administrator in a ConvexTMR administrator group. Administrator groups define the powers available to the users enrolled in them. Each administrator group is assigned a name, a password, and one of three authority levels. The authority levels are numbered from 1 (most powerful) to 3 (most restricted). The powers you may exercise are determined by the administrator group you enroll in. The powers of each authority level are listed in Table 2.

Table 2 Administrator group powers

Authority Level	Powers
3	May perform operator level 1 tasks and May grant user privileges.
2	May perform the above tasks and • May define site options. • May edit the least privileged fields of the catalog.
1	May perform the above tasks and • May define administrator and operator groups. • May edit the most privileged fields of the catalog.

Operator groups

Operator groups function the same way as administrator groups except that they define operator powers. The powers of each authority level are listed in Table 3.

Table 3 Operator group powers

Authority Level	Powers
3	• May perform mount services.
2	May perform the above tasks and • May perform tape library management functions. • May perform tape identify overrides during tape mounts.
1	May perform the above tasks and • May start and stop the servers. • May change drive assignment and status. • May change host and domain status.

Controlling administrator and operator groups

The `rlagroup` command controls the list of administrator and operator groups. With it, you can create, edit, delete, and display group entries.

Creating groups: rlagroup

To create administrator or operator groups, follow these steps as root:

Step 1 Type `rlagroup` on the command line.

Step 2 If you are creating an administrator group, indicate so. The `-A` option signals creation of an administrator group. To create an operator group, omit `-A`.

Step 3 Specify the level of authority to assign the group. The `-l` option indicates that what follows is the authority level; in this example, level 1 is selected.

Step 4 Assign the group a password. The `-P` option indicates that what follows is the group password. In the following example, the password is "hockey."

Assign the group a name. The group name is supplied at the end of the command. You may create multiple groups with the same password and authority level by providing the group names in a space-separated list, as shown in the example below.

Step 5 Press **Return**.

For example, to create administrator groups named "commander" and "general", and extend them the highest level of authority, enter

```
rlagroup -A -l 1 -P hockey commander general
```

Warning

Sites concerned with maintaining the highest level of security should never use the rlgroup command with the -P <password> option. Whenever the -P <password> option is used, there is a slight chance that unauthorized users will be able to view the password with the ps command. If the -P <password> option is omitted, you will be prompted to enter a password.

After entering the command, the following message is returned:

```
Adding Group commander ... Complete
```

Changing groups: rlagroup -c

To change the password or authority level of a group entry, follow these steps:

Step 1 Type `rlagroup` on the command line.

Step 2 Indicate that you wish to make changes to a group. The `-c` option places the `rlagroup` command in change mode.Indicate the changes you wish to make.

Step 3 Specify the group or groups to change, by name.

Step 4 Press **Return**.

For example, to change the level and the password of the group "commander" created in the above example, enter

```
rlagroup -c -l 2 -P tennis commander
```

- `-l` (alpha) `2` changes the authority level to level 2.
- `-P tennis` changes the password to "tennis."
- To change multiple groups, provide the group names in a space-separated list.

Displaying groups: rlagroup -s

To display a list of all groups and levels, follow these steps:

Step 1 Type `rlagroup` on the command line. The -s option places the `rlagroup` command in display mode.

Step 2 Indicate that you wish to view a list of groups. When issued with no arguments, this command lists all ConvexTMR administrator and operator groups. To view specific groups, provide them at the end of the command in a space-separated list.

Step 3 Press **Return**.

For example, to view all administrator and operator groups, enter

```
rlagroup -s
```

A report of the following format appears on screen:

```
Group colonel type=admn level=1 password=cYQN...
Group commander type=admn level=1 password=cYQN...
Group general type=admn level=1 password=cYQN...
```

Deleting groups: rlagroup -d

To delete an established group, follow these steps:

Step 1 Type `rlagroup` on the command line.

Step 2 Indicate that you wish to delete a group. The `-d` option places the `rlagroup` command in delete mode.

Step 3 Specify the group or groups to delete, by name. To delete multiple groups, provide the group names in a space-separated list.

Step 4 Press **Return**.

For example, to delete the groups colonel and general, enter

```
rlagroup -d colonel general
```

The following message appears on screen:

```
Deleting Group colonel ... Deleted
Deleting Group general ... Deleted
```

Configuring the server: dev_config

ConvexTMR server parameters are configured via the `dev_config` file. The server reads `dev_config` on start-up. The default location for `dev_config` is *LIBDIR*`/REEL/Librarian/dev_config`, where *LIBDIR* is the `RLLIBDIR` value specified in the file `/etc/reelenv`.

`dev_config` consists of several declarations, one per line. None of the declarations are required, and the order of the declarations in the file is unimportant. There are two types of declarations:

- RLDV
- ODS

RLDV declaration: RLDV *dev_map*

The RLDV declaration directs the server to use the named file as the source for device configuration. The configuration file must be in the format produced by the `rldev`(1) command.

To create and activate a custom *dev_map*, follow these steps:

Step 1 Write the output of `rldev -C` to a file. The utility *LIBDIR*`/REEL/io2devmap` will print all devices listed in the spu file `/ioconfig` to the screen in a format suitable for devmap.

Step 2 Edit the file to remove the device entries you want excluded from ConvexTMR.

Step 3 Make an RLDV entry in `dev_config` with the full pathname of the edited file.

ODS declaration: ODS *domain*

The ODS declaration starts the named operator domain. Operator domains are configured via a separate file. This is explained in the next section, *Defining operator domains*.

dev_config examples

The example below displays a simple server configuration:

```
RLDV /usr/lib/REEL/Librarian/dev_map
ODS default

ODS silo
```

Defining operator domains: dom_map

ConvexTMR employs the concept of operator domains. An operator domain is a declaration of operational scope and capabilities. Domains divide operational responsibility among multiple operators, whether they are human or robotic.

ConvexTMR has a default domain that covers all devices and capabilities. The default domain requires a human operator. Human operators view domain activities via the request monitor. ConvexTMR also allows for the configuration of custom domains.

Operator domains are stored in the dom_map file. dom_map resides in *LIBDIR*/REEL/Librarian/dom_map, where *LIBDIR* is the `ŘLLIBDIR` value specified in the file /etc/reelenv.

dom_map declarations

The `dom_map` file contains multiple declarations, one per line. The order of the declarations is insignificant, with the exception of the `domain` declaration, which always comes before other associated declarations to announce and name the domain.

Domains can overlap, which means that multiple domains can simultaneously declare capabilities for one or more devices or vaults.The declaration `prio` determines which domain has priority. Requests with equivalent priorities are handled on a first-in-first-out basis.

All possible domain declarations are listed in Table 4.

Table 4 Domain declarations

Declaration	Function
domain *domain_name*[1] [2]	Announces the domain definition and names the domain.
prio *priority*[3]	Determines the priority of the domain.
`mount` *dev_name*[4]*/vault*[5]	States the ability to mount tapes in the specified vault on the specified device.
`unmount` *dev_name/vault*	States the ability to unmount tapes.
`scrmnt` *dev_name/vault*	States the ability to find a scratch tape and mount it.
`vadd` *vault*	States the ability to mount a new, dynamically added scratch tape.
`vcheck` *dev_name*	States the ability to reply with the external label for a mounted tape.
`wcheck` *dev_name*	States the ability to reply with the write protection status of a mounted tape.
`dcheck` *dev_name*	States the ability to reply with the status of a device error.
`tcheck` *dev_name*	States the ability to select the recording format for the device.
`vacc` *vault*	States the ability to verify a user's right to access a volume.
`stage` *vault*	States the ability to pre-fetch and stage tapes.

Table 4 Domain declarations **(continued)**

Declaration	Function
`devup` *dev_name*	States the ability to mark a device as up.
`devdown` *dev_name*	States the ability to mark a device as down.
`stack` *dev_name/vault*	States the ability to load a stacker device.
scrstack *dev_name/vault*	States the ability to load a stacker device with scratch tapes.
`unstack` *dev_name*	States the ability to unload a stacker device.

[1]*domain_name* names the domain.
[2][*nexec robo_program*] may also be specified to direct operator requests to a script that communicates with a robot.
[3]*priority* determines which domain receives the request between overlapping domains. Range is 0 - 9, with 0 receiving highest priority.
[4]*dev_name* in a domain declaration is the ConvexOS or SPP-UX stanza name of the tape device. The special value ANY can be used to indicate any tape device.
[5]*vault* is the name of a storage vault. The special value ANY indicates any vault.

dom_map flags

Each `dom_map` capability declaration may be proceeded by an `-A` or an `-N`. The `-A` flag indicates that ConvexTMR always assumes success when this action is taken, and the operator is not notified; the `-N` flag indicates that failure is assumed. For example, the declaration

```
unmount -A ANY
```

indicates that `unmount` requests for all devices always succeed.

dom_map example

The following example sets up two domains, `default` and `special`. Note that the second domain will take priority for requests in which the domains overlap. Also note that the

second domain directs operator requests to the file `control`, a script which communicates with a tape robot.

Example: The /usr/lib/REEL/Librarian domain map

```
domain default
mount ANY/ANY
unmount ANY/ANY
scrmnt ANY/ANY
devup ANY
devdown ANY
vcheck ANY
wcheck ANY
dcheck ANY
tcheck ANY
vadd ANY

domain special 1 control
prio 1
mount tc0/ANY tc1/ANY
unmount tc0/ANY tc1/ANY
scrmnt tc0/ANY tc1/ANY
devup tc0 tc1
devdown tc0 tc1
vcheck tc0 tc1
wcheck tc0 tc1
dcheck tc0 tc1
tcheck tc0 tc1
vacc -A ANY
stage ANY
```


Defining rlaccess defaults: mmr_map

ConvexTMR supports site-configurable defaults for the following `rlaccess` and `rlnext` options:

- device model
- media type
- recording format

Default definitions for these options are stored in the file `mmr_map`. The `mmr_map` file resides in *LIBDIR*`/REEL/Librarian/mmr_map`, where *LIBDIR* is the `RLLIBDIR` value specified in the file `/etc/reelenv`.

`mmr_map` consists of one or more declarations, one per line, of the format:

dmodel mtype rformat

where *dmodel* is the device model, *mtype* is the media type, and *rformat* is the recording format.

The `mmr_map` provides default values for `rlaccess` and `rlnext` commands that do not specify *dmodel*, *mtype*, and *rformat*. Default values are selected by searching for a match in the mmr_map between any user-supplied options and the declarations.

ConvexTMR examines each declaration in order starting from the top of the file and considers a declaration to match if the user-specified values match the values in the declaration.

Note

If the user did not specify any of the three options, the first declaration is a match, and it determines the command defaults.

Consider the following `mmr_map` file:

```
3480 3480 3480
3490 3480 3480
3490 3490 3490
```

And `rlaccess` command:

```
rlaccess -d 3490
```

Because the command only specifies a value for *dmodel*, defaults for the command are supplied by the second line of the `mmr_map`. The *mtype* for this command defaults to `3480`; the *rformat* defaults to `3480`.

Defining media types: mtype_map

ConvexTMR characterizes tapes by media type. The media type identifies the storage capacity and drive compatibility of the tape. All tapes entered into the ConvexTMR library need a media type attribute.

Media types are defined in the file `mtype_map`. The `mtype_map` resides in *LIBDIR*`/REEL/Librarian/mtype_map`, where *LIBDIR* is the `RLLIBDIR` value specified in the file `/etc/reelenv`.

`mtype_map` consists of one or more declarations, one per line, of the format:

index mtype_name length rformat `[...]`, where

index is an integer representing the place of the declaration among all declarations, beginning with 1.

mtype_name is the name of the media type. It must be unique among all media type declarations.

length is the storage capacity of the media. For round or reel tapes it is measured in feet. For all others, it is measured in megabytes.

rformat is the recording format the media type supports. Up to six recording formats may be specified for each media type. Recording formats are defined in the file `record_map`, which is discussed in the next section, *Defining recording formats.*

The following `mtype_map` declares three media types: round, 3480, and eight millimeter. Note that each media type supports more than one recording format.

```
1 round 2400 800 1600 3200 6250
2 3480 500 3480 3490
3 8mm 2000 8mm 8mmh
```


Defining Recording Formats: record_map

Many types of tape media can allow data recording in various ways. For example, reel tapes can be written in 800 bpi,1600 bpi, or 6250 bpi. For this reason, ConvexTMR maintains a list of all possible recording formats.

Recording format definitions are stored in the file `record_map`. The `record_map` resides in *LIBDIR*`/REEL/Librarian/record_map`, where *LIBDIR* is the `RLLIBDIR` value specified in the file `/etc/reelenv`.

The `record_map` is a list of recording format definitions, of the format:

index rformat [*alias...*], where

index is an integer representing the place of the declaration among all declarations, beginning with 1.

rformat is an alphanumeric name up to twelve characters long which uniquely identifies the format.

alias is another name for the same format.

The following `record_map` lists 12 recording formats. Note that each declaration only lists one recording format.

```
1 800 nrzi NRZI
2 1600 pe PE
3 3200
4 6250 grc GRC
5 qic150
6 8mm XBYTE
7 8mmh XBYTEH
8 dat DAT 4mm
9 dath DATH 4mmh
10 3480
11 3490
12 disk
```

Defining storage vaults: vault_map

ConvexTMR's tape library can be stored at one or more sites, or storage vaults. Users and operators can direct ConvexTMR to move tapes between the different vaults.

Vault names are stored in the file `vault_map`. The `vault_map` resides in *LIBDIR*`/REEL/Librarian/vault_map`, where *LIBDIR* is the `RLLIBDIR` value specified in the file `/etc/reelenv`.

`vault_map` is a list of all vault names, of the format:

index vault

`index` is an integer representing the place of the declaration among all declarations, beginning with 1.

`vault` is an alphanumeric name up to twelve characters long which uniquely identifies the vault.

The following `vault_map` defines three different storage vaults.

```
1 onsite
2 offsite
3 warehouse
```

Defining user fees: dev_cost

Many ConvexTMR sites charges fees for tape drive usage and mount actions. User fees are stored in the `dev_cost` file.

The `dev_cost` file resides in *LIBDIR*`/REEL/Librarian/dev_cost`, where *LIBDIR* is the `RLLIBDIR` value specified in the file `/etc/reelenv`.

`dev_cost` contains multiple declarations, one per line. Each line must be of one of the following formats:

`d_`*dname resource_cost mount_cost*

m_*model resource_cost mount_cost*

DEFAULT *resource_cost mount_cost*, where

resource_cost is in units of cents per minute and represents the charge to the user for each minute the drive is allocated to the user.

mount_cost is in units of cents per mount and represents the charge to the user for each tape mount performed on behalf of the user.

d_*dname* declares the costs for the specific drive name.

m_*model* declares the costs for all drives of the particular model. Declarations for specific drives overrides any model declaration.

DEFAULT declares the costs for any drives not covered by any other declaration.

The following `dev_cost` file shows a default charge for all drives of $0.10 for each minute and $1.00 for each tape mount.

```
DEFAULT 10 100
```

The following `dev_cost` file shows different charges for different drives. All 3480 model tape drives have a $0.20 cost per minute and a $1.00 mount charge. Tape drive `drive1` has a $0.20 cost per minute and a $2.00 mount charge. All other devices default to $0.10 for each minute and $1.00 for each tape mount.

```
m_3480 20 100
d_drive1 20 200
DEFAULT 10 100
```

Defining site options: rlcop

Many ConvexTMR parameters can be customized according to the needs of your site. Options that may be customized include timing intervals, naming conventions, and global defaults. These options are set and displayed with the `rlcop` command. For a complete description of this command, refer to the rlcop(8) man page.

To change site options via `rlcop`, follow these steps:

Step 1 Write the output of `rlcop -s` to a file.

```
rlcop -s > foo
```

Step 2 Edit the contents.

```
vi foo
```

Step 3 Invoke the modified settings.

```
rlcop -r foo
```

Determining the external label alphabet: exalpha=

The `exalpha=` keyword of `rlcop` allows you to specify the acceptable characters for volume external labels.

The device display panel of 3480 or 3490 devices is not capable of displaying the full set of ASCII characters. If your site uses 3480 or 3490 devices, you should limit the acceptable external label alphabet to characters that the display panel can accommodate. This character set is shown in Table 5.

Table 5 Acceptable characters for volume external labels

Type	Acceptable characters
Alphabetic	A through Z (no lower case)
Numeric	0 1 2 3 4 5 6 7 8 9
National	@ $ #
Special	, . / ' ( ) * & + - =
Other	% | : _ > < ? ;

To limit the alphabet of acceptable external label characters, follow these steps:

Step 1 Follow the first two steps listed for defining site options on the previous page.

Step 2 List the characters you wish to allow at the `exalpha=` keyword.

Step 3 Save the file.

Step 4 Invoke the modified settings.

The following example limits the external label alphabet to characters which will display on the IBM 3480 and 3490 device.

After writing the output of `rlcop` to a file, type the following string at the `exalpha=` keyword:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789@$#,./'()*&
+-=%|:_><?;
```

After the new site constants are invoked, users who supply invalid external labels will receive a message similar to the following:

```
External label alphabet does not fall within
site defined alphabet
```

If using 3480 and 3490 devices, you should also set the `exmaxc=` keyword (external label maximum length) to no greater than 6; 6 character external labels are the longest labels the panel can accurately display.

User privileges defined

As a ConvexTMR administrator, you control the privileges of all ConvexTMR users and groups. These privileges are listed below.

- requesting priority levels
- bypass label processing (BLP)
- modifying the BLP permission mask
- making requests on behalf of others
- selecting specific devices
- creating media pools
- exceeding site limits

These privileges are granted, revoked, and displayed with the `rlauth`(8) command. Use of this command is explained in the next section, "Granting user privileges". This section defines each of these privileges in detail.

Priority request privilege (Prio=*n*)

Each user access request has a priority attribute. The priority determines the order in which ConvexTMR satisfies the request. The priority is an integer which ranges from 1 to 9; 1 is the highest priority. By default, user requests are queued and processed in the order in which they are received.

If you extend priority request privilege to a user or group, that user or group may assign a priority to a `rlaccess` request. When you extend this privilege, you determine the highest priority level that user or group may request. For example, if you assign user `ralph` priority request level 3, `ralph` may assign any of priority levels 3 - 9 to any `rlaccess` or `rlnext` request.

Bypass label processing privilege (BLP)

This group of privileges allows users and groups to access IBM- and ANSI-labeled tapes as if they were unlabeled tapes. In BLP mode, all label files are accessible as data files, and volume and

file labels are ignored. You may extend users and groups the privilege to perform any or all of the following activities:

- BLP reads on owned volumes
- BLP writes to owned volumes
- BLP reads on all volumes
- BLP writes to all volumes
- BLP permission mask modification to owned volumes

In order for a user to perform BLP reads or writes, the associated BLP bits in the ACL of the object of the request must be set. For more information, refer to the rllsacl(1) and rlchacl(1) man pages and to the *ConvexTMR User's Guide*, Chapter 3.

BLP access to owned volumes (BLPOR, BLPOW)

Users and groups with read and/or write BLP access to owned volumes may perform BLP reads and/or writes to any owned volumes.

BLP access to all volumes (BLPAR, BLPAW)

Users and groups with BLP read and/or write access to all volumes may make BLP read or write requests to all volumes in the ConvexTMR library for which they have been granted permission by the owner via the access control list. For more information, refer to the rllsacl(1) and rlchacl(1) man pages and to the *ConvexTMR User's Guide*, Chapter 5.

Modify BLP permission mask (BLPCM) (catalog only)

This option is only available at sites that operate with a catalog.

The last field of the ACL display (requested with the `rllsacl` command) for a volume object shows the permissions. A `b` in this string indicates BLP read permission for the specified user or group; a `B` indicates BLP write permission for the specified user or group. Users and groups with this privilege may change these permissions for owned volumes via `rlchacl`. For more information, refer to the rlchacl(1) man page and to the *ConvexTMR User's Guide*, Chapter 3.

Third party reservation privilege (OBO)

This privilege allows users to issue access requests on behalf of other users.This allows users to take on some of the request load and reduces the amount of trivial work for the operators.

Physical device request privilege (PDA)

By default, users may only ask for types of drives, not specific units.This privilege allows users to request access to a specific tape device by name.

Pool creation privilege (POOL)

All users can use tapes from tape pools to which they have been granted access. This privilege allows users to create new tape pools and grant access to the pool to other users.

Exceed site resource limit privilege (ESL)

ConvexTMR limits the maximum number of tape drives a single user or job can have in use at any one time. This privilege allows users to exceed the site limits at any time.

Granting user privileges: rlauth

`rlauth` controls the user privileges described in the previous section. You may grant privileges to individual users, to groups, and to all users.

Display user privileges: rlauth -s

To view user privileges, follow these steps:

Step 1 Type `rlauth` on the command line.

Step 2 Indicate that you wish to view privileges. The `-s` option requests a `rlauth` report.

Step 3 If you wish to view the privileges of a group, indicate that. To view the privileges of a group, use the `-g` *name* option.

Step 4 If you wish to view the privileges of an individual, specify the individual by name. In the example below, `ralph` is the user ID of the user whose privileges you wish to view. To view the privileges of all users, do not specify a name.

Step 5 Press **Return**.

The following example requests a report of the privileges for user `ralph`.

Enter

```
rlauth -s ralph
```

The following message is displayed:

```
ralph: Prio=5
Auths=BLPAR,BLPAW,BLPOR,BLPOW,BLPCM
```

Authorization codes for the `rlauth` command are defined in Table 6.

Table 6 `rlauth` authorization codes

Code	Privilege
`Prio=`*n*	May request up to level *n* priority handling
`BLPAR`	May make BLP read access to all volumes
`BLPAW`	May make BLP write access to all volumes
`BLPOR`	May make BLP read access to owned volumes
`BLPOW`	May make BLP write access to owned volumes
`BLPCM`	May change BLP permission mask
`OBO`	May make requests on behalf of others
`PDA`	May access specific physical devices
`POOL`	May create media pools
`ESL`	May exceed site resource limits
`ALL`	May perform all of the above tasks

Set priority request level: rlauth -p*best_priority*

To set a priority request level, follow these steps:

Step 1 Type `rlauth` on the command line.

Step 2 Indicate that you wish to set a priority level. The `-p`*n* argument indicates that you wish to change a request priority. In the following example, the request priority is being changed to the highest priority level, 1.

Step 3 Specify the level.

Step 4 Name the user or group to affect. In the example below, `ralph` is the ID of the user that is the object of the request. To change the priority level of a group, use the `-g` option.

Step 5 Press **Return**.

The following example changes the priority level for user ralph.

```
rlauth -p1 ralph
```

The following message is displayed:

```
ralph ... Complete
```

Grant user privileges: rlauth -a *auth*(, ...)

To grant a user privilege, follow these steps:

Step 1 Type `rlauth` on the command line.

Step 2 Indicate that you wish to grant a privilege. The `-a option` indicates that you wish to grant user privileges. In the example, below, tape pool creation and third-party reservation privileges are granted.

Step 3 Specify the privilege you wish to grant, by authorization code.

Step 4 Name the user or group to affect. In the example below, `ralph` is the ID of the user that is the object of the request. To grant privileges to a group, use the `-g` option.

Step 5 Press **Return**.

The following example extends privileges to user `ralph`.

Enter

```
rlauth -aPOOL,OBO ralph
```

The following message is displayed:

```
ralph ... Complete
```

Revoke user privileges: rlauth -a-*auth*(, ...)

To revoke a user privilege, follow these steps:

Step 1 Type `rlauth` on the command line.

Step 2 Indicate that you wish to revoke a privilege. `-a-` indicates that you wish to revoke user privileges.

Step 3 Specify the privilege you wish to revoke, by authorization code. In the example below, tape pool creation and third-party reservation privileges are revoked.

Step 4 Name the user or group to affect. In the example below, `ralph` is the ID of the user that is the object of the request. To revoke group privileges, use the `-g` option.

Step 5 Press **Return**.

The following example revokes privileges of user `ralph`.

Enter:

```
rlauth -a-POOL,OBO ralph
```

The following message is displayed:

```
ralph ... Complete
```

Delete all privileges: rlauth -d

To delete all privileges for a specified user, follow these steps:

Step 1 Type `rlauth` on the command line.

Step 2 Indicate that you wish to delete all privileges. The `-d option` indicates that you wish to delete user privileges.

Step 3 Specify the privilege you wish to revoke, by authorization code.

Step 4 Name the user or group to affect. In the example below `ralph` is the ID of the user that is the object of the request. To delete the privileges of an entire group, use the `-g` option.

Step 5 Press **Return.**

The following example deletes all the privileges of user `ralph`.

Enter

```
rlauth -d ralph
```

The following message is displayed:

```
ralph ... Complete
```

Configuring tape devices: rldev

The `rldev` command is a powerful command that provides a variety of device control services. For a comprehensive list of `rldev` functions, refer to the rldev(8) man page.

When issued with no options, `rldev` returns a current device status report. A sample `rldev` report is shown below.

Figure 4 Device list report

```
Current device status;
   ①       ②       ③        ④     ⑤       ⑥          ⑦     ⑧  ⑨       ⑩
Device    Model   Status    Tape  INTLBL  EXTLBL      ReqID  Dn User     Key
------    -----   ------    ----  ------  ------      -----  -- ----     ---
drive3    3480    nitemare  idle  ------  ------
drive2    3490    nitemare  idle  ------  ------
drive1    3490    nitemare  idle  ------  ------
```

Report columns are defined below.

① Tape Drive Name — The name of the tape device.

② Device Model — The tape device model.

③ Tape Device Status — The status of the tape device. In this figure "nitemare" is the *_hostname* that the shared device is currently attached to.

④ Tape Volume Status — The status of the volume. These states are defined in the table below.

⑤ Internal Label — The internal label of the mounted volume.

⑥ External Label — The external label of the mounted volume.

⑦ Request ID — The request ID of the request engaging the specified device.

⑧ Device Number — The device number in a multiple drive request. For requests using only one drive, this field is blank.

⑨ User — The ID of the user of the device.

⑩ Resource Key — The resource key associated with the request, if applicable.

Changing device status: rldev -s status=*state*

At times, you will need to take a device out of service for maintenance or diagnostics.

To change the status of a device, follow these steps:

Step 1 Type `rldev` on the command line.

Step 2 Indicate that you wish to change the status attribute. The `-s` option indicates that you wish to change a device attribute.

Step 3 Set the attribute. The keyword `status=` indicates the device status. The value `down` takes the device out of service. Values for `status=` include

- `up` - makes the device available for use.
- down - the device is taken down immediately. If the device is in use, the session is aborted and the user is notified.
- `pdown` - brings the device down as soon as it is free.
- `reset` - causes a software reset. If the device is in use, the session is aborted and the user is notified.

Step 4 Specify the device. The device name is supplied immediately after the keyword value; in this example, `device1` is affected. More than one device may be specified with a space-separated list.

Step 5 Press **Return**.

For example:

```
rldev -s status=down tc7
```

Note

There can be no spaces in status=*state*

A message similar to the following is displayed:

```
Device: tc7 Updating...Device request accepted
```

Configuring the ACS silo

TThe CONVEX Tape Library Interface (TLI) allows CONVEX C2, C3, and C4 Series machines to connect to the StorageTek automated cartridge system (ACS), commonly referred to as the *silo*. The silo is an automatic cartridge system that stores and tracks the location of thousands of tape cartridges. Within the silo, a robot arm mounts and unmounts tapes on the available drives as they are needed without operator intervention.

When a tape request is made, the electromechanical robotic arm locates the requested tape by bar code and delivers it to the cartridge tape drive or passes it to another silo. When the tape is no longer needed, it removes the tape from the drive and returns it to its library slot.

This section describes the steps you must take to configure the system for the silo.

Step 1 Set the `SILOHOST` environment variable.

To use the commands necessary to manage the silo system, you must set your environment variable, `SILOHOST`, to the host name of the Sun workstation that controls the silo on your CONVEX system. For example, if you use `csh`, and the host name of the Sun CPU that controls the silo is "jupiter," enter

```
setenv SILOHOST jupiter
```

If you use `sh` or `ksh`, add the following line to your .login or .profile file, depending on the shell you use:

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

Step 2 Define all drives and devices associated with the silo to the tape system. This is done by invoking the *LIBDIR*/REEL/io2dev_map script. The script reads the ioconfig file on the SPU and creates device entries in the *LIBDIR*/REEL/Librarian/dev_map.

Step 3 Get drive coordinates for each of these drives from your StorageTek representative. Drive coordinates define to the mechanical tape operator (robotic arm) the position occupied by the tape drive in the silo. This enables the mechanical tape operator to mount and unmount cartridges on the drive, as well as to report its status. The drive coordinates are four integers separated by commas, as in

```
0,0,0,0
```

Step 4 Map the tape devices to silo tape drives.

The file /usr/convex/robot/config/silodrivelist maps the silo's tape drives' special device files to a silo drive coordinate, which

allows the device file to access the associated drive. The silodrivelist file is a simple ASCII file. You must create this file before using ACS tape commands. Figure 5 illustrates an example silodrivelist file.

Figure 5 Example silodrivelist file

```
tc0                 0,0,10,0
tc:0                0,0,10,0
/dev/rtc0           0,0,10,0
/dev/rtc0n          0,0,10,0
/dev/rtc0nu         0,0,10,0
/dev/rtc0u          0,0,10,0
tc1                 0,0,10,2
tc:1                0,0,10,2
/dev/rtc1           0,0,10,2
/dev/rtc1n          0,0,10,2
/dev/rtc1nu         0,0,10,2
/dev/rtc1u          0,0,10,2
```

Each line in this file represents one entry. Each entry is a CONVEX tape drive or special device file name followed by the silo drive coordinates. Name and coordinates are separated by a space or tab. Each coordinate is separated by a comma. No comments are allowed.

Step 5 Add the devices to the *LIBDIR*/REEL/Librarian/dom_map file. Refer to "Defining operator domains: dom_map" section on page 34 for instructions on how to do this.

Step 6 Add the following lines to /etc/rc.local (see figure 6) to start the silodaemon(8). Replace `silohost` with the name of the workstation that controls the silo.

Figure 6 Silo daemon lines to add to /etc/rc.local

```
if [-f /usr/convex/robot/bin/silodaemon]; then
   /usr/convex/robot/bin/silodaemon -h silohost &
fi
```

The silodaemon starts a storage server interface (ssi) daemon to communicate with the StorageTek server.

Step 7 Add an entry to /etc/services identifying the port number that the ssi daemon monitors.

The line should be similar to the following:

```
silo    50004/tcp   #silodaemon pid
```

Note **If no port is specified, port number 50004 is used by default.**

Administration 3

Enrolling as a tape administrator

You may not enroll as an administrator until an administrator group has been configured. This is done via the rlagroup(8) command. For more information on configuring administrator groups, see "Administrator and Operator Groups" in Chapter 2.

In order to perform administrator tasks, you must first enroll. To enroll as a tape administrator, follow these steps:

Step 1 Type `rladm` on the command line.

Step 2 Enter the name of the group you intend to enroll in.

Step 3 Press **Return**. A password prompt should appear.

Step 4 Type your password.

Step 5 Press **Return**.

A sample session in Chapter 2 created the administrator group "commander" with the password "hockey." For example, to enroll in this group, enter

```
rladm commander
```

The following message is displayed:

```
Password:
```

Enter

```
hockey
```

- You may enter the administrator group and password in the same command. To do this, include the `-P` *password* option in the command before typing the group name. For example, enter:

```
rladm -P hockey commander
```

Warning

Sites concerned with maintaining the highest level of security should never use the rladm command with the -P <password> option. Whenever the -P <password> option is used, there is a slight chance that unauthorized users will be able to view the password with the ps command. If the -P <password> option is omitted, you will be prompted to enter a password.

Requesting an enrollment summary: -s

To view a list of all currently enrolled administrators and operators and verify that your enrollment was successful, follow these steps:

Step 1 Type `rladm` on the command line.

Step 2 Indicate that you wish to view an enrollment summary. The `-s` option requests an enrollment summary.

Step 3 Press **Return**.

For example, enter

```
rladm -s
```

A report similar to Figure 7 appears on screen.

Figure 7 Enrollment summary

```
①                    ②      ③          ④  ⑤
ID   User      Type   Group      L  Message Destination
--   ----      ----   -----      -  -------------------
100  lfw       Admin  commander  1  _NO_MSG
```

① The number recorded in the `ID` column is the request ID of the login session.

② The `Type` column indicates if the group specified is an operator or administrator group.

③ The `Group` column displays the group name.

④ The `L` column indicates the authority level of the specified group.

⑤ The `Message Destination` column shows where operator messages will be displayed or stored; in this example, messages are not sent.

Resigning your enrollment: -x

To resign an administrator enrollment, follow these steps:

Step 1 Type `rladm` on the command line.

Step 2 Indicate that you wish to resign an enrollment. The `-x` option requests that the specified administrator enrollment be cancelled.

Step 3 Identify the enrollment you wish to cancel by ID number. The `-I` option indicates that the enrollment session to cancel will be identified by session ID number; in this example, session 100 (from the previous example) is selected. If you do not know the ID number of the enrollment you wish to cancel, request an enrollment summary by following the steps outlined in the previous section, "Requesting an Enrollment Summary."

Step 4 Press Return.

For example:

```
rladm -x -I 100
```

Message options

When you enroll in an administrator group, you may select a destination for administrator messages. By default, no messages are sent unless you specify a destination.

ConvexTMR offers four message destination options. Messages may be sent to the following destinations:

- the terminals of all currently active administrators (`-t`)
- the specified mail address (`-m` *mail_id*)
- the specified pathname (`-p` *path*)
- the system log (`-S`)

Only one message destination may be specified with each `rladm` command. Additional `rladm` commands may be issued to direct messages to more than one destination.

Route messages to administrator terminals: -t

To send administrator messages to the terminals of all currently active administrators (`/dev/tty`), use the `-t` option when issuing the `rladm` command.

For example,

```
rladm -t -P hockey commander
```

Route messages to a specified address: -m *mail_id*

To send administrator messages via electronic mail to a specified address, use the `-m` option when issuing the `rladm` command.

For example,

```
rladm -m fred -P hockey commander
```

Route messages to a specified pathname: -p *path*

To send administrator messages to a specified pathname, use the `-p` option when issuing the `rladm` command.

For example,

```
rladm -p /home/lfw/msgfile -P hockey commander
```

Warning

Sites concerned with maintaining the highest level of security should never use the rladm command with the -P <password> option. Whenever the -P <password> option is used, there is a slight chance that unauthorized users will be able to view the password with the ps command. If the -P <password> option is omitted, you will be prompted to enter a password.

Route messages to the system log: -S

To send `rladm` messages to the system log, use the `-S` option when issuing the `rladm` command. Messages are sent to `syslog()` using the `LOG_TMS` facility.

For example,

```
rladm -S -P hockey commander
```

Using the accounting data

ConvexTMR generates tape accounting records. By default, these records are sent to the file `/var/adm/tapeacct`. The destination file is controlled by the `acctfile=` site constant, and can be changed via `rlcop`.

ConvexTMR accounting records can be generated in standard ASCII or binary format. This is controlled by the `binacc=` site constant, and can be changed via `rlcop`.

Database management

The information in this section is only correct for sites that operate with a catalog.

ConvexTMR provides two utilities for maintaining the integrity of the catalog records: `rlrebuild` and `rldbck`. In addition, the database must be periodically backed up and monitored for file system space use.

Recreating database keys: rlrebuild

If the system crashes, database keys can become corrupted. The ConvexTMR utility `rlrebuild` is available to recover the catalog and recreate database keys in the event of a power failure or bad media.

`rlrebuild` works in two steps. First, `rlrebuild` creates a new database from the records recovered from the most recent backup. Second, `rlrebuild` applies the transactions recorded in the journal to the newly-created database.

If your database keys become corrupted, system messages will alert you to the situation.

To recover the ConvexTMR catalog and restore database keys, follow these steps:

Step 1 According to your site's operating procedures, recover the most recent copy of the ConvexTMR database. This is stored in *LIBDIR*`/REEL/Librarian/Tdb`, where *LIBDIR* is the `RLLIBDIR` value specified in the file `/etc/reelenv`.

Step 2 According to your site's operating procedures, recover the database journal files created since the last backup. These are located in the directory specified by the `dbjdir=` site constant.

Step 3 Issue the `rlrebuild` command.

Note

The server must be down in order to run the `rlrebuild` command.

Step 4 If you wish to check the integrity of the new database, issue the `rldbchk` command. See the next section, *Checking Database Logic*, for details and instructions.

For example, with the server down, type

```
rlrebuild
```

Messages similar to the following appear on screen:

```
Old DB time stamp: Mon Aug 30 16:00:00 1993
Starting New DB
```

(Initialization messages are listed.)

```
Entering records from old DB
```

(Records are listed.)

```
Applying Journal transactions
```

(Transactions are listed.)

```
Database rebuilt to: Tue Aug 31 16:53:09 1993
```

- Prior to issuing the `rlrebuild` command, you may wish to view a file summary to verify that all the journal files required for a rebuild are present. To do this, enter

  `rlrebuild -t`

- You may wish to delete all but the most recent journal file displayed by the output of `rlrebuild -t`. To do this, enter

  `rlrebuild -B`

- If your attempt to rebuild the database is aborted, use the `rlrebuild -u` option for the next rebuild attempt; this employs the old database rather than the partially-created, newer database.

- Several other options are available with the `rlrebuild` command; for a complete description of the command, refer to the rlrebuild(8) man page.

Checking database logic: rldbck

The utility `rldbck` checks the databases for logical soundness and reports and/or fixes errors that are encountered. Potential database logic problems include:

- pools, rotations, or directory objects that do not have directory entries
- unreferenced parent directories
- directories without parents
- missing ConvexTMR root, /pool or /rotations directories
- volumes that reference nonexistent pools
- volumesets that reference nonexistent rotations

- files that reference nonexistent volumes or volumesets
- volumes that reference nonexistent volumesets
- discrepancies between the actual pool volume/scratch volume counts and the pool record

To check the database and fix errors, issue the `rldbck` command with no options. The system will ask you if you wish to fix errors as they are encountered.

Several other options are available with the `rldbck` utility, including options to fix errors without prompting, report errors without fixing, and view memory table data. Refer to the rldbck(8) man page for a complete description of the command.

Backing up the ConvexTMR catalog

The ConvexTMR catalog database must be backed up periodically. This can be done using /bin/dd or ConvexOS /etc/dump. The most recent copy of the ConvexTMR database is stored in *LIBDIR*`/REEL/Librarian/Tdb`, where *LIBDIR* is the `RLLIBDIR` value specified in the file `/etc/reelenv`.

To back up the ConvexTMR catalog database, follow these steps:

Step 1 Set the request queue (Qmode) to "refuse." Enter

```
rlrm -q refuse -O "Request queue down. \
Catalog backup in progress"
```

Step 2 Check the processing queue by typing `rlrm` followed by a **Return** on the command line.

Step 3 Repeat step two until you can confirm that the processing queue is empty and the ConvexTMR system is in a quiescent state.

Step 4 Use /bin/dd or /etc/dump to backup the *LIBDIR*`/REEL/Librarian/Tdb` directory to tape or another file system.

Step 5 Set the request queue (Qmode) to "queue." Enter

```
rlrm -q queue -O "Catalog backup completed. \
Request queue is up"
```

Restoring the ConvexTMR catalog

If necessary, the ConvexTMR catalog database can be restored from a backup. This can be done using /bin/dd or ConvexOS /etc/restore. This would be necessary if the ConvexTMR utilities `rldbck` and `rlrebuild` fail to restore the existing database to a sound state.

To restore the ConvexTMR catalog database, follow these steps:

Step 1 Set the request queue (Qmode) to "refuse." Enter

```
rlrm -q refuse -O "Request queue down. Catalog
restore in progress"
```

Step 1 Set the processing queue (Pmode) to "hold." Enter

```
rlrm -p hold -O "Restoring catalog"
```

Step 2 Use /bin/dd or /etc/restore to restore the *LIBDIR*/REEL/Librarian/Tdb directory from the most recent backup of the catalog on tape or from another file system where it is stored.

Step 3 Set the processing queue (Pmode) to "process." Enter

```
rlrm -p process -O "Catalog restore completed"
```

Step 4 Set the request queue (Qmode) to "queue." Enter

```
rlrm -q queue -O "Catalog restore completed.
Request queue is up"
```

Monitoring ConvexTMR catalog file system use

The ConvexTMR catalog will grow as more volumes are created and/or are placed under ConvexTMR control. It is possible to eventually use up all of the available space in the file system. When this happens ConvexTMR will not be able to use the catalog to track any new information during normal operations. Therefore, the catalog database directory must be monitored periodically in anticipation of this event. It is a good idea to place the catalog in a separate file system mounted on LIBDIR/REEL/Librarian/Tdb. This will make maintenance easier if you need to increase file system space.

Using the log files

ConvexTMR employs comprehensive logging to allow you to track system activity. ConvexTMR log files are located in the directory *LIBDIR*/REEL/logs, where *LIBDIR* is the RLLIBDIR value specified in the file /etc/reelenv.

Five ConvexTMR logs are available:

- administrator log
- operator log
- user log
- job log
- system log

The administrator log: REEL/logs/ADMIN

The administrator log records all administrator messages. The date, time, system message number and message text are displayed for each message, in chronological order. To keep a constant, dynamically updated administrator log on display, issue the following command from a separate terminal session (substitute your RLLIBDIR value for *LIBDIR*):

```
tail -f LIBDIR/REEL/logs/ADMIN
```

A sample administrator log is shown in Figure 8.

Figure 8 Administrator log

```
Jun 18 13:43:25> RL5135: System Signal(11) - rlsvc Aborting
Jun 18 13:43:29> RL5270: Could not raise priority
Jun 18 13:45:31> RL5058: Mount request system started
Jun 18 13:45:31> RL5270: Could not raise priority
Jun 19 11:01:07> RL5058: Mount request system started
Jun 19 11:01:07> RL5270: Could not raise priority
```

The operator log: REEL/logs/OPER

The operator log records all operator messages, including device configuration messages. The date, time, system message number and message text are displayed for each message, in chronological order. To keep a constant, dynamically updated operator log on display, issue the following command from a separate terminal session (substitute your RLLIBDIR value for *LIBDIR*):

`tail -f` *LIBDIR*`/REEL/logs/OPER`

A sample operator log is shown in Figure 9.

Figure 9 Operator log

```
Jun 17 13:27:49> RL5058: Mount request system started
Jun 17 13:27:53> RL5253: Device drive1, added
Jun 17 13:27:53> RL5252: Device drive1, configuration updated
Jun 17 13:27:53> RL5257: Device drive1, request up
Jun 17 13:27:54> RL5253: Device drive2, added
Jun 17 13:27:54> RL5252: Device drive2, configuration updated
Jun 17 13:27:54> RL5257: Device drive2, request up
Jun 17 13:27:56> RL5253: Device drive3, added
Jun 17 13:27:56> RL5252: Device drive3, configuration updated
Jun 17 13:27:56> RL5257: Device drive3, request up
Jun 18 13:43:22> RL5057: Mount request system going down
Jun 18 13:43:29> RL5058: Mount request system started
```

The user log: REEL/logs/u_*uname*

The user log records all user messages, including error messages and messages from the system operator. The date, time, system message number and message text are displayed for each message, in chronological order. To keep a constant, dynamically updated user log on display, issue the following command from a separate terminal session (substitute your `RLLIBDIR` value for *LIBDIR* and your own user ID for *uname*):

`tail -f` *LIBDIR*`/REEL/logs/u_`*uname*

A sample user log is shown in Figure 10.

Figure 10 User log

```
Jul  1 12:37:12> RL5239: Request 1027 obtained resources
Jul  1 12:37:41> RL5198: Device: 380: Allocate Scratch Tape 'test/tape1' (ANSI)
Jul  1 12:37:44> RL5241: Request 1027 Device /trial/stress ready
Jul  1 12:37:44> RL5242: Request 1027 return success
Jul  1 12:38:15> RL5198: Device: 380: Allocate Scratch Tape 'test1/tape2' (ANSI)
Jul  1 12:38:41> RL5198: Device: 380: Allocate Scratch Tape 'test2/tape3' (ANSI)
Jul  1 12:39:44> RL5198: Device: 380: Allocate Scratch Tape 'test4/tape5' (ANSI)
```

The job log: REEL/logs/J_*uname/key*

The job log records the same types of information as the user log, but for a specific job, as identified by a job key. Job logs are stored in the directory `logs/J_uname`, in files identified by specific job keys. For more information on job keys, refer to the section titled *Resource Key* in Chapter 2 of the *ConvexTMR User Guide*.

To keep a constant, dynamically updated job log on display, issue the following command from a separate terminal session (substitute your `RLLIBDIR` value for *LIBDIR* and your own user ID and job key for *uname* and *key*):

`tail -f` *`LIBDIR`*`/REEL/logs/J_`*`uname/key`

The system log: REEL/logs/SYS

The system log is a comprehensive log of all ConvexTMR activity. Messages are identified by message type (`ADM` or `OPR`) and severity. Level 1 messages are low priority, and are recorded in the log only. Level 0 messages are considered critical messages, and immediately display to all logged-in, interested and affected parties as soon as they are generated.

The following command, when issued from a separate terminal session, requests the display of a constant, dynamically updated system log (substitute your `RLLIBDIR` value for *LIBDIR*):

`tail -f` *`LIBDIR`*`/REEL/logs/SYS`

A sample system log is shown in Figure 11.

Figure 11 System log

```
Jun 19 11:01:07> RL5058: Mount request system started
Jun 19 11:01:07> RL5270: (ADM:1) Could not raise priority
Jun 19 11:01:13> RL5253: (OPR:1) Device drive1, added
Jun 19 11:01:13> RL5252: (OPR:1) Device drive1, configuration updated
Jun 19 11:01:13> RL5257: (OPR:1) Device drive1, request up
Jun 19 11:01:14> RL5253: (OPR:1) Device drive2, added
Jun 19 11:01:14> RL5252: (OPR:1) Device drive2, configuration updated
Jun 19 11:01:14> RL5257: (OPR:1) Device drive2, request up
Jun 19 11:01:15> RL5253: (OPR:1) Device drive3, added
Jun 19 11:01:15> RL5252: (OPR:1) Device drive3, configuration updated
Jun 19 11:01:15> RL5257: (OPR:1) Device drive3, request up
Jun 19 11:37:59> RL5264: Secure: Bad Password user=120  host=sam
Jun 21 13:12:48> RL5252: (OPR:1) Device drive1, configuration updated
```

Glossary

A

ACL
Access Control List; a list of user and group IDs and access permissions associated with a ConvexTMR object. The ACL of a ConvexTMR object determines what kind of access specific users and groups have to the object. Viewed via `rllsacl`; controlled via `rlchacl`.

access permissions
A string of alphabetic characters associated with a user or group ID that controls access to a ConvexTMR object. An element of the access control list (ACL).

active state
The state of a volume that contains unexpired and wanted data.

administrator groups
A group of administrators assigned a specified authority level, name, and password. An administrator must be enrolled in a group to perform administrative tasks.

authority levels
A hierarchical division of powers; administrator groups are always assigned an authority level.

B

BLP
Bypass Label Processing; a processing mode that ignores the volume and file tape labels. Tape positioning and navigation occur as under standard label processing, but the labels are ignored. A ConvexTMR privilege.

C

catalog
A database that tracks all of the objects in the ConvexTMR library. Catalog records are maintained on ownership, contents, pools, vaulting, status, and many other details.

command line interface
The interface to the ConvexTMR mount request system that accepts ConvexTMR commands at the command line.

container
The container in which a volume is shipped and/or stored. Often included in the *vol_def*.

D

database key
A system-generated string of characters that uniquely identifies a ConvexTMR file, volume, volumeset, pool, or rotation.

dataset
See tape file.

device
A tape drive. Specific devices are referenced by *dev_name*.

dev_name
A name for a specific tape device. A device can be identified by the device's special file pathname.

display panel
The eight-character LED display panel on IBM 3480/90 devices. ConvexTMR utilizes this panel to inform library operators of pending requests.

domain
A set of tasks and devices defined by the ConvexTMR administrator. ConvexTMR has a global domain that covers all requests. Multiple, custom domains are also supported.

domain server
A server that controls a specific operator domain.

E

EOV
End-of Volume; the physical end of a tape.

external label (*ext_lbl*)
The user-defined, unique string of from 4 to 12 characters that identifies the volume.The external label name is often recorded on a sticker on the outside of the volume.

ESL
Exceeding site resource limits for the number of devices that can be allocated simultaneously to one user or allocated under one resource key; a ConvexTMR privilege.

F

file
See tape file.

file access mode
Accessing tape data on a file basis only. When a tape is accessed in this manner, no other portion of the tape can be accessed.

file section
The segment of a tape file which resides on a single volume. If a tape file spans three tape volumes, then it has three file sections.

fileset
A set of associated, adjacent files residing on a volume or volumeset.

file_spec
A very specific method of file referencing recognized by the ConvexTMR catalog. For an expanded definition, see the `rlflc`(1) manpage.

fingerprint
A string of characters, calculated from label data and recorded in the ConvexTMR catalog, used to electronically identify volumes when they are mounted.

full-screen interface
An easy-to use alternative to the command-line interface, comprised of three windows: the request queue window, the device list window, and the message window.

G

generation
An instance of a file or volumeset. Generations allow multiple instances of a file or volumeset to exist at the same time and to be referenced independently. Newer instances are given higher generation numbers. A file with the name `payroll:G2:V1`, is the second instance, or generation, of the file payroll. The

number proceeded by "G" is the generation number; the number proceeded by "V" is the version number (see version).

H

header labels
Electronic labels at the beginning of volumes (volume header labels) and tape files (file header labels) that contain identification and other data.

hold state
A volume no longer part of a volumeset that is reserved for future use.

I

implicit pools
A site configuration that creates private user pools for all library users. Indicated by the tag `dpool=IMPOOL` in the site constants report.

initialization
The process by which labels are written to ConvexTMR tape volumes; accomplished via `rlinit`.

internal label (*`int_lbl`*)
The string of characters recorded in the VOL1 label of all IBM and ANSI volumes; often referred to as the volume serial number (VSN). In many tape libraries, internal and external labels are identical. Maximum length for an internal label is 6 characters.

K

key
See database key.

L

labels
See header, trailer, user, internal, or external labels.

LIBDIR
The RLLIBDIR value specified in the file /etc/`reelenv`.

low water mark
A number that, if set, is the lowest number of scratch volumes ConvexTMR will allow in a pool without sending warning messages to the pool owner. This number is set via `rlpoolc` or `rlpoole`.

M

mask
An ACL entry that specifies the maximum permissions allowed to all users and groups. The mask entry overrides all specific user and group entries, but does not override the entry for other users.

O

OBO
Initiating and controlling tape sessions "on behalf of" other users; a ConvexTMR privilege.

OSH
Off-line storage hierarchy; the ConvexTMR directory structure. The OSH is not specific to a particular UNIX shell or process; do not confuse the OSH with your UNIX directory structure.

operator groups
A group of operators assigned a specified authority level, name, and password. An operator must be enrolled in a group to perform operator tasks.

P

pathname
A ConvexOS directory or file name. Examples: /usr, /etc/passwd, ./passwd.

permissions
See access permissions.

PDA
Accessing specific physical devices by name; a ConvexTMR privilege.

physical volume
A single volume, (see volume).

pools
A group of volumes. Tape pools partition the tape library into sub-libraries.

R

receipt number
A five-digit, random number proceeded by "R." Receipts are generated by the ConvexTMR software when volumes are submitted to or retrieved from the tape library. Receipt numbers can be used to reference volumes and volumesets.

REEL-librarian objects
Objects that can be referenced or manipulated with the ConvexTMR software. These objects include files, volumes, volumesets, pools, and rotations.

request ID (*reqid*)
A five-digit, sequential number assigned to a `rlaccess` or `rlnext` request. Request IDs appear on the request monitor and can be used to reference specific ConvexTMR requests.

request monitor
The monitor through which the operator views and services ConvexTMR requests, messages, and device status; command-line and full-screen interfaces are available.

request queue
The list of all outstanding operator requests that displays on the request monitor.

resource key (*reskey*)
A user-defined alpha-numeric sequence of up to twelve characters that distinguishes a tape session from other sessions under the same user ID. Tape requests made under the same resource key affect the same tape session. Tape requests made by the same user ID under a different resource keys affect different sessions.

retension
To rewrap tape on its storage spool to avoid loose spooling; a maintenance procedure to safeguard against tape breakage.

rewind
To position to the beginning of a tape.

role groups
See operator or administrator groups.

rotations
A list of vault locations and durations assigned to a volumeset. Rotations are ConvexTMR database objects.

S

scratch state
A volume that is not in current use, and on which any data has expired. Scratch volumes are available for assignment to a volumeset.

session
The period during which a user accesses a tape or tapes on one or more drives using the same resource key.

slot
The unique address of a volume in the storage vault. At some sites tapes are organized by external label rather than slot. Often included in the *vol_def*.

T

tape file
A set of related bits written to tape. Tape files can span multiple volumes within a volumeset.

tape mark
A delimiter used to indicate end of tape files.

trailer labels
Electronic labels at the end of volumes (volume trailer labels) and tape files (file trailer labels) that contain identification and other data.

U

user labels
Electronic labels reserved for user or application data; user labels follow the file header labels.

V

vault
The storage location for a group of volumes; ConvexTMR supports multiple vaults.

version
An instance of a generation of a file or volumeset; versions allow another level of object subscripting beyond generations. Newer versions are assigned higher numbers. A file with the name `payroll:G2:V1`, is the first instance, or version, of the second generation of the file `payroll`. The number proceeded by "V" is the version number; the number proceeded by "G" is the generation number (see generation).

vol_def
A method of volume referencing that may include any or all of the following values: internal label, external label, slot, container. *vol_def* is used to reference uncataloged volumes.

vol_spec
A very specific method of volume referencing recognized by the ConvexTMR catalog. For an expanded definition, see the `rlvsc`(1) manpage.

vs_spec
A very specific method of volumeset referencing recognized by the ConvexTMR catalog. For an expanded definition, see the `rlvsc`(1) manpage.

volume
A single tape.

volume access mode
Accessing an entire volumeset. When a volume is accessed in this manner, any and all files on the volumeset can be accessed.

volumeset
A logical volume consisting of one or more physical volumes.

Index

R

S

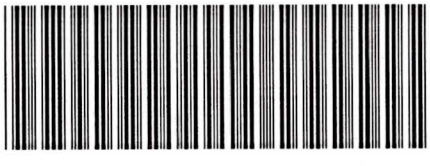
T

U

V

ORDER NUMBER
DSW-480

DOCUMENT NUMBER
710-029630-000



CONVEX PRESS