

StorageTek®

REELlibrarian™

Master Guide

Version 3.5

Part Number: 311217101

First Edition

Proprietary Information Statement

The information in this document is confidential and proprietary to Storage Technology Corporation and may be used only under the terms of the product license or nondisclosure agreement. The information in this document, including any associated software program, may not be disclosed, disseminated, or distributed in any manner without the written consent of Storage Technology Corporation.

Limitations on Warranties and Liability

This document neither extends nor creates warranties of any nature, expressed or implied. Storage Technology Corporation cannot accept any responsibility for your use of the information in this document or for your use of any associated software program. You are responsible for backing up your data. You should be careful to ensure that your use of the information complies with all applicable laws, rules, and regulations of the jurisdictions in which it is used.

Warning: No part or portion of this document may be reproduced in any manner or in any form without the written permission of Storage Technology Corporation.

Restricted Rights

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227–7013 or subparagraphs (c) (1) and (2) of the Commercial Computer Software — Restricted Rights at 48 CFR 52.227–19, as applicable.

Trademarks

StorageTek® is a registered trademark of Storage Technology Corporation.

Other product names mentioned in this manual may be trademarks. They are used for identification purposes only.

First Edition: May 1997

This edition applies to REELlibrarian Release 3.5. Information in this publication is subject to change. Comments concerning the contents of this manual should be directed to:

Storage Technology Corporation
Ann Arbor Software Development Lab
3626 West Liberty Road
Ann Arbor, Michigan 48103-9014
(800) 678-4430

© 1997 Storage Technology Corporation. All rights reserved

Table of Contents

Preface	xiii
REELlibrarian Documentation	xiii
How to Use This Guide.....	xiii
Guide Contents	xiii
UNIX Chapter Conventions	xv
Guide Conventions	xv
Chapter 1. Overview	1
Tape Management Overview	1
How REELlibrarian Works	3
Client-Server Model with Daemons.....	3
The Mount Request System	3
Catalog and Tape Management	3
Getting Started	4
Installation	4
Adding Tapes to the Library	4
Conducting Operations	4
A Test Run	5
Chapter 2. Configuration	15
Overview.....	15
Configuration	15
Administration	16
Role-Based Interfaces.....	16
Concepts	17
The Library	17
Volume Naming and Identification	17

Volume Submission and Retrieval	18
Volumesets	19
Security	20
Reports	20
Pools	20
Media Movement	22
Library Maintenance	22
The Catalog	22
Device Manager/Mount Request System	22
Device Resource Manager	23
Operator-Assisted Mount Requests	23
Idle Devices	24
The Servers	24
Networks	25
Operators and Daily Operations	25
The Environment File: reelenv	25
Vault Management	27
Running the Configuration Program	29
Interaction & Special Keys	29
Starting the rlconfig Program	30
Operators	31
Tape Drives	34
Tape Drive Locking	40
Storage Sites	41
Media Type Aliases	42
Alias List	45
Miscellaneous	45
Drive Configuration	50
Confirming a Drive Entry: rldtest	50
EOT Detection	51
Configuring the trusted_hosts File	52
Configuring Autoloader Tape Devices	53
Standard Site Exit Scripts	54
Modifying Stacker Indexing Times	55
Configuring the Vault Management System	55

VMS Implementation: Existing REEL Libraries	56
VMS Implementation: New Installations	57
An Example Of Setting up Vault Management.	57
Site Information.	61
Configuration File - rlvms_config(4)	62
Configuration Explanation.	62
Form Factor and Media Type Specification	62
Vault Specification	63
Slot Specification	63
VMS Operations at the Example Site	64
Adding Tapes	64
Moving Volumes.	64
Checking for Volumes Awaiting Movement	65
Confirming Volume Movement	65
Deactivating Vault Management	66
Other Configuration Issues	67
System Capacity	67
Terminal Displays	67
Key Commands	69
Administration	69
Using Tracing Facilities.	69
Turning Tracing On	69
Trace File Disposition	71
Accounting Log	71
The Catalog Journal and Backups	72
Controlling Journal Size	73
Server Failure and Recovery	73
Recovering the Catalog: The RLrebuild Command	74
License Control/Adding New Clients	74
REELlibrarian NetMaster.	74
Upgrading for NetClient Support	75
REELlibrarian NetClient	75
Changing/Updating a Hostname	75
REELbackup & REELlibrarian Issues	76
Command Summary	77

Chapter 3. Operations	79
Overview	79
Servicing Mount Requests	79
Tending the Tape Library	80
Controlling the Network Servers	80
The rl Program	80
Operator Paging	81
Controlling the Network	81
Checking Network Status	82
Controlling Server and Client Processes	82
Checking Node Program Status	82
Starting Node Programs	83
Stopping Node Programs	83
Servicing the Request Monitor	83
Full-Screen Operation: rlmon	83
ACT: Action Requests	88
Tape Identity Problems	89
SMNT: Mount a Scratch Tape	89
WMNT: Mount for Write Request	89
RMNT: Mount for Read Request	89
UMNT: Unmount Request	90
DCHK: Density Check Request	90
VADD: Add New Volume	90
Command Line Operation: rlop	91
WMNT: Mount Request	91
Tape Identity Problems	92
Skipping the Mount Request	92
RMNT: Mount for Read Request	92
UMNT: Unmount Request	92
DCHK: Density Check Request	93
VADD: Volume Add Request	93
Working with Autoloader Tape Devices	93
Stacker Messages	93
Problem with site_exit config...	94
Problem with site_exit device...	94

Can't access tape.....	94
Unknown error encountered...	94
Handling the Last Tape in the Magazine.....	95
Tape Management Duties	95
The Tape Catalog.....	95
Handling the Tape Life Cycle	96
Entering Tapes into the Library.....	99
Full-Screen Operation	99
Command Line Operation	102
Exiting Tapes from the Library.....	103
Full-Screen Operation	103
Command Line Operation	103
Editing a Tape Catalog Entry	104
Full-Screen Operation	104
Command Line Operation	104
Tape Movement, Cleaning, & Removal	105
Moving Tapes	106
Cleaning Tapes	106
Disposing of Tapes.....	106
Moving Tapes	106
Cleaning Tapes	106
Disposing of Tapes.....	106
Reports.....	107
Command Summary.....	107
Chapter 4. Using REELlibrarian	109
Overview.....	109
The Library.....	109
Volumesets	110
Volumeset Name	110
The Catalog	111
Tape Sessions	111
Pools.....	111
User Interfaces	112
The Full-screen Interface.....	113

Navigating the Full-screen Interface	114
Submitting a Volumeset	115
Editing a Volumeset	121
Retrieving a Volumeset	123
Creating a Volumeset from Pool Volumes	124
Deleting a Volumeset	125
Submitting a Scratch Volume	126
Manually Identifying (Fingerprinting) a Volume	128
Retrieving a Scratch Volume	129
Editing the Volume Catalog Entry	130
Creating/Editing/Deleting Pools	131
Creating a Pool	131
Editing a Pool	134
Deleting a Pool	135
Creating/Editing/Deleting Rotations	135
Creating a Rotation	135
Editing a Rotation	138
Deleting a Rotation	138
A Volumeset Session	139
Storing Files on a Volumeset	139
Reading Files from a Volumeset	140
Table of Contents Report	141
Transient Volumesets	141
Using the Catalog	141
Volumeset Format Issues	141
TAR Tape Handling	141
TAR Tape Handling Qualifications	142
Creating a TAR Volumeset	142
Reading TAR Tapes	142
Writing TAR Tapes	143
CPIO Tape Handling	143
RAW Tape Handling	143
ANSI and IBM Tape Handling	143
Volume Labels	143
Creating an IBM or ANSI Labeled Tape	144

File Labels	144
IBM File Formats	145
ANSI File Formats	145
Record Conversion	146
Text and Data File Formats	147
ANSI Text Files	147
IBM Text Files	148
ANSI Data Files	148
IBM Data Files	148
IBMU (Unlabeled) Tape Handling	148
Record Conversion	149
Volumeset Defaults	149
Default Volumeset Priority	149
Volumeset Default Formats	150
Creating and Using Default Definition Volumesets	150
File Name Templates	151
Direct Device and Volume Access	153
Using Multiple Tape Drives	154
Deadlock Situations	155
Freeing Reserved Drives	155
Command Summary	156
Chapter 5. Using Reports	159
Overview	159
User Reports	159
Full-Screen User Reports	159
Volumeset List	160
Volumeset Attributes	162
Volume List	165
Volume Attributes	167
Pool List	167
Pool Attributes	168
Pool Volumes	170
File List	171
File Attributes	172

Command-Line User Reports	175
Rotation Information	175
Rotation List	176
Storage Location	176
Media Type	177
Volumeset File List	179
Volumeset Volume List	180
Operator Reports	181
Full-Screen Operator Reports	182
Volume Maintenance	182
Volume Inventory	185
Command-Line Operator Reports	186
Device Information	186
Tapecap Report	188
Vault Content	190
Chapter 6. Programming Interface	193
Overview	193
Tape Programming Requirements	193
Compiling With the Library: librltape.a	193
Tape Formats	193
Data Structures	194
Sample Programs	194
Example 1	194
Example 2	196
Example 3	199
User Commands (UNIX Section 1)	203
C Library Functions (UNIX Section 3)	291
File Formats (UNIX Section 4)	351
Maintenance Commands (UNIX Section 8)	379
System Messages	451
Log Messages	451

Message Classes	452
Message List	452
User Messages	472
Message List	472
Index	501

Preface

REELlibrarian Documentation

The documentation for REELlibrarian includes:

- *REELlibrarian Installation Guide* — step-by-step instructions for installing the REELbackup software.
- *REELlibrarian Master Guide* — presents REELlibrarian processes and procedures in detail. Use these to tailor the software for your site's specific tape library requirements.

How to Use This Guide

The *REELlibrarian Master Guide* is your source of information on using tapes with the REELlibrarian software.

Guide Contents

- **1 Overview** — Read this chapter to introduce yourself to REELlibrarian. This chapter presents REELlibrarian and the fundamentals of tape management in non-technical language. Tape library organization and operations, data access methods, tape life cycles, and tape objects are presented and defined. This chapter also outlines steps for getting started and walks you through a tutorial.
- **2 Configuration** — Library administrators should refer to this chapter for instructions on configuring and controlling REELlibrarian.

- **3 Operations** —Library operators should turn to this chapter for instructions on conducting routine library tasks such as tape mounts and tape maintenance.
- **4 Using REELlibrarian** — End users should use this chapter to learn how to conduct REELlibrarian tape sessions.
- **5 Using Reports** — Refer to this chapter for information on using the REELlibrarian reports.
- **6 Programming Interface** —Turn to this chapter for information on using the REELlibrarian C function library to create customized applications.
- **User Commands (UNIX Section 1)** — This appendix contains UNIX-style reference pages for each of the REELlibrarian (UNIX Chapter 1) user commands. These pages are also provided in electronic form with the software, and can be installed in the standard UNIX on-line manual system. Please see *UNIX Chapter Conventions* on page xv for more information.
- **C Library Functions (UNIX Section 3)** — This appendix contains UNIX-style reference pages for all of the (UNIX Chapter 3) REELlibrarian C Library functions. These pages are also provided in electronic form with the software, and can be installed in the standard UNIX on-line manual system. Please see *UNIX Chapter Conventions* on page xv for more information.
- **File Formats (UNIX Section 4)** — This appendix contains the UNIX-style reference pages for all of the (UNIX Chapter 4) REELlibrarian file formats. These pages are also provided in electronic form with the software, and can be installed in the standard UNIX on-line manual system. Please see *UNIX Chapter Conventions* on page xv for more information.
- **Maintenance Commands (UNIX Section 8)** — This appendix contains UNIX-style reference pages for each of the REELlibrarian (UNIX Chapter 8) operator and administrator commands. These pages are also provided in electronic form with the software, and can be installed in the

standard UNIX on-line manual system. Please see *UNIX Chapter Conventions* on page xv for more information.

- ***System Messages*** — This appendix lists and describes all of the REELlibrarian log and user messages.
- ***Index*** — A detailed index of the terms, concepts, and functionality presented in this document.

UNIX Chapter Conventions

Historically, the UNIX manual has been divided into standardized chapters (called “sections”) by topic. These sections and their associated topics are:

- Section 1: User Commands and Application Programs
- Section 2: System Calls
- Section 3: Libraries and Subroutines
- Section 4: File Formats
- Section 5: Miscellaneous
- Section 6: Games
- Section 7: Special Files
- Section 8: System Management Commands

REELlibrarian commands follow these conventions. For convenience, REELlibrarian commands are followed by the corresponding UNIX chapter in parenthesis. For example, `rlvsubmit(1)` and `rlaccept(8)`.

Guide Conventions

This guide employs several notational conventions to make it easier for you to identify different types of information and follow instructions. These conventions are described below.

Monospace Bold	This typeface represents literal input by you, the user.
Monospace	Text as it appears on screen is represented with this typeface; all specific user commands and objects such as filenames and environment variables also appear in this typeface.
<i>Monospace Italic</i>	Names of variables to which values must be assigned (such as <i>password</i>) appear in this typeface. In actual usage, replace the variable text with an appropriate value.
[[Key]]	Keyboard keys to press appear in double square brackets. Keys to press simultaneously (such as control key combinations) are separated by a dash.
[]	Square brackets enclose command options and arguments that are optional.
{ }	Braces enclose a series of arguments from which you may select one.
	The vertical bar separates optional arguments from which you may choose one.
\	A backslash (\) indicates a continuation of a command-line entry that has been wrapped to a second line in hardcopy.
, ...	A comma and a series of three periods indicates comma-separated lists.
...	A series of three periods indicates a space-separated list.
<input type="checkbox"/>	A box indicates a checklist.
	This symbol marks useful hints.



This flag marks exceptions and notes.



Caution symbol



Warning symbol

Acknowledgments

The following terms used in this document are trademarks of the named organizations:

- UNIX — AT&T Bell Laboratories
- IBM—International Business Machines Corporation
- REELbackup—Storage Technology Corporation
- REELlibrarian—Storage Technology Corporation

Chapter 1. Overview

Tape Management Overview

If you are new to tape management systems or you would like to understand how REELlibrarian works, read this overview section and the overviews in Chapters 2, 3, and 4.

Computer users store and retrieve disk files (online files) on magnetic tape (offline media) for two main reasons:

- to make permanent, archival copies of important files
- to free occupied and needed disk space by moving seldom-used files to tape files

The UNIX utilities, `cpio` and `tar`, write disk files to tape, but they lack the security controls, contents catalogs, tape drive arbitration, and other facilities necessary for reliable and useful tape activities. REELlibrarian provides the services and facilities which make tape usage convenient and secure.

REELlibrarian is designed to address the functional needs of three classes of users:

- system administrators
- operators
- end users

REELlibrarian provides both full-screen and command-line interfaces to support the activities of each of these groups.

REELlibrarian places all tapes in the library under the control of the operator. The operator is the person who mounts and unmounts tapes on the system drives. When users request access to tapes, REELlibrarian processes and translates each request into an instruction to the operator to mount or unmount the specified tape on the assigned drive. User requests are ordered in a queue and are serviced as drives become available. By controlling physical access to tapes and drives, REELlibrarian maintains security over tape drives and user data.

Users submit their tapes to the library and then use them via tape access commands. If the library is set up with a public tape pool, users with the proper permissions can also utilize tapes from the public pool.

Users store data on offline tapes by creating and accessing volumesets. A volumeset is a logical tape which can consist of one or more physical tapes. A volumeset is organized and accessed as a single tape. When a user writes data files to the volumeset, transitions from the end of one tape volume to the beginning of the next tape volume are handled transparently to the user. If a large file fills the first volume in the volumeset before REELlibrarian has finished writing the file, REELlibrarian adds another volume to the volumeset automatically. REELlibrarian then instructs the operator to unmount the first volume and mount the next volume before it resumes writing the file. Volume transitions are handled similarly during requests to read data from offline tapes.

Tape drives are assigned to users automatically via their requests for access to volumesets. Each access request to a volumeset causes REELlibrarian to assign an appropriate tape drive for the user's session. When there are more requests than there are drives, then the requests are queued and satisfied as drives become available.

The online catalog stores information about library volumes, volumesets, files, pools, and rotations. Several user commands are available to perform catalog activities.

REELlibrarian is network software. It provides a universal tape management service to multiple systems on a network. All tape drives are centrally controlled and are accessible for tape

operations. A user can access tapes via a tape drive on another network node just as if it were on a local node.

How REELibrarian Works

REELibrarian consists of two software components:

- NetMaster - the server software for the master node. It includes the NetClient software for the master node.
- NetClient - the client software for all computer nodes.

Client-Server Model with Daemons

REELibrarian software is based on the client-server model of network software. The main server provides services on a single node (the master node) to client processes located anywhere on the network. Most REELibrarian commands can be run anywhere on the network. These commands act as clients since they contact the server to perform their functions.

REELibrarian also uses daemons. A daemon is a program that executes in the background, and waits to be triggered into active processing. The REELibrarian daemon sits on all network nodes and awaits instructions from the server. The daemon is used to start tape reading and writing activities.

The Mount Request System

The Mount Request System (MRS) maintains a queue of tape mount requests and presents each in turn to the operator for action via the request monitor. The operator mounts the specified tape and confirms it to the request monitor. The MRS then notifies the user command which in turn reports to the user.

Catalog and Tape Management

The server maintains a catalog which tracks all tapes in the REELibrarian library and all data stored on the tapes. All access to tapes and the data on the tapes is handled via the server and the catalog.

Getting Started

Installation

Installation instructions are provided in the *REELlibrarian Installation Guide* that is shipped with the distribution media.



Note: After you install the software, and before you add any tapes to the library or perform any other configuration tasks, you may want to try the practice session outlined in *A Test Run* on page 5.

If you are installing REELlibrarian and need to change the hostname, please see *Changing/Updating a Hostname* on page 75.

Adding Tapes to the Library

Before you can use REELlibrarian, you must enter tapes into the library.

Each tape belongs to a pool. A pool is nothing more than a set of tapes with the same owner. Each user has at least one pool named `private` that is created by default. When a user submits tapes to REELlibrarian, they enter the user's pool `private` by default.

Users can create additional tape pools. Pools have security permission masks, which restrict who can use the tapes in the pool. This makes it possible for the system administrator to create pools which are public and can be used by everyone.

To get started with REELlibrarian, the REELlibrarian administrator must create a public pool and place tapes in it, or each user must submit tapes to the user's private pool. Pool creation is discussed in *Creating a Pool* on page 131. The submission process starts with the `rlpsubmit` program; scratch volume submission is discussed in *Submitting a Scratch Volume* on page 126. All submitted volumes must be accepted by the library operator via the `rlaccept` program; volume acceptance is discussed in *Entering Tapes into the Library* on page 99.

Conducting Operations

The root ID is configured as an approved operator. To perform the operator functions, the operator must be logged in as root.

To conduct library operations, the REELlibrarian server and client software must be running on the appropriate nodes. For more

information, refer to *Controlling Server and Client Processes* on page 82.

Mount and unmount requests are communicated to the library operator via the request monitor; the request monitor is accessed via the `rlmon` program. Refer to *Servicing the Request Monitor* on page 83 for a full description on how to service the mount requests that appear.

A Test Run

If you would like to test the software before starting general operations, a practice session is offered here. This test run demonstrates a typical use of REELlibrarian.



Note: REELlibrarian must be installed before you can use the software. Install REELlibrarian according to the instructions provided in the *REELlibrarian Installation Guide* that is shipped with the distribution media.

The test run requires root privileges; you *must* be logged in as root to perform some of the following tasks.

To conduct this session, you will act as both an operator and an end-user. Since these roles require simultaneous interaction with the system, two separate logins are needed either via two terminals or two windows on a windowing system.

Log in as root on one window and as any non-root user ID on the other. Throughout the test run, the text refers to these as the root window and the user window.

1. From the **root** window, start the REELlibrarian servers.
Enter:

```
reel start
```

Messages similar to the following will appear on screen:

```
Starting rllog...
```

```
Becoming background daemon
```

```
rllog startup Complete
Starting RLnet...

Becoming background daemon

RLnet startup Complete
Starting RL...

REELlibrarian version 3.3 (serial#
000000)

Network Connection Capacity:

    Class E - 10

        - start after normal shutdown...

Starting Mount request services
    - Initializing Device drive1

Starting Database transaction log

Becoming background daemon

RL startup Complete

3 Server(s) started.
```

2. From the **user** window, give REELlibrarian a tape to work with. Enter:

```
rlpsubmit
```

A message similar to the following will appear on screen:

```
Scratch Volume Submitted, Volume ID:
lath-369 (onsite/)
```

This command informs the system that the user is placing an empty tape under its control. Note that the name `lath-369` is assigned as a volume ID (VID) by the system in the example. REELlibrarian will assign a different name during your test run. The rest of this Test Run uses the `lath-369` as the VID.

3. From the **root** window, accept the tape into the library and assign it a rack number. Enter:

```
rlaccept rack=100 lath-369
```



Note: Be sure to substitute the volume ID returned in step 2 for lath-369.

A message similar to the following will appear on screen:

```
Volume Accepted
Volume Location: onsite/100
Volume id: lath-369
```

4. From the **root** window, initialize or “fingerprint” the tape. Enter:

```
rlid adn=drive1 lath-369
```



Note: Be sure to substitute your drive name for drive1 and the volume ID returned in step 2 for lath-369.

A message similar to the following will appear on screen:

```
Reserving Device...
Reservation Complete
Volume 'lath-369' - mount with type=CART hit return:
```

Mount the specified tape in the drive and press Return. A message similar to the following will appear on screen:

```
Volume Identification Complete
```

5. From the **user** window, confirm the tape is in the library by running the volume list report:

```
rlr vlist
```

Output similar to the following will appear on screen:

Volumes Report: Mon Sep 19 11:50:15 1994					
VID	STAT	Type	Length	Location	Pool
---	---	---	---	---	---
lath-369	scr	CART	2400	onsite	>private
Command: rlr user=lfw full=yes vlist					

6. From the **user** window, create a volumeset. Enter:

```
rlvcreate ftrack=y format=ANSI\  
ftemp=@F9@ testvs
```

A message similar to the following will appear on screen:
Allocated Volume: lath-369

Note that the volumeset has been given the only tape available.

7. From the **user** window, confirm that the volumeset is in the library by running the Volumeset List report. Enter:

```
rlr vslist
```

Output similar to the following will appear on screen:

```
Volumeset List Report: Mon Sep 19 12:06:10 1994  
Vname      STAT   Vexpire      Expires      Vcomment  
-----  
testvs     init   S            -99999 D  
Command: rlr user=lfw full=yes vslist
```

8. From the **user** window, access the tape for a writing session. Enter:

```
rlvaccess write=y testvs
```

A message similar to the following will appear on screen:

```
Device Reserved
```

9. From the **user** window, request that a file be written to the volumeset. Enter:

```
rlvwrite fid=file1 rformat=vbs:4096:512\  
conv=text if=/etc/passwd
```

A message similar to the following will appear on screen:

```
Requesting tape lath-369...Awaiting  
Mount...
```

The `rformat=` keyword specifies the record and blocking formats. The example shows an appropriate setting for a text file like `/etc/passwd`. For other types of files, other values are appropriate. The `conv=` keyword indicates that the file

data is text and should be appropriately converted. For a complete description of `rlvwrite`, refer to Appendix A, “Command Man Pages”.

10. All mount requests appear to the library operator via the Request Monitor; the Request Monitor is shown in Figure 1-1. From the **root** window, start the Request Monitor.
Enter:

```
r1mon
```

A scratch mount request (ACT: SMNT) for the requested drive (ADN: drive1) and volume (Location/Rack: onsite/100) will appear in the bottom half of the Request Monitor, as shown in Figure 1-1.

```
-----| REQUEST MONITOR |-----
Device  Type  Stat VID          UID          Key          Pseudo
drive1  CART  off  -----  1fw          1fw          default

-----
MID  ACT  ADN  Type  Location/Rack  Pool
>020* SMNT drive1  CART  onsite/100(2400)  1fw/private

-----
Action: *MNT-DONE  MNT-SKIP  HELP  EXIT

-----
REELlibrarian 3.3 (C)opyright 1993-4 by Storage Technology Corporation
```

Figure 1-1 The Request Monitor

11. Mount the specified tape on the specified drive and press Return from the **root** window.
12. A pop-up window will appear in the **root** window prompting for the volume ID of the mounted tape, as shown in Figure 1-2. Enter the volume ID of the volume submitted in step 2 and press Return. In this example, the volume ID is 1ath-369.

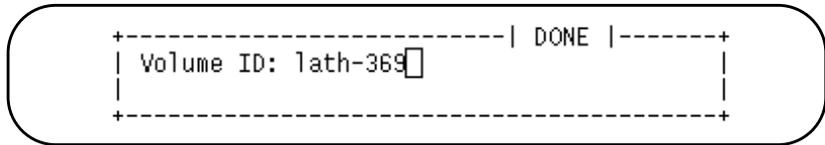


Figure 1-2 *The Volume ID Prompt*

13. From the **user** window, write two more files to the tape. Enter:

```
rlvwrite fid=file2 rformat=vbs:4096:512\  
conv=text if=/etc/passwd
```

```
rlvwrite fid=file3 rformat=vbs:4096:512  
conv=text \ if=/etc/passwd
```



Note: REELibrarian does not automatically issue unmount requests. An accessed tape will remain online in the premount condition. If it is accessed while premounted, no mount is required and the tape is ready to go. If the premount period expires, then an unmount request is generated.

14. From the **user** window, release the device and tape. Enter:

```
rlvrelease
```

A message similar to the following will appear on screen:

```
Device drive1 Freed
```

15. An unmount request (UMNT) will appear on the **root** window; press Return to satisfy the request.
16. From the **user** window, request a report of the files on the volumeset. Enter:

```
rlr vol=testvs vsflist
```

Output similar to the following will appear on screen:

```

Volumeset File List: Mon Sep 19 16:11:52 1994

Fid      SC Blocks Fexpire Expires Fcomment
---      --  -----  -----  -----
file1    1  8          -----  -----
file2    1  8          -----  -----
file3    1  8          -----  -----
End of Tape
Command: rlr volset=testvs full=yes vsflist

```

- The next few steps demonstrate how to read files from a volumeset. From the **user** window, access the tape for a reading session. Enter:

```
rlvaccess testvs
```

Output similar to the following will appear on screen:

```
Device Reserved
```

- From the **user** window; read the second file on the tape to your current working directory. Enter:

```
rlvread fid=file2 of=_pass
```

A message similar to the following will appear on screen:

```
Requesting tape lath-369...Awaiting
Mount...
```

- A read mount request (RMNT) should appear on the request monitor that is displayed in the **root** window. Mount the specified tape on the specified drive and press Return from the **root** window.
- A pop-up window will appear in the **root** window prompting for the volume ID of the mounted tape. Enter the volume ID of the mounted tape and press Return.
- To check the contents of the file that you read from tape, use the UNIX `diff` command; from the **user** window. Enter:

```
diff _pass /etc/passwd
```

No output to this command means that the file specified in the first argument does not differ from the file specified in the second argument; in this example, the file read from the tape (`_pass`) is identical to the file written to the tape (`/etc/passwd`).

22. From the **user** window, release the device and tape. Enter:

```
rlvrelease
```

A message similar to the following will appear on screen:

```
Device drive1 Freed
```

23. An unmount request (UMNT) will appear on the **root** window; press Return to satisfy the request.

24. From the **user** window, delete the volumeset. Enter:

```
rlvscratch testvs
```

This command disbands the volumeset.

25. From the **user** window, request a Volumes report and a Volumeset List report. This demonstrates that the volume you submitted to the library is still there, but that the volumeset it was drafted into no longer exists. Enter:

```
rlr vlist
```

Output similar to the following will appear on screen:

```
Volumes Report: Mon Sep 19 16:56:09 1994
VID          STAT Type      Length  Location      Pool
---          -
lath-369     scr  CART      2400    onsite        >private
Command: rlr user=lfw full=yes vlist
```

Enter:

```
rlr vslist
```

Output similar to the following will appear on screen:

```

Volumeset List Report: Mon Sep 19 16:56:23 1994
Vname          STAT   Vexpire      Expires      Vcomment
-----
No volumes selected.

```

26. From the **user** window, initiate the retrieval of your scratch tape from the library. Enter:

```
rlretrieve lath-369
```



Hint: Be sure to substitute the volume ID returned in step 2 for lath-369.

A message similar to the following will appear on screen:

```

Retrieved Scratch Tapes:
      VID          Receipt
      ---          -
      lath-369     R017846

```

During normal library operations, the user would present the receipt number to the library operator in order to repossess the tape.

27. From the **root** window, complete the retrieval process by issuing the `rlreturn` command with the receipt number returned in the last step (in this example, it's R017846). Enter:

```
rlreturn R017846
```

A message similar to the following will appear on screen:

```
Location - onsite:100
```

The output of this command displays the location of the tape to return. During normal library operations, the operator would retrieve the tape from the vault and return it to the user.

28. From the **root** window, re-issue the `rlreturn` command, this time with the volume ID of the returned tape (lath-

369 in this example). Enter:

```
rlreturn vid=lath-369 R017846
```

A message similar to the following will appear on screen:

```
Database Record Deleted
```

```
Return volume to user...
```

This step clears the database of the volume record. The library has now been emptied of the one tape submitted during this session, and the system is just as it was after installation.

Chapter 2. Configuration

Overview

This chapter is for the REELlibrarian system administrator. It describes how the system is configured and administered.

This section gives a brief overview of the capabilities provided by REELlibrarian. Readers familiar with other tape management systems will recognize the features described here. Readers learning about tape management for the first time will find this section a useful introduction to the scope of services provided by the REELlibrarian system.

Configuration REELlibrarian comes configured to run out-of-the-box. However, it is easy to change the configuration to fit your site's needs. REELlibrarian configurable parameters are listed below.

- Tape Drives - the identity and capabilities of the tape drives on the computers.
- Tape Media Types - the identity and attributes of the tape media supported by the configured drives.
- Storage Sites - the sites (sometimes referred to as vaults) to and from which tapes can be moved.
- Operators - the list of system IDs which may perform REELlibrarian operator tasks.
- Miscellaneous - other configurable options.

Administration Once REELibrarian is installed and operating, it does not require the regular attention of the administrator. It does however provide several reports which might be useful for accounting purposes: number of mounts per user, tapes in storage, etc.



Caution: If new systems are added to the computer network, REELibrarian can be extended to provide services to the new nodes. The license control facility allows new nodes to be added = immediately to the REELibrarian network without delivery of new software.

Role-Based Interfaces REELibrarian is designed for both single user systems and large multi-user networks. At large sites, library activities may be divided among three different groups: administrators, operators, and the system users. At small sites, a single person may serve all three roles.

REELibrarian roles and activities are listed in the table below.

Table 2-1 REELibrarian Role Groups

Role	Activities
Administrator	<ul style="list-style-type: none">• Installs and configures REELibrarian• Generates reports detailing REELibrarian activities
Operator	<ul style="list-style-type: none">• Services mount and unmount requests• Generates reports detailing REELibrarian activities• Performs tape maintenance tasks
User	<ul style="list-style-type: none">• Writes data to and reads data from library volumesets• Manages and organizes library objects via the catalog

The root user is the only qualified REELibrarian administrator. Operators must be identified to REELibrarian for security

reasons. Thus, program access is limited to only approved operators.

Concepts

REELlibrarian consists of two independent parts, an on-line catalog and a Mount Request System (MRS). The catalog tracks what volumes exist, who owns them, and what data they contain. The MRS arbitrates competing user tape drive requests and makes it possible for a user sitting at a terminal to ask an operator to mount and unmount volumes from the library.

REELlibrarian provides both a command-line and full-screen interface to all its facilities. The following discussion uses command examples for illustrative purposes. The full-screen interface is available via the program `r1`.

The Library

The library is the total collection of tapes and other media submitted to the control of REELlibrarian. Most often, the library is a series of racks of volumes in a restricted area. Generally, only qualified operators have access to the library area and its constituent volumes. The catalog tracks each volume in the library.

Volume Naming and Identification

REELlibrarian uses three identifiers for each volume in the library. These identifiers are shown in Table 2-2, *Volume Identifiers*.

Table 2-2 *Volume Identifiers*

Identifier	Description
volume ID	A twelve-character name by which users identify tape volumes. Volume IDs are unique in the library. REELlibrarian automatically generates a tape's volume ID unless the user supplies it.

Table 2-2 Volume Identifiers (Continued)

Identifier	Description
rack number	A twelve-character name by which operators identify the physical location (or slot) of tape volumes. Rack numbers are unique in the library. The operator must supply the rack number when the tape is accepted (<code>rlaccept</code>) into the library.
fingerprint	<p>The fingerprint is a coded identity derived from the data on the volume. REELlibrarian can initially establish a volume's fingerprint when the tape is identified with the <code>rlid</code> program or do so when the volume is first mounted via REELlibrarian. The fingerprint is always kept current by the system. Whenever a volume is mounted, REELlibrarian takes its fingerprint for comparison with the fingerprint in the catalog and corrects any mistaken mounts by the operator.</p> <p>Note: We do not recommend having volumes with identical fingerprints, otherwise data may be overwritten.</p>



Volume Submission and Retrieval

Submission is the process of placing volumes under REELlibrarian control. Volumes are submitted to REELlibrarian as a two-step process:

- The volume owner runs a command (`rlpsubmit`) which creates a catalog entry for the volume and then presents the volume to an operator.
- The operator runs another command (`rlaccept`) and then places the volume in the library.

Users are free to retrieve volumes from the library at any time. Users initiate volume retrieval with the command `rlretrieve`. `rlretrieve` confirms the user has permission to retrieve the volume and instructs the user to approach the operator. The operator runs the `rlreturn` command which tells the operator which rack number to retrieve. The operator retrieves the volume and presents it to the user.

Volumesets The common unit of storage in REELlibrarian is the **volumeset**. A volumeset consists of one or more physical volumes organized and accessed as one logically continuous tape for data storage that contains one or more data files. Volumes can be used to construct volumesets. Volumesets can also be submitted to the library.

A volumeset is created by specifying the individual volumes which constitute it or by naming the **pool** (described in *Pools* on page 20) from which its members should be drawn. Commonly, a user has a set of volumes in the user's private pool and any volumeset activity draws on these volumes for storage. A volume is added to the volumeset when the current volumes are filled and additional data is being written to the volumeset. Once a volume is made part of a volumeset, it is reserved for that volumeset until the volumeset expires.

REELlibrarian supports the following standard tape labeling formats for volumesets:

- ANSI (default)
- IBM (use when sharing tapes with MVS systems)
- ANSI2 (ANSI without HDR2 labels)
- IBM2 (IBM without HDR2 labels)
- TAR (a generic format usable by most utilities)
- CPIO (a generic format usable by most utilities)
- RAW (for storing raw partitions)

These formats specify how labels will be written to the volumeset. REELlibrarian automatically handles all issues regarding the formats, so users typically can use the default format (ANSI). The other formats are available for specific circumstances (such as writing tapes that must be shared with an MVS system and storing raw partitions).

The commands `rlvsubmit` and `rlvretrieve` operate on volumesets as `rlpsubmit` and `rlpretrieve` perform for individual volumes.

Security An important aspect of a media management system is controlling access to volumes in the library. Files stored on volumesets should be as secure as files stored on disk. REELibrarian provides a UNIX-like permission mask (the `vmode` keyword of the `rlvsubmit` and `rlvedit` commands) that allows separate read/write/execute permission for owner, group and others. To have execution permission means that one can view the catalog entry. However, only the owner can modify the volume's catalog entry.

In addition to the UNIX permission mask, a volume owner can assign a password. When a volume is password protected, REELibrarian prompts for the password before allowing access to it.

Reports Both users and operators can generate a variety of reports summarizing the contents of the library. Users can generate a summary of all volumes they own as well as a detailed report on any particular volume. Operators can generate inventory reports on all volumes in the library, an individual volume, pools, and volumes requiring special attention (cleaning, removal, and movement from one location to another). REELibrarian reports are described in Chapter 5, *Using Reports*.

Pools A pool is a collection of volumes. A volume can belong to only one pool. Each volume in a pool is in one of two states: scratch or active. A scratch volume is not in current use and does not contain any data. For example, when a user submits a blank tape to the library it enters the catalog as a scratch volume in the user's private pool. An active volume, in contrast, already has data stored on it.

Each user has a pool named `private` which, by default, contains all of the user volumes not explicitly assigned to another pool. The `private` pool is not created until it is explicitly referenced by name through any of the REELibrarian commands (e.g., `rlr pool=private pinfo`, `rlpcreate`, `rlpdelete`, etc.) Thus,

the `private` pool will be created automatically for the user when it is first referenced.



Note: Use the `rlr pool=private pinfo` command to explicitly create a `private` pool.

A user can create pools as needed and move volumes from one pool to another if they own that pool. However, a user cannot submit or remove volumes from a pool unless he or she owns it.



Note: The full specification of pool is `userid/pool`. If `userid` is omitted, the current effective user ID is assumed. Only operators may create pools for IDs other than their own.

The user can also specify what other users may access particular volumes within a pool. For example, to create a public pool a user would enter:

```
rlpcreate uacc=ANY gacc=ANY rllib/oz  
  
rlpsubmit type=XBYTE length=2000  
pool=rllib/oz
```

The user would then enter the following to access the public pool:

```
rlvcreate pool=rllib/public type=XBYTE \  
length=2000 toto
```

To access a public pool with no current members you must have the automatic volume addition flag set to `yes`. To do this enter:

```
rlpcreate uacc=ANY gacc=ANY padd=yes  
rllib/oz
```

When a user creates a volumeset, the user defines the pool(s) from which REELlibrarian should draft volumes into service for the volumeset. As needed, REELlibrarian acquires a volume from the designated pool and makes it part of the volumeset. The volume goes from scratch to active status while it is a member of the volumeset. When the volumeset is deleted, the volume returns to scratch status and is available for further use.

Media Movement

For security reasons it is often wise to keep copies of important volumes at locations other than the central library. REELlibrarian can track volume storage at any number of different locations. Users can request volumeset movement from one location to another with the command `rlvmove`. Volumes are actually moved by the operator who periodically generates a list of all volumes waiting for movement and transports them to their new location. The administrator controls what sites are available and whether to accept mount requests for volumes stored at each site.

Library Maintenance

REELlibrarian tracks the number of times a volume is accessed to facilitate periodic cleaning and eventual removal. At any time, the operator can generate a list of volumes whose usage count exceeds the cleaning or removal interval. When a volume is cleaned, the operator confirms cleaning and the cleaning count is reset. When a volume is removed, the catalog entry for the volume is deleted. Volumes are only scheduled for cleaning and removal when they are in the scratch state.

The Catalog

The catalog is an on-line database which contains the REELlibrarian records. Its contents include:

- an entry for each volume in the library
- an entry for each file on a volumeset
- an entry for each pool

REELlibrarian provides several reports that extract information from the catalog.

**Device
Manager/Mount
Request System**

The Device Resource Manager (DRM) provides a device reservation facility. The Mount Request System (MRS) coordinates operator assisted volume mounts. Combined, these two components provide users with a method for interacting with on-line, mounted volumes on an ad-hoc basis.

A user needs to only specify the volume or volumeset to be mounted. REELlibrarian automatically determines the type of drive needed, reserves it, instructs the operator to mount the

volume, and then electronically verifies that the correct volume was indeed mounted.

Device Resource Manager

The Device Resource Manager maintains a prioritized queue of users waiting to reserve tape drives. When a drive becomes available, the DRM assigns the device to the next request in line. To avoid deadlocks, the DRM requires users who need simultaneous access to more than one drive to reserve all necessary drives with a single reservation request.

The command `rlreserve` reserves drives. An example of an `rlreserve` command is shown below. In the example, two tape drives are reserved, one capable of supporting 1600 bpi and one capable of supporting 6250 bpi.

```
rlreserve type=1600,6250 as D1,D2
```

Once you have completed your tape activity, you should free devices with the command `rlfree`. The example below frees the drives reserved by the `rlreserve` example shown earlier.

```
rlfree D1
```

```
rlfree D2
```

Operator-Assisted Mount Requests

Users access volumesets via the `rlvaccess` command which automatically reserves an appropriate drive for the specified volumeset (if one is not already reserved). An example that requests the volumeset `testset` is shown below.

```
rlvaccess testset
```

After the volumeset is accessed, the user can read and write files with the `rlvread` and `rlvwrite` commands. An example of `rlvread` is shown below. In the example, the first three files off the volumeset accessed in the previous example are read into files named `tf1`, `tf2`, and `tf3`.

```
rlvread > tf1
```

```
rlvread > tf2
```

```
rlvread > tf3
```

The `rlvread` example below demonstrates how the command can be used to write only the third file on the volumeset to a file.

```
rlvread fseq=3 > tf3
```

Volumeset access is terminated with the `rlvrelease` command. The example shown below releases the volumeset that was accessed in the previous examples.

```
rlvrelease
```

`rlvaccess` and `rlvread` can handle IBM, ANSI, TAR and CPIO volumesets. A RAW tape type is available for tapes that do not fit one of these classifications.

REELibrarian handles multi-volume volumesets transparently to the user. The library operator is instructed to mount and unmount the volumes in the volumeset as needed to satisfy read and write requests that span volumes.

Idle Devices

Occasionally users access volumesets and forget about them. As a result, devices are needlessly tied up. REELibrarian has a facility to reclaim devices that accumulate excessive idle time.

In the event a user logs out without freeing a device, REELibrarian reclaims the device after a *timeout* and makes it available to other users. The default timeout is twenty minutes.

The Servers

REELibrarian employs server programs which run in the background on each computer system. These server programs provide the coordination for mount requests, tape drive access, and access to the on-line catalog.

The server node in the network runs the RL server program. It provides the catalog service and the Device Manager/Mount Request System. All nodes, including the server node, must run the RLnet server program.

How to start and stop the server programs is described in Chapter 4.

Networks

REELibrarian supports networks with distributed users and tape drives. One node on the network is designated the server and runs the REELibrarian NetMaster software. All other nodes are clients and must be equipped with the REELibrarian NetClient software.



Note: User IDs (UID) and Group IDs (GID) must be consistent on all nodes running REELibrarian.

Operator interaction is the same for both stand-alone systems and network systems.

Operators and Daily Operations

REELibrarian requires an operator to perform daily tasks. These tasks include:

- accepting volumes into the library
- returning volumes from the library
- mounting volumes on drives
- moving volumes between the various storage sites
- cleaning volumes and removing old volumes

Each operator must be identified to REELibrarian via the configuration program. When operator assistance is needed, REELibrarian locates one via its operator paging function.

The Environment File: `reelenv`

REELibrarian environment variables are stored in the `reelenv` file. By default, this file is located in `/etc/reelenv`, but you may have selected an alternate location for the file during installation. To determine the location of the `reelenv` file, enter the command:

```
echo $REELENV
```

The location of the `reelenv` file will echo to the screen. To view the contents of the `reelenv` file, enter the command:

```
cat $REELENV
```

Sample output for this command is shown below.

```
RLLIBDIR    /usr/local/lib
RLBINDIR    /usr/local/bin
RLPBASE     667777220
RL_MACH     sam
CLNTNAME    HOST
#RLLOGDIR   /usr/tmp
```

REELlibrarian library files are located in the directory specified by the `reelenv` variable `RLLIBDIR`. By default, this directory is `/usr/local/lib`, but you may have selected an alternate location during installation.

REELlibrarian binary files are located in the directories specified by the `reelenv` variable `RLBINDIR`. By default, this directory is `/usr/local/bin`, but you may have selected an alternate location during installation.

The `reel_env` environment reporter command is available for determining the contents of a single `reelenv` field. For example, the following command requests the location of the REELlibrarian library directory:

```
reel_env RLL
```

Sample output:

```
/usr/local/lib
```



Caution: REELlibrarian uses Remote Procedure Calls (RPC) to communicate between client and server processes. The `RLPBASE` entry in the `reelenv` file specifies a range of RPC program numbers that are used for this communication. The RPC program number specified in `RLPBASE`, as well as the next 100 numbers, must *not* be used by any other RPC application on your network. The client machines must have the same `RLPBASE` as the REEL master that serves them. So if there are multiple REEL masters, each must have its own distinct `RLPBASE` that matches its own clients. If you use the same `RLPBASE` on multiple servers, a client will respond to RPC broadcasts that come from the wrong master. This

will cause the client to reset and cancel REEL operations that are in process.



Note: For a complete description of the `reelenv` file, refer to the `reelenv(4)` manpage in Appendix C; for a complete description of the `reel_env` utility, refer to the `reel_env(8)` manpage in Appendix D.

Vault Management

The Vault Management System (VMS) is a fully integrated component of REELlibrarian—which is activated by the Administrator. The VMS performs the following functions without human intervention:

- tracks the location of tapes within and between vaults
- tracks movement of tapes within and between vaults
- tracks occupied slots within vaults
- tracks unoccupied slots within vaults
- re-slots tapes as they move from one vault to another

The VMS has a configuration interface and full reporting capabilities. Detailed information on these capabilities can be found in the `rlvms_report(8)` manpage in Appendix D.

The VMS should be used when a large number of volumes are handled manually. It is especially useful when vaults contain labeled slots. For example, an installation with three vaults A, B, and C, each containing slots labeled 1-20, would be an appropriate situation in which to utilize the VMS.

The VMS is controlled by a file named `rlvms_config` which defines the configuration of the vaults and sites available for VMS to use. This file contains the following information:

- `FORM_FACTOR` which declares the family media type acceptable (e.g., `34XX` to accept either 3480 or 3490)
- `MEDIA_TYPE` which lists the actual media capable of being contained in the `FORM_FACTOR`

- VAULT which declares the name of the vault and the vault type (which can be a Volume ID (VID), Volumeset Name (VSN), or actual slot number)
- SLOT which defines the FORM_FACTOR supported by the slot, its priority (1-100), the first slot number, and the slot count for the vault itself



Note: For more information on configuring the `rlvms_config` file, please see the *Configuration File - rlvms_config(4)* on page 62.

The `rlvms_config` file connects the REELlibrarian configuration program (`rlconfig`) and the VMS. The `rlconfig` program specifies the device, media, and storage sites available for VMS to use. The `rlvms_config` file organizes this data into a format which can be read and executed by VMS. A diagram of how REELlibrarian works with VMS appears in Figure 2-1:

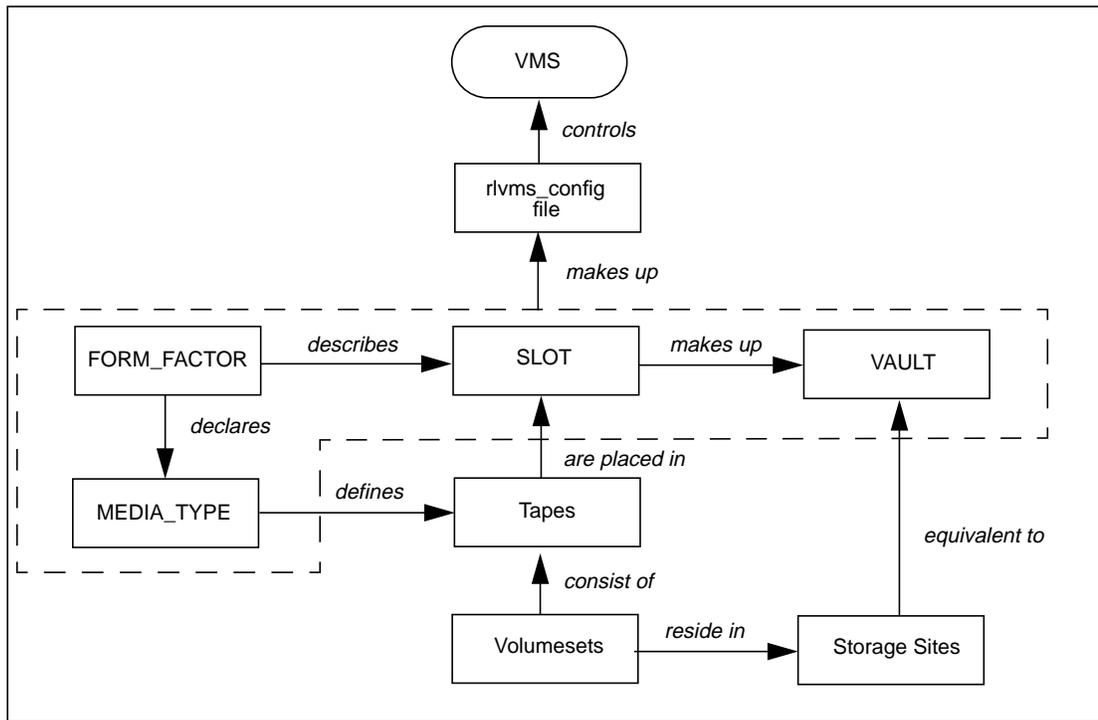


Figure 2-1 The Vault Management System (VMS)

A more detailed discussion on how to configure the VMS appears in *Configuring the Vault Management System* on page 55.

Running the Configuration Program

The configuration system is accessed via the `rlconfig` program.

Interaction & Special Keys

The various methods of entering data in the `rlconfig` screen and other REELlibrarian full-screen interfaces are listed in Table 2-3, *Full Screen Data Input Methods*.

Table 2-3 Full Screen Data Input Methods

Input Method	Description
Menu Selection	A list of items is presented and you enter the item number of your choice.
Master List	A small window is displayed with a list of entries. You may <code>[[a]]</code> add, <code>[[e]]</code> edit, or <code>[[d]]</code> delete these entries, or you may <code>[[q]]</code> quit the list and return to the configuration menu. Also, you may <code>[[j]]</code> move down one item or <code>[[k]]</code> move up one item.
Field Entry	Text input is required and you type your reply into a field.
Space-Bar	A choice is required and the few options are displayed in a line with one of the choices in reverse video (or with an asterisk). You can move the reverse video bar via the spacebar and indicate your selection by pressing <code>[[Return]]</code> .
Command Entry	The cursor is to the right of a “>” and a command will be requested from you. You respond by keying a single letter command.

Full-screen navigation keys are listed in Table 2-4, *Full-Screen Navigation Keys*.

Table 2-4 Full-Screen Navigation Keys

Name	Key	Description
ESCAPE	F1 or CNTL-A	Abandons the current activity and cancels any current data entry.
HELP	F2 or CNTL-B	Help information is presented on the screen.
FORM	CNTL-F	Enter and process the current menu or form. This is equivalent to keying a return in the last field on the menu or form.
REDRAW	CNTL-R	Redraws the current screen.
CLEAR FIELD	CNTL-O	Clears the contents of the current field.
BACK FIELD	CNTL-U	Move back one field.

The control key values can be reassigned via the keys file found in the file `rl_menus/keys` in the REELlibrarian binary directory. See the *Key Commands* section later in this chapter.

Starting the rlconfig Program

The configuration program `rlconfig` can only be run on REELlibrarian NetMaster nodes.

The configuration program is accessible via `rlconfig` command. To start the `rlconfig` program, simply issue the command:

`rlconfig`

The Configuration Menu is shown in Figure 2-2.

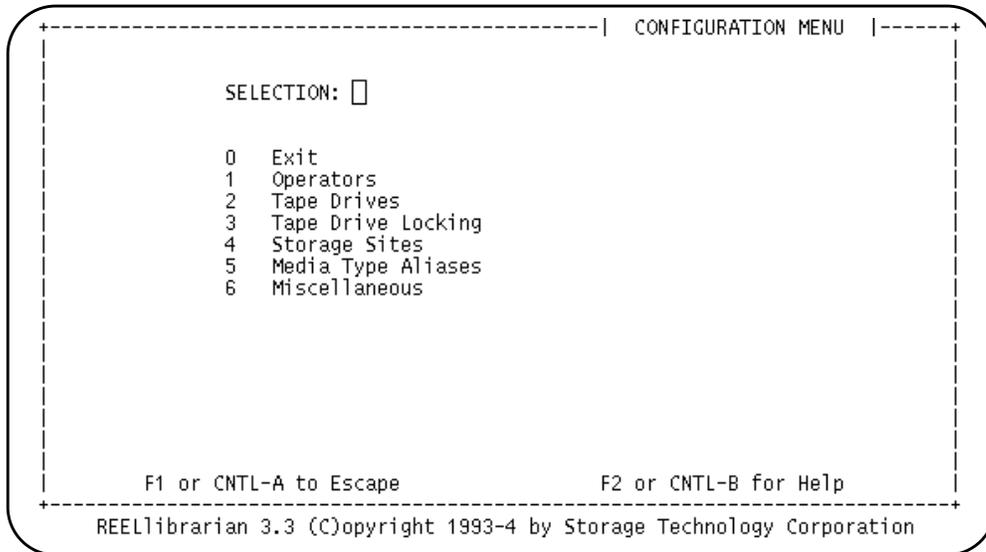


Figure 2-2 *rlconfig Program Menu*

To make a selection, enter the number of the desired item and press [[Return]]. Each item is explained in detail in the following pages.

Operators Operators perform mount requests and service the library - accepting and returning volumes, moving volumes, and cleaning volumes. REELlibrarian requires that all eligible operators be identified via the Operators screen. The screen is initially displayed covered by a Master List.

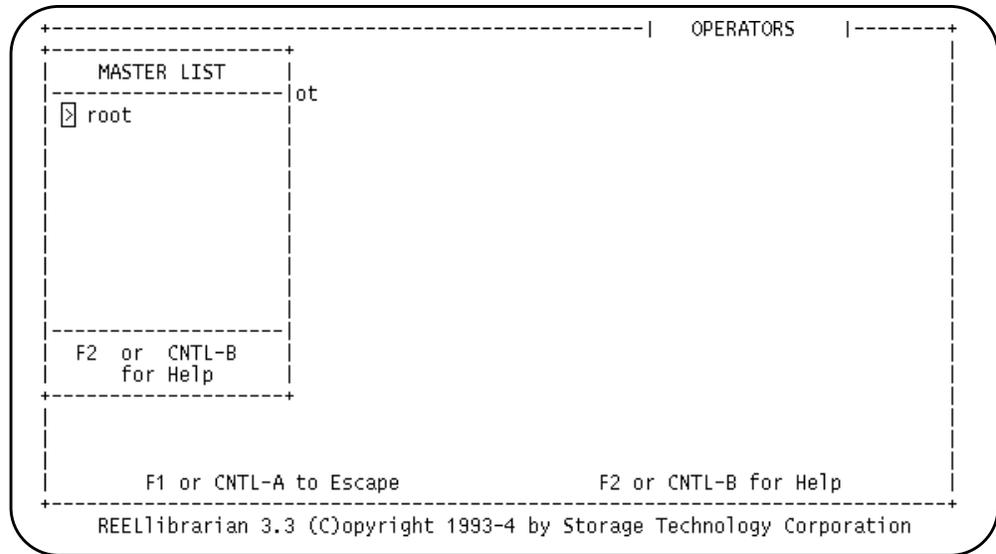


Figure 2-3 The Master List

The Master List contains the IDs of all the operators. The highlighted name is the current operator. Master List key commands are shown in Table 2-5, *Directional Keys*.

Table 2-5 Directional Keys

list:input:2	Description
a	Add a new operator to the list. The Operators screen is displayed for input.
d	Delete the current operator from the list.
e	Edit the current operator definition. The Operators screen is displayed for modification.
j	Move down the list to the next operator.
k	Move up the list to the previous operator.
q	Quit. Exit this screen and return to the configuration menu.

To add an operator, press a. This dismisses the Master List, revealing the Operators screen, as shown in Figure 2-4.

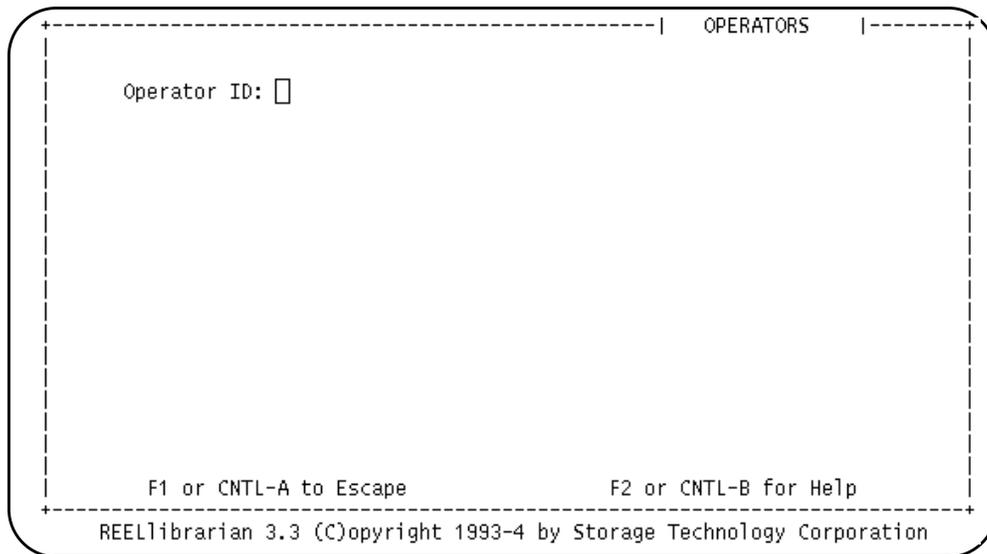


Figure 2-4 The Operators Screen

As will be the convention for the rest of this document, each screen is followed by a description of each field. The fields of the Operators screen are described in Table 2-6, *Operators Screen Fields*.

Table 2-6 Operators Screen Fields

Field	Description
Operator ID	Enter any valid user ID. Note: The root user does not need to be identified as an operator; the root user is automatically granted administrator and operator privileges. Example: reel



REELlibrarian employs a paging system to recruit an operator to act on incoming user mount requests if none are currently servicing the request queue. The paging system is invoked when there is no operator active and a mount request is received.

The paging system searches the current list of reserve operators and then pages each of these IDs wherever they may be on the network.

The paging system terminates when one of the reserve operators issues the `rlop` command or invokes the Request Monitor with the `rlmon` command. At that point, the respondent becomes the current operator.



Note: `rlmon` can only be run on the server machine running the REELlibrarian NetMaster.

An operator joins the list of reserve operators by running the command

`rlop -r`. This command is commonly placed in each operator's `.profile` or `.login` file.

Tape Drives

REELlibrarian requires that each tape drive be identified by an entry in the Tape Drives screen. When this screen is selected, it is initially displayed covered by a Master List of the currently defined drives, as shown in Figure 2-6. Master List key commands are shown in Table 2-5, *Directional Keys*.

When the Master List is dismissed (via add or edit commands), the Tape Drives screen is displayed. This screen is shown Figure 2-6. The fields of the Tape Drives screen are described in Table 2-7, *Tape Drives Field Definitions*.

Table 2-7 Tape Drives Field Definitions

Field	Description
Assigned Name (ADN):	A unique name up to twelve characters long. This name is the sole means of identifying the tape drive to operators. Example: drive1
Machine Name:	The name of the computer on which the drive resides. The name can be determined by using the UNIX command <code>uname -n</code> (or <code>hostname</code>) on the particular computer. <code>rlconfig</code> automatically determines this for the computer on which it is running and displays the appropriate name as the default. To accept the default, just press <code>[[Return]]</code> . Example: unix1 Note: Be sure to verify the existence of the hostname before entering it. This can be done by using the UNIX commands listed above. Otherwise, REELlibrarian will appear to hang as it searches for an unknown hostname.
Auto Density Detect	Should be set to YES if the drive automatically adjusts to the recording density of each mounted volume. Default is YES.
Auto Density Select	YES indicates that the drive switches between densities under software control. Default is NO.
Format List	See Table 2-8, <i>Device Format Field Definitions</i> .
Other Device List	See Table 2-10, <i>Other Devices Field Definitions</i> .



After entering the machine name, the cursor proceeds to the Format List Window, shown in Figure 2-7. This window accepts the Master List key commands described in Table 2-5, *Directional Keys*. Each entry represents a media type which the drive supports. Most often this represents the various recording densities available on a multiple density drive.

When an entry is added or edited, the Device Format pop-up window appears; this is shown in Figure 2-8. Fields of the Device Format window are described in Table 2-8, *Device Format Field Definitions*.

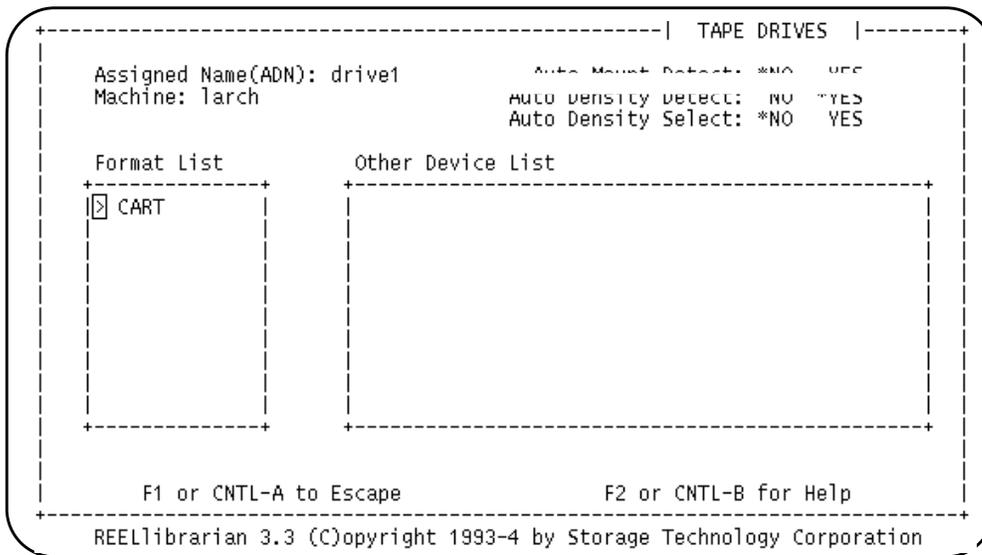


Figure 2-7 The Format List Window

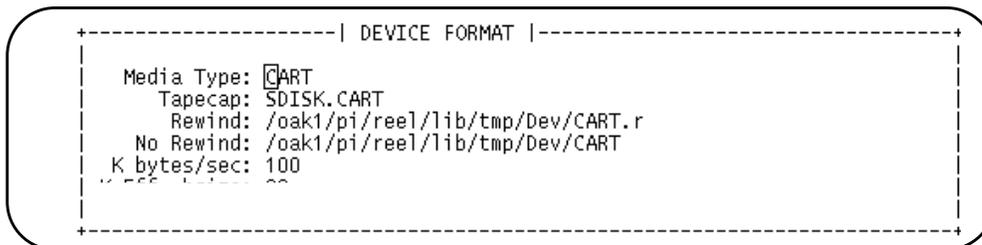


Figure 2-8 The Device Format Window

Table 2-8 Device Format Field Definitions

Field	Description
Media Type:	Enter any unique name up to nine characters long. This name indicates that the drive can support volumes of this type. For convenience, it is recommended that the media type name correspond to the recording density or cartridge standard. Example: 1600
Tapecap:	Enter one of the drive types listed in the Tape-cap Report. The drive type is analogous to a <code>termcap</code> entry for a terminal. Example: 1600
Rewind Name:	Enter the rewind device filename for the particular drive at the specified density. Example: <code>/dev/rmt/0m</code>
No Rewind:	Enter the no-rewind device filename for the particular drive at the specified density. Example: <code>/dev/rmt/0mn</code>
K bytes/sec:	Enter the maximum data throughput the drive delivers when writing a tape. Enter in units of kilobytes (1024 bytes) per second. Please see Table 2-9, <i>Device Throughput Speeds</i> for estimated throughput speeds for devices commonly used by REELlibrarian. Example: 100

Whenever a new drive is added or a drive is modified, the `rldtest` program should be run against it. It tests the drive and may indicate that a different type should be used for the particular device. See *Drive Configuration* on page 50 for more information.

The table below serves as a guide for some common devices and estimates of their typical maximum throughputs.

Table 2-9 Device Throughput Speeds

Device Type	Typical Maximum Throughput (Kbytes/sec.)
XBYTE	
8200	250
8200 (with compression)	500
8500	500
8500 (with compression)	1000
DAT	
4 mm	340
4 mm (with compression)	500
DAT (High Density)	
4 mm	250
4 mm (with compression)	500
QIC	100
3480	1000-3000 (depending on bus bandwidth)
3490	

The Other Device List contains any other file names that refer to the specified device. REELlibrarian needs these names in order to adequately secure the device through ownership changing. To get to the Other Device List, key q in the Format List.

The Other Device List operates just as the Format List. Entering q returns the user to the Master List. When a or e is entered, the Other Devices popup window is displayed; as shown in Figure 2-9. Fields of the Other Devices window are described in Table 2-10, *Other Devices Field Definitions*.

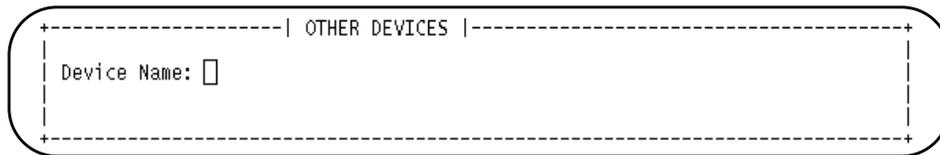


Figure 2-9 The Other Devices Window

Table 2-10 Other Devices Field Definitions

Field	Description
Device Name:	Enter the device name up to sixty characters long. Example: /dev/tape

Tape Drive Locking

When a drive is allocated to a user, REELibrarian changes the access permissions on the filesystem devices associated with the drive to prevent unauthorized access to the drive or the data. It also changes ownership of the device names to the particular user.

The Drive Locking screen controls which permission schemes are used for allocated drives, free drives, and the ownership of free drives. The screen is shown Figure 2-10. Fields of the Drive Locking screen are described in Table 2-11, *Drive Locking Field Definitions*.

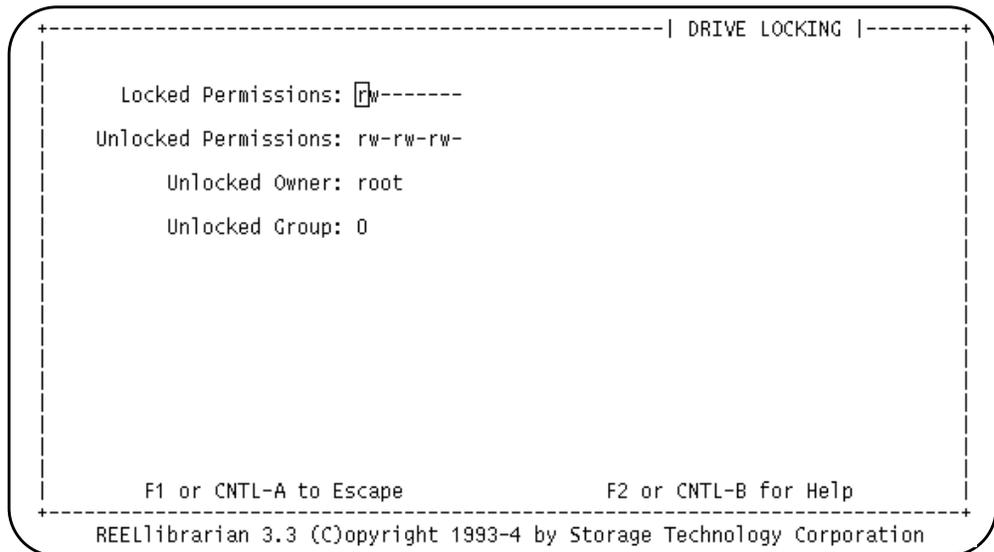


Figure 2-10 The Drive Locking Screen

Table 2-11 Drive Locking Field Definitions

Field	Description
Locked Permissions:	Enter the file permission scheme for allocated drives. REELlibrarian changes the permissions on each allocated drive's filesystem device names to this scheme. The permission scheme example permits only the user to read and write to the drive. Group members and others have no access.
Unlocked Permissions:	Enter the file permission scheme for free drives under REELlibrarian's control. REELlibrarian changes a drive's permissions to this scheme each time it returns to free status.
Unlocked Owner	Enter the user ID (either the name or the decimal number) which should be made owner of all free drives. When allocated, each drive's ownership is transferred to the particular user. When, afterward, the drive becomes free, its ownership is changed back to the Unlocked Owner value.
Unlocked Group	Enter the group ID (either the name or the decimal number) which should be made the group owner of all free drives. When allocated, each drive's group ownership is transferred to the particular user's group. When, afterward, the drive becomes free, its group ownership is changed back to the Unlocked Group value.

Storage Sites REELlibrarian provides for volume storage at multiple sites. Volume movement commands allow users to request movement and operators to confirm the actual transfers. The Storage Sites screen allows the administrator to define and control multiple storage sites.

When this screen is selected, it is initially displayed covered by a Master List of the currently defined storage sites. Master List key commands are shown in Table 2-5, *Directional Keys*.

When the Master List is dismissed (via add or edit commands), the Storage Sites screen is displayed. This screen is shown Figure 2-11. The fields of the Storage Sites screen are described in Table 2-12, *Storage Sites Field Definitions*.

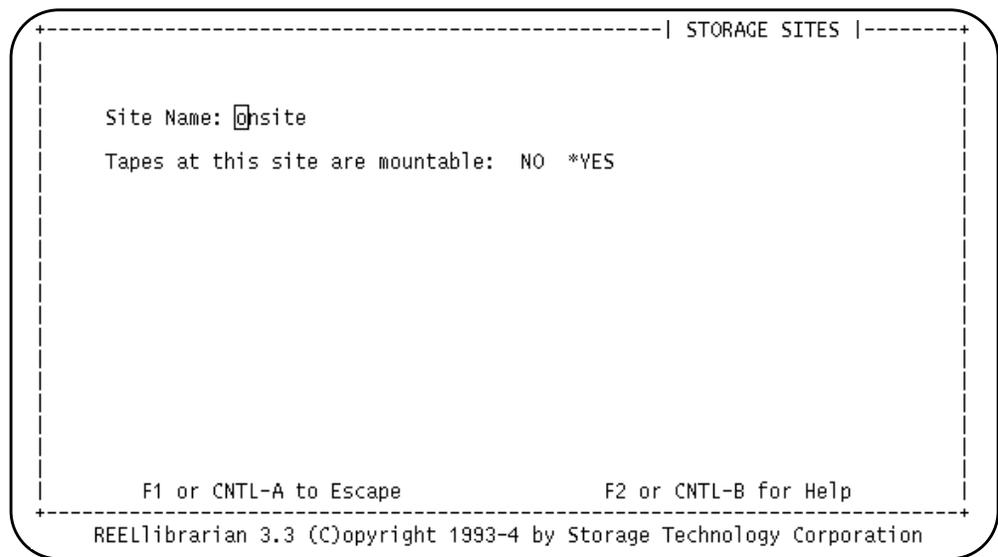


Figure 2-11 Storage Sites Screen

Table 2-12 Storage Sites Field Definitions

Field	Description
Site Name:	Enter a site name up to twelve characters long. Example: onsite.
Tapes at this . . .	Hit the spacebar to select No or Yes. If No is selected, then tapes stored at this site are not available for mounting. Mount requests naming tapes at this site are refused due to inaccessibility.

Media Type Aliases Each tape drive entry can support one or more media types. A media type typically specifies a form of media (reel or cartridge) and a recording density (e.g. 1600 bpi).

Volumes, however, can often support more than one storage density. For example, a reel (also known as nine-track) tape which is designed for 6250 bpi recording can also be used for 1600 bpi and 800 bpi recording.

In pools, available volumes are marked as being SCRATCH. A user can request a specific media type from the pool. If the pool does not have a volume of that media type the user is so notified. However, REELlibrarian provides an alias feature that allows it to substitute a volume with a different media type. Media Type Aliases allow the administrator to name which media types can be substituted for other media types.

When this screen is selected, it is initially displayed covered by a Master List of the currently defined media types that have been assigned aliases. Master List key commands are shown in Table 2-5, *Directional Keys*.

When the Master List is dismissed (via add or edit commands), the Media Type Aliases screen is displayed. This screen is shown Figure 2-12. The fields of the Media Type Aliases screen are described in Table 2-13, *Media Type Aliases Field Definitions*.

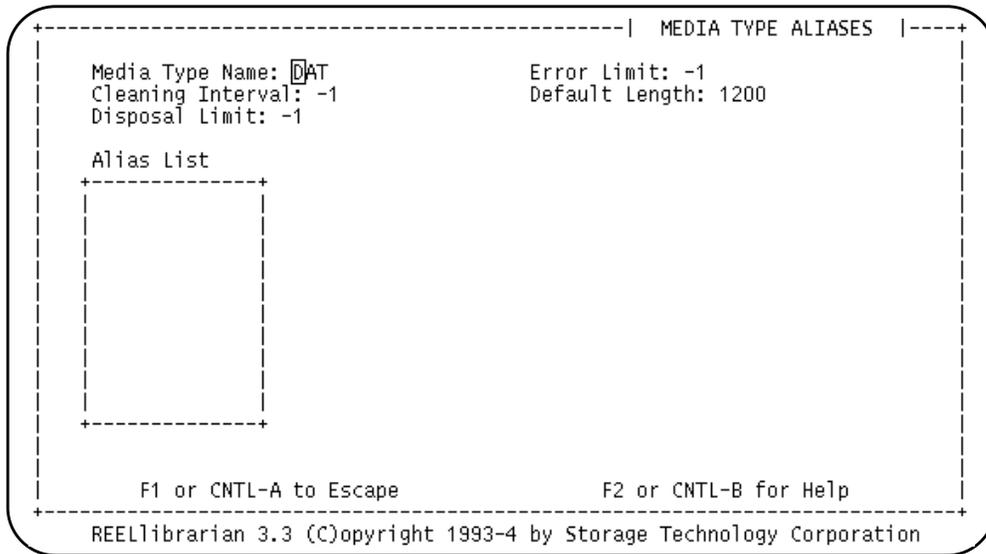


Figure 2-12 Media Type Aliases Screen

Table 2-13 Media Type Aliases Field Definitions

Field	Description
Media Type Name:	Enter the media type name up to nine characters long. Example: 1600
Cleaning Interval:	The volume will be scheduled for cleaning when its volumeset is scratched and the volume has been mounted more than this many times since its last cleaning. A value of 0 means cleaning will not be automatically scheduled. A value of -1 means the system default will be used. In order for this limit to be observed, the pool containing the volume must have been configured to observe automatic cleaning.
Disposal Limit:	The volume will be scheduled for removal when its volumeset is scratched and the volume has a total mount count greater than the disposal limit. A value of 0 means removal will not be automatically scheduled. A value of -1 means the system default will be used. In order for this limit to be observed, the pool containing the volume must have been configured to observe automatic removal.
Error Limit:	The volume will be scheduled for removal when the error limit is reached. Although the removal and certification bits are set in the volume catalog when the error limit is reached, the volume will not show up on the <code>rlr maint</code> report until the volumeset has been scratched. A value of -1 means the system default error limit will be used. A value of 0 means an infinite number of errors are allowed.
Default Length:	For nine-track or reel tapes, this is the default length in feet. For cartridge, 8mm, or DAT tapes, this is the default capacity in megabytes.
Alias List	See Table 2-14, <i>Alias Field Descriptions</i> .

Alias List The Alias List contains the media types which can be substituted for the designated media type. The list accepts the Master List key commands shown in Table 2-5, *Directional Keys*. A request to add (a) or edit (e) an entry invokes the Alias window. This window is shown in Figure 2-13. The fields of the Alias window are described in Table 2-14, *Alias Field Descriptions*.

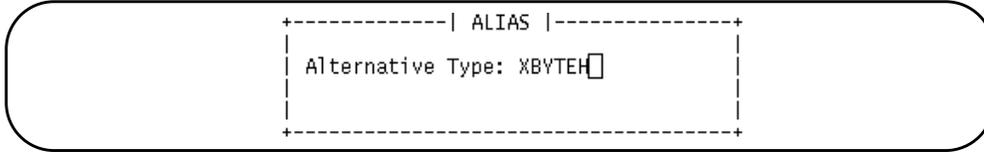


Figure 2-13 The Alias Window

Table 2-14 Alias Field Descriptions

Field	Description
Alternative Type:	Enter the media type name. Example: 6250

Miscellaneous The Miscellaneous screen controls the remaining REELlibrarian configuration parameters. It is shown in Figure 2-14. The fields of the Miscellaneous screen are described in Table 2-15, *Miscellaneous Field Definitions*.

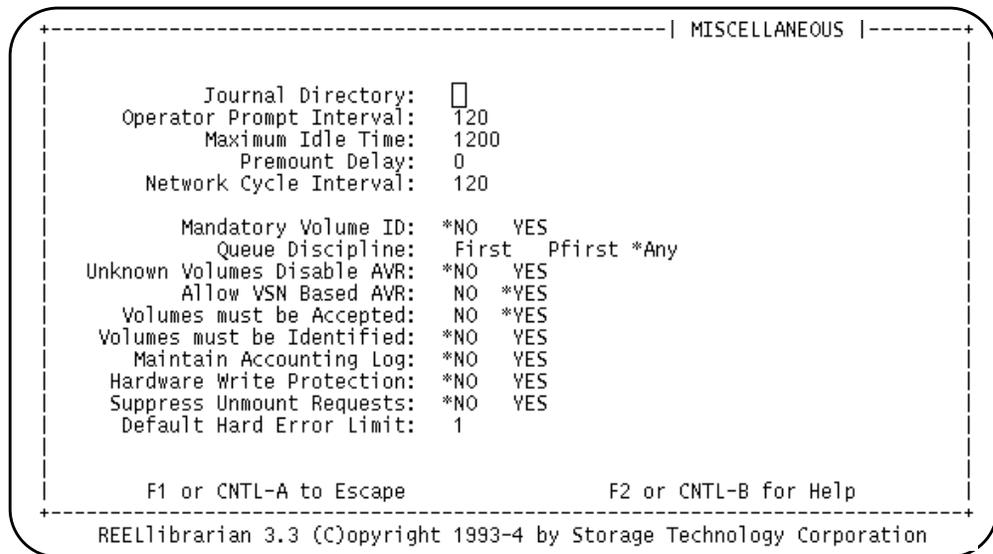


Figure 2-14 The Miscellaneous Screen

Table 2-15 Miscellaneous Field Definitions

Field	Description
Journal Directory:	<p>Enter the directory path name where REELlibrarian should keep its transaction journals. REELlibrarian keeps records of each transaction so that it can reconstruct its on-line catalog in case of damage. To increase probability of catalog survivability, it is recommended (but not required) that the journal directory be on a separate disk or disk partition from the library directory.</p> <p>The default is no path which means that journalling is disabled. Therefore, an existing path must be specified to use journalling.</p> <p>Example: /var/journals</p> <p>Note: See the RLbackup manpage for information on how to clear old catalog journal entries once the online catalog has been backed up.</p>



Table 2-15 Miscellaneous Field Definitions (Continued)

Field	Description
Operator Prompt Interval:	Enter in units of seconds the period between each paging call for an operator and operator mount/unmount prompts. Example: 60
Maximum Idle Time:	Enter in units of seconds the period that is the maximum amount of time a drive may remain idle before REELlibrarian reclaims it. When reclaimed, the current user is given an explanatory message. The default is 0 seconds, which disables the timeout.
Premount Delay:	Enter the number of seconds REELlibrarian should allow an inactive volume to remain mounted before it automatically instructs the operator to unmount it. An inactive volume can occur in two ways. The first way is when a user mounts the volume and then releases it. If another user request is not waiting on the drive, REELlibrarian leaves the volume mounted in the chance that it may be needed again. The second way is when an operator puts an unsolicited volume on-line.
Network Cycle interval:	Enter the period in units of seconds between each report from distributed nodes with tape drives. This item controls how often the network software communicates among the nodes. The shorter the period, the more often the reports. The recommended value is 120 seconds. Example: 120

Table 2-15 Miscellaneous Field Definitions (Continued)

Field	Description
Mandatory volume ID:	<p>Press the spacebar to select NO or YES. YES means that operators must always enter the volume ID when confirming a tape mount. NO means that operators only have to enter volume IDs to confirm mounts if REELlibrarian does not have a fingerprint for the volume. (See the “Volumes must be Accepted” field on how to generate fingerprints for each volume entering the library).</p>
Queue Discipline	<p>Press the spacebar to select one of the options. The request queue contains all outstanding mount/unmount requests in order by priority, and within each priority class by first-in-first-out order.</p> <p>Note: Priorities range from zero to nine (0 to 9); 0 priority is the highest.</p> <p>The <code>FIRST</code> option sets a discipline under which only the request at the top of the queue is examined for servicing. If it cannot be serviced, then the system waits until it can.</p> <p>The <code>PFIRST</code> option means that the system will try to satisfy any and all requests in the queue with a priority attribute equal to the priority of the request at the top of the queue. Requests with higher priorities will wait for servicing until all requests in the preceding priority class are satisfied.</p> <p>The <code>ANY</code> option directs the system to examine all requests in the queue and to service all that can be satisfied.</p>



Table 2-15 Miscellaneous Field Definitions (Continued)

Field	Description
Unknown Volumes disable AVR:	<p>AVR stands for Automatic Volume Recognition. AVR means that REELlibrarian will automatically try to identify any unsolicited volume mount. It reads the first part of the volume and checks it against the catalog for a match.</p> <p>This option, if set to YES, disables AVR if the catalog contains one or more volumes which have not been mounted and “fingerprinted” by the catalog for future AVR purposes. The “Volumes must be identified:” field prevents this situation from occurring.</p>
Allow VSN Based AVR:	<p>If set to NO, AVR is based on the fingerprint made by REELlibrarian in the catalog. If set to YES, AVR is based on only a volume’s Volume Serial Number (VSN). This is useful for sites with a large population of volumes predating the arrival of REELlibrarian.</p>
Volumes must be Accepted:	<p>If set to YES, the operator must accept submitted volumes via the <code>rlaccept</code> command or the <code>rlt1m</code> full-screen program. The volume may or may not be required to be identified as specified by the next field. For one-step volume submission, set this field to NO.</p>
Volumes must be identified:	<p>If and when the volume has been accepted, this option can mandate that the volume be mounted and fingerprinted for the catalog. The operator uses the <code>rlid</code> command to identify each volume.</p>
Maintain Accounting log:	<p>If set to YES, REELlibrarian maintains transaction log files in the library directory <code>ALog/</code>. The accounting files contents are documented later in this chapter.</p>

Table 2-15 Miscellaneous Field Definitions (Continued)

Field	Description
Hardware Write Protection:	If set to YES, then operators must enable the write capability when mounting tapes for write operations. Likewise, tapes mounted for read operations must be mounted with write protection. If set to NO, then the write protection capability is assumed to always be off.
Suppress Unmount Requests:	If set to YES, then unmount requests are disabled. When a tape is released, it remains on the tape drive until a subsequent mount request causes the operator to unmount it so that the newly requested tape can be mounted. This feature reduces what is normally a two step process - unmount then mount - to a single mount activity which implicitly unmounts the residual tape. If set to NO, then released tapes generate unmount requests.
Default Hard Error Limit:	The number of hard errors a tape volume can accrue before REELlibrarian schedules it for removal from the system. This is a global default for all tapes - individual media types can have their own hardware limits set via the MEDIA TYPES screen.

Drive Configuration

Each tape drive used by REELlibrarian must be identified by a tapecap entry. REELlibrarian is delivered with a set of standard tapecap entries that define the most common tape drives; these standard tapecap definitions are listed in the TapeCap Report, described on page 191. To generate a report of all defined tapecap entries known to your system, enter the command:

```
r1r tapecap
```

Confirming a Drive Entry: rldtest

The `rldtest` program exercises the tape drive to confirm that the configured drive type entry accurately describes the drive's capabilities. If the test fails, `rldtest` can be used to create a

custom drive type entry for the drive. Confirming a Drive Entry:
`rldtest`

After selecting a tapecap entry and entering it through the configuration program's Tape Drives screen, follow these steps to confirm the drive's configuration.

1. Enter the command:

```
rldtest
```

2. When prompted, enter the name of the device you are testing and press `[[Return]]`.
3. When prompted, mount a scratch tape on the designated drive (with a write ring in place) and press `[[Return]]`.

If "Test Complete" is displayed, then the drive is fine as configured. You can proceed with your use of REELlibrarian.

EOT Detection

REELlibrarian can take advantage of early end of tape notification when it is provided by the underlying hardware and tape driver. EOT notification enables REELlibrarian to close the current tape data file and, for a labeled tape, write trailer labels. Such notification is necessary to make effective use of the compression capability provided by many tape devices.

To find out if a specific tapecap entry employs EOT detection, request a tapecap report. Type:

```
rlr entry=tapecap_name tapecap
```

Substitute the name of the entry you want to check for *tapecap_name*.



Note: For more information on the tapecap report, refer to *Tapecap Report* on page 191.

Sample output for this command is shown Figure 2-15.

```
Variable block type device
Normally closed
Not Seekable
Not Byte Swapped
Min/Mod/Max Buffer:                20/1/32768
Inter block gap (mils):            0
Tapemark size (mils):              6000
Density (bpi):                     0
→ End of Tape Detection:           Calculated
Ioctl support (FSF/BSF/REW/EOD/EOF): 0/0/0/0/0
Non-standard 'capabilities' (0x14):
    Append at EOT only
```

Figure 2-15 The TapeCap Report

- If the EOT field (highlighted) reads `Calculated`, EOT detection is disabled.
- If the EOT field reads `Sensed by error error errno errno`, EOT detection is enabled.



Note: If EOT is enabled for a device, the value specified for tape media length/capacity in `rlconfig` is ignored; EOT automatically determines if the tape is full.

Configuring the `trusted_hosts` File

The `trusted_hosts` file is an optional configuration item which is recommended for sites with large networks. When properly configured, this file improves REELlibrarian's performance.

In the absence of a `trusted_hosts` file, the REELlibrarian servers communicate with client systems via a network broadcast call. The `trusted_hosts` file identifies the specific hosts REELlibrarian needs to service.

To construct the `trusted_hosts` file, create the library file `REEL/Librarian/trusted_hosts`. Enter into this file a list of all the host names of computers equipped with REELlibrarian, one

per line. The following example lists three computers under REELLibrarian's control.

```
north
south
east
```

Configuring Autoloader Tape Devices

REELLibrarian supports devices capable of automatic tape mounting. The site exit facility associates a given device with a set of three executable routines. At those times when an operator action request is made, i.e. mount, unmount, or device off-line, the system determines whether the device has an automatic routine to handle the action and, if so, executes it. The routine performs the action and acknowledges on successful completion, allowing REELLibrarian to continue without any intervention by the operator. If the routine fails it will not submit an acknowledgment. Then, the operator must complete the action and acknowledgment manually.



Note: Whether or not an automatic routine exists for a given device, the operator request messages are always sent to all registered operators. If the automated routines complete successfully they acknowledge the requests and no operator action is required nor should any be performed. If a routine fails it uses the `rlopmsg(1)` command to indicate to the operator the reason for the failure.



Note: The `stkmount` program will produce messages that appear on your console. If you do not want these messages to appear on the console (or want them to appear somewhere) else, edit the `stc_prolog` script's `$opmsg` variable to reflect where you want the message to appear (e.g., `rlopmsg=echo`). However, a value must correspond to the `$opmsg` variable, or the program will not execute.

To enable the autoloader support, you must edit the file `REEL/Librarian/site_exits` found in the REELLibrarian

library directory; if you do not know the location of this directory, issue the command:

```
reel_env RLL
```

`site_exits` contains multiple drive records, one per line of the form:

```
adn mount_path unmount_path offline_path
```

These variables are defined in Table 2-16, *The site_exits File*.

Table 2-16 *The site_exits File*

Variable	Description
<code>adn</code>	The ADN name of the tape drive as specified in the Tape Drives Screen. Note: If the ADN name is changed in the <code>site_exits</code> file, then it must be changed in <code>rlconfig</code> to match.
<code>mount_path</code>	The pathname of the mount site exit script. Use the value <code>none</code> if a site exit script is not needed.
<code>unmount_path</code>	The pathname of the unmount site exit script. Use the value <code>none</code> if a site exit script is not needed.
<code>offline_path</code>	The pathname of the off-line site exit script. Use the value <code>none</code> if a site exit script is not needed.



Standard Site Exit Scripts

The standard site exit scripts `stkmount` and `stkunmount` are found in the REELlibrarian binary directory. If you do not know the location of this directory, issue the command:

```
reel_env RLB
```

To configure a drive with the ADN `drive1` as a standard stacker device on a standard system, edit the `site_exits` file to contain the following single line entry:

```
drive1 RLBINDIR/stkmount  
RLBINDIR/stkunmount none
```

Refer to the manpage `site_exits(4)` in Appendix C, and manpages `stkmount(8)`, `stkunmount(8)`, and `posn_stack(8)` in Appendix D for more information on configuring your system to run with autoloaders.

Modifying Stacker Indexing Times

By default, the supplied site exit scripts wait 60 seconds for the next tape in the stack to be mounted. If your autoloader requires more or less time to mount the next tape in the stack, you may modify this period via the environment variable `RL_INDEX_TIME`.

For example, if you wish to change the indexing time to 30 seconds, before starting the servers, type:

```
RL_INDEX_TIME=30
export RL_INDEX_TIME
```



Note: To accommodate variable rewind times, set `RL_INDEX_TIME` to a few seconds longer than the indexing time for your device.

Configuring the Vault Management System

The Vault Management System (VMS) is a fully-integrated (and optional) component of REELlibrarian. The VMS performs the following functions without human intervention:

- tracks the location of tapes within and between vaults;
- tracks movement of tapes within and between vaults;
- tracks occupied slots within vaults;
- tracks unoccupied slots within vaults; and
- re-slots tapes as they move from one vault to another.

The VMS has a configuration interface and full reporting capabilities. For more information on these capabilities please see the `rlvms_report(8)` manpage.

For general information about the VMS and how it works with REELlibrarian, please refer to *Vault Management* on page 27.



Note: Vaults should be verified to be empty before deleting them. An effective way to do this would be to use the Volume Inventory Report.

For example, enter:

```
rlr vinventory | grep offsite
```

to check for volumes in the (*offsite* for instance) site.

VMS Implementation: Existing REEL Libraries

To implement vault management in an **existing** REELlibrarian tape library, follow these steps:

1. Create the file `rlvms_config` to define the vaults and slots available at your site. Refer to the `rlvms_config(4)` manpage for a complete syntactical description of this file. A practical example of using `rlvms_config(4)` is presented in *An Example Of Setting up Vault Management* on page 57.



Hint: Be sure to include in the configuration file vault and slot definitions for **all** existing volumes; failure to do this will cause the new file to be rejected by `rlvms_config`.

2. Run the command `rlvms_config(8)`; refer to the `rlvms_config(8)` manpage for details.

The REELlibrarian servers must be running when you issue `rlvms_config`, but the servers must be stopped when running `RLrebuild`.

3. Rebuild the REELlibrarian database with the `RLrebuild(8)` command. If the rebuild fails, it is possible that your `rlvms_config(4)` file does not define vaults and slots for all existing library volumes. If this is the case, modify the configuration, and attempt the `RLrebuild` command again.

VMS Implementation: New Installations

To implement vault management during a **new** REELlibrarian installation, follow these steps:

1. Create a temporary configuration file to define the vaults and slots available at your site, according to the specifications given in the `rlvms_config(4)` manpage. A practical example of `rlvms_config(4)` is presented in *An Example Of Setting up Vault Management* on page 57.
2. Activate the configuration with the command:

```
rlvms_config < file
```

where *file* is the name of the temporary file that contains the configuration.



Note: The REELlibrarian servers must be running when you issue this command.

An Example Of Setting up Vault Management

The following example presents a practical use of the Vault Management System. It is not a substitute for the complete syntactical description contained in the `rlvms_config(4)` and `rlvms_config(8)` manpages.

Before configuring VMS, use the `rlr` command to ensure that all media types in use and all sites defined by `rlconfig` are also defined in the VMS configuration. The following example demonstrates this technique.

1. Type:

```
rlr types > myfile  
rlr sites >> myfile
```

The file `myfile` will then look something like this, but will list only the media types defined in your site's configuration:

Media Type List: Tue Dec 6 12:08:11 1994

Type	Clean	Remove
-----	-----	-----
D6250		
D1600		
DCART		
DXBYTE		
DAT	-1	-1
DATHD	-1	-1
XBYTE	-1	-1
XBYTEHD	-1	-1
1600	-1	-1
800	-1	-1
6250	-1	-1
3200	-1	-1
3480	-1	-1
C20	-1	-1
C40	-1	-1
C60	-1	-1
C80	-1	-1
C100	-1	-1
C120	-1	-1
C150	-1	-1

Storage Location List: Tue Dec 6 12:08:17
1994

Name	Mountable
-----	-----
onsite	yes
offsite	no
safetape	no

2. Next, edit `myfile`. Remove the Clean, Remove, and Mountable columns, any media types that are not used at your installation, and the lines with dates. The remaining list shows all the media types available at your site. File `myfile` should then look something like this when you are done:

```
Type
----
XBYTE
XBYTEHD
C20
C40
C60
C80
C100
C120
C150
```

```
Name
----
onsite
offsite
safetape
```

3. Remove the Type heading and put a `FORM_FACTOR` heading before each group of media types that are physically similar. Make up a name for each group, and put it after `FORM_FACTOR`. Add the keyword `MEDIA_TYPE` before each media type in the list.

File `myfile` should then look something like this:

```
FORM_FACTOR xbyte
MEDIA_TYPEXBYTE
MEDIA_TYPEXBYTEHD

FORM_FACTOR qic
MEDIA_TYPEC20
MEDIA_TYPEC40
MEDIA_TYPEC60
MEDIA_TYPEC80
MEDIA_TYPEC100
MEDIA_TYPEC120
MEDIA_TYPEC150

Name
----
onsite
offsite
safetape
```

4. Remove the Name heading. Prefix each site (vault) with "VAULT" starting in column one. Append each VAULT line with "SLOT". File `myfile` should then look something like this:

```
FORM_FACTOR xbyte
      MEDIA_TYPEXBYTE
      MEDIA_TYPEXBYTEHD

FORM_FACTOR qic
      MEDIA_TYPEC20
      MEDIA_TYPEC40
      MEDIA_TYPEC60
      MEDIA_TYPEC80
      MEDIA_TYPEC100
      MEDIA_TYPEC120
      MEDIA_TYPEC150

VAULT onsite SLOT
VAULT offsite SLOT
```

5. Add a list of slots for each vault that physically describes the space available in the vaults. In the example below, the user had 10 shelves in each vault, 5 of which had space for 20 xbyte tapes and 5 that had space for 15 qic tapes. Some shelves were more easily accessible, and so were given priority over others.

```
FORM_FACTOR xbyte
      MEDIA_TYPEXBYTE
      MEDIA_TYPEXBYTEHD

FORM_FACTOR qic
      MEDIA_TYPEC20
      MEDIA_TYPEC40
      MEDIA_TYPEC60
      MEDIA_TYPEC80
      MEDIA_TYPEC100
      MEDIA_TYPEC120
      MEDIA_TYPEC150

VAULT onsite SLOT
```

```
SLOT xbyte 1 A-01 20
SLOT xbyte 2 B-01 20
SLOT xbyte 3 C-01 20
SLOT xbyte 4 D-01 20
SLOT xbyte 5 E-01 20
SLOT qic    1 F-01 15
SLOT qic    2 G-01 15
SLOT qic    3 H-01 15
SLOT qic    4 I-01 15
SLOT qic    5 J-01 15
```

VAULT offsite SLOT

```
SLOT xbyte 1 A-01 20
SLOT xbyte 2 B-01 20
SLOT xbyte 3 C-01 20
SLOT xbyte 4 D-01 20
SLOT xbyte 5 E-01 20
SLOT qic 1 F-01 15
SLOT qic 2 G-01 15
SLOT qic 3 H-01 15
SLOT qic 4 I-01 15
SLOT qic 5 J-01 15
```

Site Information

The following is true of the example tape library implementing the VMS system on the pages which follow (but not the site documented in the previous section, *Setting up Vault Management*):

- The tape library uses three storage locations (vaults):
- *onsite*: This vault is where most of the tapes reside. It is close to the system tape drives, and has five storage racks, each of which can accommodate 50 tapes. Total capacity for this vault is 250 tapes.
- *offsite*: This vault is located in a nearby building; it can accommodate 100 tapes.
- *safetape*: This vault is actually 100 slots leased from a tape storage service.
- The tape library currently uses only 3480 type cartridges but anticipates moving to 3490-type cartridges in the future.

- All storage slots are capable of holding either 3480 or 3490 tape cartridges.

Configuration File -
rlvms_config(4)

From the site information, the tape library administrator created the following `rlvms_config(4)` file:

```
FORM_FACTOR 34XX  
  
MEDIA_TYPE 3480  
  
VAULT onsite VID  
  
SLOT 34XX 10 on0100 50  
  
SLOT 34XX 10 on0200 50  
  
SLOT 34XX 10 on0300 50  
  
SLOT 34XX 10 on0400 50  
  
SLOT 34XX 10 on0500 50  
  
VAULT offsite SLOT  
  
SLOT 34XX 10 of0000 100  
  
VAULT safetape SLOT  
  
SLOT 34XX 10 sf0000 100
```

**Configuration
Explanation**

Form Factor and
Media Type
Specification

The first line of the configuration, `FORM_FACTOR 34XX`, declares the name of a media type, as found in the `fmts` file in the library directory. The subsequent lines, prefixed by `MEDIA_TYPE`, indicate that the form factor is capable of containing 3480-type media. This meets the site requirement for future compatibility with 3490 tape cartridges. To enable vaulting for media type 3490, a `MEDIA_TYPE 3490` line would have to be appended to the `FORM_FACTOR` after `MEDIA_TYPE 3480`.

Vault Specification The lines prefixed by `VAULT` declare the name of the vault and the vault type. Vault type can be one of `VID`, `VSN`, or `SLOT`. The `onsite` vault is declared as a `VID` type vault; this means that volumes in that vault are slotted by volume ID. When a volume moves into a `VID` type vault, it is always reassigned a slot number corresponding to its vid.

The `VSN` is the volume serial number, which is automatically generated by REELlibrarian.

The `offsite` and `safetape` vaults are both `SLOT` type vaults; they can accommodate volumes that have vids that do not match their slot values. When a volume is moved to one of these vaults, it is always assigned the first available free slot in the vault.

The vault specification scheme in this example effectively makes the vid, rack, and vsn value the same for each volume.

Slot Specification The lines prefixed by `SLOT` define the supported form factors, allocation priority, first slot number, and slot count for the associated vault. In this example, slot numbers will be used for volume serial numbers, so the slot numbers must be 6 characters or less.

Note that each storage rack in the vault `onsite` has its own slot declaration, forcing slot incrementing to begin at a specified value for each of the five racks. This allows the operator to use the first four characters of the slot address (`on02`, `on03`, etc.) to identify which of the five physical racks contains a specific volume. In this example, the prefix `onnn` was selected in preference to a prefix `onn` to allow for future expansion to more than ten racks. This example only uses lowercase alpha characters; lowercase characters are easier to type, and make it easier to distinguish between o (alpha) and 0 (numeric).



Note: Sites using REELlibrarian to interchange ANSI-labeled tapes should not use lowercase characters for the `VSN` field; this is not allowed by the ANSI standard.

VMS Operations at the Example Site

Adding Tapes

The tape operator at the example site wishes to submit an initial supply of 100 scratch tapes to the library. This is accomplished with the command:

```
rlpsubmit nvol=100 type=3480 vid=<vid> \  
vsname=vms rack=vms
```

where <vid> is the Volume ID of the first file you wanted added to the library.



Note: Setting the keywords `vms`, `vsname` and `rack` to `vms` causes those values to be selected by the vault manager.

Output for this command is shown in Figure 2-16.

```
Scratch Volume Submitted, Volume ID: os0100 (onsite/os0100)  
Scratch Volume Submitted, Volume ID: os0101 (onsite/os0101)  
Scratch Volume Submitted, Volume ID: os0102 (onsite/os0102)  
.  
.  
.  
Scratch Volume Submitted, Volume ID: os0247 (onsite/os0247)  
Scratch Volume Submitted, Volume ID: os0248 (onsite/os0248)  
Scratch Volume Submitted, Volume ID: os0249 (onsite/os0249)
```

Figure 2-16 VMS: Initial Tape Submission

All future tape additions that specify the value `vms` for `vid`, `rack`, and `vsname` will increment these values based on where the last `rlpsubmit` left off. In the event of tape removal, the next tape addition that specifies the value `vms` for `vid`, `rack`, and `vsname` will fill the hole automatically.

Moving Volumes

Use the `rlvmove` command to move volumes from one slot to another as shown in the following example:

```
rlvmove location=offsite <volset>
```

where `volset` is the name of the volumeset you want to move.

Checking for Volumes Awaiting Movement

The tape operator at the example site wishes to view a list of tapes that are awaiting movement. This is accomplished with the command:

```
rlr act=move maint
```

Output for this command is shown in Figure 2-17.

```
Volume Maintenance Report: Wed Feb  2 15:00:12 1994 (RL3.2.3)

  action= move
  location=
```

Move:	From	To	VID
	----	--	---
	onsite/os0204	offsite/of0000	os0204
	onsite/os0205	offsite/of0001	os0205
	onsite/os0206	offsite/of0002	os0206
	onsite/os0200	safetape/sf0000	os0200
	onsite/os0201	safetape/sf0001	os0201
	onsite/os0202	safetape/sf0002	os0202

Figure 2-17 VMS: Tapes Awaiting Movement



Note: The `To` field of the volume maintenance move report now contains both the destination vault and the assigned slot within the vault. The assigned slot is selected by the vault manager when the volume is scheduled for movement.

Confirming Volume Movement

The tape operator at the example site wishes to confirm the movement of the tapes shown in the volume maintenance move report. This is accomplished with the following commands:

```
rlmoved loc=offsite os0204 os0205 os0206
```

```
rlmoved loc=safetape os0200 os0201 os0202
```

Output for these commands is shown in Figure 2-18.

```

$ rlmoved loc=offsite os0204 os0205 os0206

Volume os0204: Moved
Volume os0205: Moved
Volume os0206: Moved

$ rlmoved loc=safetape os0200 os0201 os0202

Volume os0200: Moved
Volume os0201: Moved
Volume os0202: Moved

```

Figure 2-18 VMS: Confirming Volume Movement

Issuing the `rlmoved` command as above moves the specified volumes into slots selected by the vault manager. To move volumes to specific slots, use the syntax:

```
rlmoved loc=offsite os0100/of0001
```

The command shown above declares that volume `vid=os0100` now resides in slot `of0001` in the `offsite` vault. In the above case, if volume `os0100` was pending movement to the `offsite` vault, even to a different slot, the pending move will be satisfied. If the volume is pending movement to a vault other than `offsite`, the pending movement request will remain pending.



Note: Under normal circumstances, it is best to allow the vault manager to select the destination slots.

Deactivating Vault Management

To deactivate vault management, perform the following steps as `root`:

1. Type:

```
reel stop
```

2. Type:

```
rm `reel_env`
RLL`/REEL/Librarian/rlvms_config
```

3. Type:

```
Rlrebuild -v
```



Note: The `Rlrebuild` with the `-v` option works significantly quicker than when rebuilding an entire catalog.

Other Configuration Issues

System Capacity

When configuring REELlibrarian, it is important to consider the limitations of the software. Although REELlibrarian can accommodate large numbers of tapes, volumesets, files, drives, and pools, there are system limits. If a limit is exceeded, a system error message will alert you. REELlibrarian capacity limits are listed in the table below.

Table 2-17 REELlibrarian Capacity Limits

Item	Limit
Tapes	4 million tapes
Tape Files	6 million tape files
Volumesets	4 million volumesets
Tape Pools	1 million tape pools
Tape Drives	256 drives

Terminal Displays

REELlibrarian works with most terminals by using the `termcap` database. If it can, it will use character graphics to give menus and forms smooth border lines and other graphic features. To do so requires that the `termcap` entry for the user's terminal be correct. Specifically this requires definition of the `termcap` capabilities `as` and `ae`. These capabilities are *alternate set start* and *alternate set end*. They represent the special sequences the terminal requires to display character graphics. If `termcap` does not have these

defined, then the screens will be created with regular keyboard characters.

For example, the correct `as` and `ae` entries for the DEC VT-100 are:

```
as:eE(0:as:eE(B:
```

The complete VT-100 termcap entry might look like this:

```
d0|vt100|vt100-am|dec vt100:\
:cr=^M:do=^J:nl=^J:bl=^G:co#80:li#24:\
:cl=50E[;HeE[2J:le=^H:bs:am:cm=5eE[%i%d;%dH
:\
:nd=2eE[C:up=2eE[A:ce=3eE[K:cd=50eE[J:\
:so=2eE[7m:se=2eE[m:us=2eE[4m:ue=2eE[m:\
:md=2eE[1m:mr=2eE[7m:mb=2eE[5m:me=2eE[m:\
:is=eE[1;24reE[24;1H:rf=/usr/lib/tabset/vt1
00:\
:gr=eE(:gs=0:ge=B:go=_:\
:rs=eEeE[?3leE[?4leE[?5leE[?7heE[?8h:\
:ks=eE[?1heE=:ke=eE[?1leE:\
:ku=eEOA:kd=eEOB:kr=eEOC:kl=eEOD:kb=^H:\
:ho=eE[H:k1=eEOP:k2=eEOQ:k3=eEOR:k4=eEOS:\
:ta=^I:pt:sr=5eEM:vt#3:xn:\
:sc=eE7:rc=eE8:cs=eE[%i%d;%dr:\
:as=eE(0:ae=eE(B:
```

In addition, REELlibrarian uses a file named `rl_menus/trantabl` located in the REELlibrarian binary directory. If you do not know the location of this directory, issue the command:

```
reel_env RLB
```

This file contains special entries for numerous terminals. If the character graphics do not work, examine this file. If an entry corresponds to the specific terminal being used but differs only in name, duplicate the entry in the `trantabl` file and change the name appropriately. If character graphics still do not work, call customer service for assistance.

Key Commands The special key commands recognized by the REELlibrarian full-screen programs `rlconfig` and `rl` can be modified through the

file `rl_menus/keys` in the REELlibrarian binary directory. The default configuration of the file is:

```
redraw=^R
escape=^A
help=^B
top=^U
form=^F
toggle=^T
clear=^O
```

To alter any key, edit the keys file and substitute the desired control key value for the current value.



Note: Control characters must be entered as a single character, not as a '^' followed by an alphabetic character. In the editor `vi`, this is accomplished by keying `[[CRTL-V]]` followed by the desired control character (`[[CRTL-A]]` for example).

Administration

Using Tracing Facilities

REELlibrarian contains extensive built-in tracing facilities for diagnosing problems with the software. Incidents can often be resolved by turning on tracing and sending the trace files to customer support.



Note: Trace files are for the specific purpose of aiding customer support to diagnose incidents and are not meant for diagnosis by the customer. Therefore, tracing facilities should be activated only upon the direction of StorageTek customer support and should be deactivated once the needed trace files have been generated.

Be aware that REELlibrarian trace files can become very large. Be sure that you have ample disk space prior to turning on tracing.

Turning Tracing On

Follow these steps to turn on tracing:

1. Decide upon the appropriate level of tracing. REELlibrarian supports nine levels of tracing—the higher the number, the more information kept in the trace files.
2. Tracing can be enabled for the entire REELlibrarian system, or alternately, tracing can be enabled for a particular REELlibrarian process. Decide whether to trace the whole system or just a single process.
3. Edit the `reelenv` file to set the appropriate environment variables. REELlibrarian does not need to be restarted in order to implement the change in tracing levels. It will automatically recognize the change in the tracing levels within a few minutes after you modify `reelenv`.

The following table lists the `reelenv` variables which control tracing.

Table 2-18 *reelenv* Tracing Variables

Name	Description
<code>RLOG_LVL n</code>	Turns on dynamic tracing for the entire system.
<code>RLOG_[cmd pid] n</code>	Turns on selective tracing for a specific command or process ID.

For example, the following `reelenv` file:

```

RLLOGDIR      /usr/tmp
RLOG_LVL      0
RLOG_RLnet    9

```

turns tracing off for all REELlibrarian processes (`RLOG_LVL 0`) and enables level nine tracing for the `RLnet` daemon (`RLOG_RLnet 9`).

It is also possible to enable tracing for a specific process ID. For example:

```

RLLOGDIR      /usr/tmp
RLOG_LVL      0
RLOG_12345    9

```

enables level nine tracing for process 12345.



Note: When requesting selective tracing, you must always specify an `RLOG_LVL` of 0; `RLLOGDIR` alone requests level nine logging for all processes and programs.

For more information on the `reelenv` file, please refer to the `reelenv` manpage in Appendix C, “File Format Man Pages”.

Trace File Disposition

The `reelenv` variables `RB_LEAVE_LOG` and `RL_NOTSHORT` control trace file disposition.

By default, trace files for programs that terminate successfully are deleted. If `RB_LEAVE_LOG` is set, these files are retained.



Note: Trace files for commands or processes that have their trace level set explicitly in `reelenv` via `RLOG_cmd` or `RLOG_pid` will not be removed regardless of `RB_LEAVE_LOG`.

By default, REELlibrarian trace files begin to wrap after 200,000 characters. If `RL_NOTSHORT` is set, logs do not wrap.

Accounting Log

REELlibrarian maintains an accounting log which records tape resource activities. The log consists of daily files kept in the library directory `Allog/`. File names are in the format year-month-day, `yy.mm.dd`. Each file contains transactions for that day only. The types of entries are enumerated in the table below.

Table 2-19 Accounting Log Entries

Format	Description
<code>(hh:mm:ss) uid gid SUB vid fmt</code>	Tape Submission. Indicates when a user submitted a tape, the volume ID assigned and the tape’s format.
<code>(hh:mm:ss) uid gid RET vid fmt</code>	Tape Retrieval. Indicates when a user retrieved a tape.
<code>(hh:mm:ss) uid gid MNT dev fmt vid</code>	Tape Mount. Indicates when a tape was mounted on a drive.

Table 2-19 Accounting Log Entries (Continued)

Format	Description
<i>(hh:mm:ss) uid gid MOV fmt vid from to</i>	Tape Move. Indicates when and where a tape was moved.
<i>(hh:mm:ss) uid gid CLN fmt vid</i>	Tape Clean. Indicates when a tape was cleaned.
<i>(hh:mm:ss) uid gid ALL pool fmt</i>	Tape Allocation. Indicates when a tape was allocated from a pool to a user's volumeset.
<i>(hh:mm:ss) uid gid RES dev fmt nsec</i>	Device Reservation. Indicates when a reserved device was released and the number of seconds it was under the user's control.
<i>(hh:mm:ss) QST</i>	Server Start. Indicates when the NetMaster servers were started.

All entries are prefixed with a time stamp of the format "(hours:minutes:seconds)". Shown below is a sample log file.

```
(12:57:14) QST
(12:57:40) root SUB keen-397 1600
(12:58:02) root RES drive1 1600 1831
(13:01:44) root RES drive1 1600 655
(13:01:53) root MNT drive1 1600 keen-397
(13:04:39) root MOV 1600 keen-397 onsite offsite
```

The Catalog Journal and Backups

A journal of all catalog transactions is kept by REELlibrarian. In the event that the catalog is corrupted, the servers automatically reconcile the catalog against the journal.

To ensure survival of the catalog, it is strongly recommended that the journal be kept on a disk or disk partition separate from the library directory. The location is specified at installation time and is also controlled by the Miscellaneous Screen in `rlconfig`.

Controlling Journal Size

The journal keeps every transaction logged to it until the `RLbackup` command is run. This command clears out all entries from the past up to yesterday's entries. This command should only be run when the catalog has been backed up. Therefore, in the event of disaster, the backed up version of the catalog can be restored, and a current catalog constructed from it and the Journal.

It is recommended that the standard backup process include running this command each time the catalog is backed up. The catalog is located in the library directory `Tdb/`.

Server Failure and Recovery

The REELibrarian server maintains the catalog. Certain events like power failures can cause unexpected interruptions. The server failure can result in damage to the catalog that requires recovery action. When this is the case, the REELibrarian server complains on startup with one of the following messages. If one of these messages appears, follow the solution description.

- `RL0001 - get_vinfo: seek error`

Solution: run the `RLrebuild` command as detailed in the next subsection.

- `RL0002 - get_vinfo: read error`

Solution: run the `RLrebuild` command as detailed in the next subsection.

- `RL0003 - get_vinfo: time error`

Solution: run the `RLrebuild` command as detailed in the next subsection.

- `RL0004 - Record newer error`

This error indicates that the catalog shows a last modification time located far in the future relative to the current system time. It may be that the current system time is actually set way in the past. If so, reset the system time and restart the servers.

- RL0005 - Record check error

Solution: run the `RLrebuild` command as detailed in the next section.

Recovering the Catalog: The RLrebuild Command

`RLrebuild` reconstructs the catalog by examining the transaction Journal and reconciling it against an older version of the catalog. Any areas of disagreement are resolved to reflect the journal's view. The command also completely rebuilds the catalog by creating a new version of it - this allows it to examine each catalog entry for correctness.

So, to reconstitute the catalog so that the servers can run:

1. Recover the catalog from the backup system. The catalog consists of the library directory `Tdb/` and all descendent files.

2. Run the command:

```
RLrebuild
```

3. When `RLrebuild` finishes, restart the servers.

```
reel start
```

License Control/Adding New Clients

REELlibrarian software is controlled by licensing restrictions. It is shipped with the specific licensing parameters for the purchased configuration. These licensing parameters are reviewable and upgradable. They are described on a product basis below.

REELlibrarian NetMaster

The command `RLlicense` produces a report on the license parameters for REELlibrarian NetMaster.

RLlicense info

```
Application: REELlibrarian  
Machine: myown  
Serial no: 10000000  
Vendor: StorageTek  
Nodes: A(0)B(0)C(0)D(0)E(3)  
License: purch  
Maint. Rel.: A00
```

The Machine name is the name of the node on which the software is installed. The software cannot be moved without upgrading the software license. This requires contacting the REELlibrarian supplier to get the appropriate upgrade code.

The Nodes field indicates how many REELlibrarian NetClient licenses of each machine class the REELlibrarian NetMaster software is licensed to support. In the example, this particular copy can support three Class E NetClients. If you need information on the various machine classes, ask your REELlibrarian supplier for details.

The License field denotes the type of license: `perch` means the license is a perpetual license, `demo` means it is a temporary license which becomes disabled after a fixed period (usually thirty days) - the `demo` license is renewable by contacting the supplier.

Upgrading for NetClient Support

The REELlibrarian NetMaster software can be upgraded to support additional REELlibrarian NetClient nodes. It requires that additional licenses be purchased and an upgrade code be provided by the software vendor. Before upgrading contact the vendor for licensing and code instructions.

Run the command as shown here:

```
RLlicense node
```

The command prompts for machine class and node capacity information. It also asks for the upgrade code (provided by your REELlibrarian supplier).

REELlibrarian NetClient

Each REELlibrarian NetClient software copy comes coded for a specific machine class. It can only be supported by a REELlibrarian NetMaster which is licensed to support a REELlibrarian NetClient of that class or greater.

Changing/Updating a Hostname

On occasion, a hostname will need to be changed or updated. This change can occur on a client or a master. When doing this, you must consider the impact of the change on all system running REELlibrarian, especially if the change is made on the master.

Failure to do so will result in loss of data and network communication problems.

Follow these steps when updating the hostname:



Note: These changes should be made *before* starting REELlibrarian on the host whose name is changing.

1. Edit the `reelenv` file to reflect the change in hostname. If the name of the master is changing, edit the `RL_MACH` and `CLNTNAME` fields. If the name of a client is changing, edit that client's `CLNTNAME` field.



Note: If the hostname of the master is being changed, then the `reelenv` file on all affected systems (or clients) must be updated to reflect this change.

2. On the client, type:

```
cd `reel_env RLL`/REEL/Clients;  
mv <oldhostname> <newhostname>
```

to move the stream catalogs to the new hostname.

3. After bringing up the system, use the `rlconfig` program to replace the old hostname under Tape Drives.

REELbackup & REELlibrarian Issues

REELbackup is data backup and recovery software and is a sister product to REELlibrarian. If both are running on the same network, some overlap occurs. This affects your use of REELlibrarian in that some of its functions are essentially the same functions found in REELbackup.

By convention, all REELbackup functions begin with `rb`; all REELlibrarian functions start with `rl`.

REELbackup and REELlibrarian share many resources. For example, they reside in the same `/bin` and `/lib` location and share database and configuration resources. However,

REELbackup and REELlibrarian have distinct differences, including:

- backup catalogs on each node (REELbackup only)
- journal directory (REELlibrarian only)
- streams database (REELbackup only)
- pool ownership: REELbackup must have all of its tapes in a pool called backup and this pool must be owned by root—whereas, REELlibrarian can have pools named anything except backup and can be owned by anyone in the network.

In summary, REELlibrarian controls and administers REELlibrarian directories and files while REELbackup handles REELbackup directories and files and backup catalogs found on each node. REELbackup doesn't permit tapes to be imported into its control that were not created by REELbackup. For example, if a tape from REELlibrarian is submitted to REELbackup as a valid backup tape, REELbackup will have no way of being able to read what's on the tape. The tape will not have the information concerning streams, catalogs, or crash recovery that is required by REELbackup. In addition, the tapes will be incompatible because REELbackup has a unique method of writing tapes whether it is in sctype, tar, or cpio format.

Instructions for using REELbackup are provided in the *REELbackup Master Guide*.

Command Summary

The table below summarizes the administrator programs. Refer to the manpages in Appendix D for complete descriptions of each command.

Table 2-20 Administrator Programs

Program	Description
rlconfig	Full-screen configuration program that controls all REELlibrarian parameters.

Table 2-20 Administrator Programs (Continued)

Program	Description
rl	Full-screen interface to all user and operator programs.
RLbackup	Backup the catalog.
RLrebuild	Rebuild the catalog
RLdump	Dump RL state information.
RLlicense	License management program.
rldtest	Tests tape drives.
rlr	Generates all REELibrarian reports.
rlvms_conf ig	Validate and update VMS catalog.
rlvms_conf irm	Inform the VMS of the physical location of a volume.
rlvms_move	Initiate movement of a volume to a new vault.
rlvms_repo rt	Generate VMS catalog report.
rlvms_retr ieve	Delete volumes from the VMS catalog.
rlvms_subm it	Submit volumes to the VMS catalog.

Chapter 3. Operations

Overview

This chapter describes the duties of the REELlibrarian operator. The operator provides regular, ongoing assistance to the REELlibrarian software and to the REELlibrarian users. The REELlibrarian operator:

- services the mount request system by mounting and unmounting tapes on system tape drives
- performs library maintenance activities such as volume entry, exit, movement, cleaning, erasure, and disposal
- controls network status via the server programs

Servicing Mount Requests

The mount request system collects user mount requests and assigns appropriate tape drives. Each request is presented in turn to the operator for action on the request monitor (a full-screen display of outstanding requests and tape drive activities). The operator fetches the requested volume from storage, mounts it on the designated drive, and confirms the action. A message informs the user that the volume is mounted and ready for use. The operator interacts with the MRS constantly through the `rlmon` full-screen program (the request monitor) or through the command-level programs `rldone`, `rlskip`, and `rls`.

Although the REELlibrarian administrator can designate more than one library operator, There may only be one operator at any given time. If the current operator logs out or neglects the outstanding operator instructions, REELlibrarian pages all logged-on, qualified operators. It continues paging until one of the

qualified operators becomes the current operator via the `r1mon` or `r1op` commands.

Tending the Tape Library

REELlibrarian schedules volume movement, cleaning, and removal activities on a daily basis. These activities require operator assistance:

- volume movement
- volume cleaning
- volume removal

The REELlibrarian report command `r1r` details outstanding library tasks. The operator confirms library task completion through the full-screen program `r1` or the commands `r1moved`, `r1cleaned`, and `r1removed`.

The operator processes the entry of volumes into the library and their exit from the library. For a volume to enter the library, the owner must submit it, and the operator accept it. For a volume to exit the library, the owner must retrieve it, and the operator return it.

Controlling the Network Servers

A network server is a program that runs at all times and provides requested services to client programs. REELlibrarian conducts its activities on the network via two server programs: `RL` and `RLnet`. `RL` runs on the master or server node. `RLnet` runs on the master node and every other network node. All other REELlibrarian programs act as clients to these servers. Therefore, it is imperative that the servers be adequately monitored so services are maintained.

The r1 Program

All operator tasks can be conducted through the `r1` program. It provides a master menu as a gateway to the request monitor and the tape library management screens. Alternatively, there is a separate command-line interface consisting of several programs which duplicate the `r1` facilities.

Operator Paging

REELlibrarian allows more than one person to be an operator. A paging system is employed to recruit an operator, if no operator is currently active, to process incoming tape requests. Paging is invoked when there is not an active operator responding to a current mount request.

The paging system pages each of the current list of reserve operators with a message requesting service on the request queue.

Need Tape Operator

The paging system terminates when one of the operators issues the `rlop` command or the `rlmon` command. At that point, the respondent becomes the active operator.

Qualified operators join the list of reserve operators via the `rlop` command; type:

```
rlop -r
```

Reserve operators can be located anywhere on the network and will be paged across the network by the REELbackup server. However, the request monitor (`rlmon`) must be run on the server node.

Controlling the Network

REELlibrarian conducts its network operations through the use of server programs and network daemons. The server programs run on the master or server node. The network daemons run on each node in the network. If these programs are not running, tape activities can fail.

Usually, these programs are configured to start on each computer when it is booted. However, due to various factors some nodes may become inactive because one or more of the programs has terminated. The following pages describe how to detect any program absences and how to start and stop them.

Checking Network Status

To verify that the NetMaster server is running, use the `rls` command; type:

```
rls
```

Sample output is shown below.

```
Request Q: Empty
```

```
Device Status:
```

ADN	Type	Stat	VID	UID	Key	Psd
---	----	----	----	----	----	----
Adrive1	1600	off	-----			
Bdrive2	1600	off	-----			
Cdrive3=	1600	off	-----			

If the server is down, `rls` reports it, otherwise a report similar to the one above is displayed.

If any of the network nodes are not responding, the drives associated with that machine will have an equal sign (=) appended to their name. In the preceding example, the drive `Cdrive3` is so marked. Restart the server by logging into the drive's system and run the `reel start` command. Then, run the `rls` program again and verify that the drive is now available - its equal sign suffix should be gone.

Controlling Server and Client Processes

Checking Node Program Status

Use the `reel` program to determine if the REELlibrarian services are running on any node. Consider the following example run on the server node.

```
reel
```

```
Checking status of REELlibrarian servers:
```

```
Checking rllog Network Log Daemon server status:
```

```
rllog server is up.
```

```
Checking RL Mount Request Daemon server
status:
```

```
RL server is up
```

```
Checking RLnet Network Daemon server status:
RLnet server is up.
```

If the servers are down, start them as described below.

Starting Node Programs

Note: Only the root user can stop or start the servers.

Start REELlibrarian services on any node, whether it be the server or a client, by logging into the node and executing the `reel` program.

```
reel start
```

Stopping Node Programs

To stop REELlibrarian services on a node, login to the node and use the `reel` program.

```
reel stop
```

Servicing the Request Monitor

REELlibrarian allows users to make ad-hoc requests for mounts and unmounts. The requests are queued and presented one at a time to the current operator for action.

Full-Screen Operation: `rlmon`

The program `rlmon` displays the Request Monitor. The lower half of the Request Monitor presents mount and unmount requests to the operator; the upper half of the screen displays the status of all tape drives. Figure 3-1 shows a pending mount request. Request Monitor fields are described in Table 3-1, *Request Monitor Field Descriptions*.

```

+-----+-----+-----+-----+-----+-----+-----+-----+
| Device  Type  Stat VID          UID          Key          Pseudo      |
| drive1  CART  off  -----      root          root          default     |
+-----+-----+-----+-----+-----+-----+
| MID  ACT  ADN  Type  Location/Rack  Pool      |
|>027  MMNT drive1  CART  onsite/998      |
+-----+-----+-----+-----+-----+-----+
| Action: *MNT-DONE  MNT-SKIP  HELP  EXIT      |
+-----+-----+-----+-----+-----+-----+
REELibrarian 3.3 (C)opyright 1993-4 by Storage Technology Corporation

```

Figure 3-1 The Request Monitor

Table 3-1 Request Monitor Field Descriptions

Field	Description
Device	Assigned device name. The name of the tape drive. Drives with a dash (-) appended to their names are inactive or down. Use the DEV-UP command to return them to up status. Likewise, the DEV-DOWN command moves a drive to the down state. Drives with an equal sign (=) appended to their names are on network nodes not currently communicating with the master node. Check the server status on the client node.
Type	The media type supported by the drive.

Table 3-1 Request Monitor Field Descriptions (Continued)

Field	Description
Stat	<p>The drive's status. It can be:</p> <p><code>vr fy</code> - The mounted volume is being electronically identified.</p> <p><code>off</code> - The device is off-line. It does not have a volume mounted on it.</p> <p><code>on</code> - The device is on-line. There is an unidentified volume mounted on the drive.</p> <p><code>dchk</code> - The device is awaiting a density check.</p> <p><code>init</code> - The on-line volume is being initialized by REELlibrarian.</p> <p><code>prem</code> - The mounted volume has been identified and is in waiting for a user request. Premounted volumes become so either by the Automatic Volume Recognition (AVR) system or as a leftover. The AVR watches drives for unsolicited mounts and attempts to identify any volumes so mounted. A leftover is a volume used in a previous user session which is now finished. Since the drive did not have any succeeding business to force an unmount, the volume is kept on-line in the event it is needed again. The system does not know if the volume is writable or read only.</p> <p><code>wprem</code> - Same as the <code>prem</code> state except that the volume is write enabled.</p> <p><code>rprem</code> - Same as the <code>prem</code> state except that the volume is mounted read only.</p> <p><code>umnt</code> - The User Mount State. A user is accessing the drive.</p> <p><code>rew</code> - The mounted volume is rewinding.</p> <p><code>oct1</code> - The drive is about to enter the <code>off</code> state.</p>
VID	The volume ID mounted on the drive.
UID	The UID of the user controlling the drive.
Key	An identifier for a user session, the key is almost always the same as UID.

Table 3-1 Request Monitor Field Descriptions (Continued)

Field	Description
Pseudo	A pseudonym for the drive created by the user.
MID	A unique identifier for the mount/unmount request.
ACT	One of SMNT (scratch mount), WMNT (mount for writing), RMNT (mount for reading), UMNT (unmount), DCHK (density check), or VADD (volume addition).
ADN	This is the name of the drive on which the designated volume is to be mounted.
Type	The media type of the volume.
Location/Rack	The site the volume can be found and the rack number of the volume.
Pool	The pool to which the volume belongs.

The cursor(>) moves along the list of drives and mount/unmount requests with the keys k to move up one and j to move down one. To move from the upper drive area to the lower request area use the J key. K moves from the lower area to the upper area.

The bottom part of the screen contains a spacebar entry area. The prompt changes depending on whether the cursor is in the upper area or lower area. When the cursor is in the upper area of the screen (the drive status area), the associated spacebar response area provides the options listed in Table 3-2, *Drive Status Options*.

Table 3-2 Drive Status Options

Option	Description
DEV-UP	Mark the device as up and ready to use. Down devices are unavailable for use and their ADN is suffixed with a - symbol.

Table 3-2 Drive Status Options (Continued)

Option	Description
DEV-DOWN	Mark the device as down and unavailable to use. If a volume is on-line, an unmount request is generated and entered into the request queue.
DEV-RESET	Takes the device away from the current user and makes it available for general use. If a volume is on-line, an unmount request is generated and entered into the request queue.
HELP	Display help information.
EXIT	Exit the request monitor.

Options are selected by using the spacebar to move the highlight (*) from one item to the next and keying Return on the desired item.

When the cursor is in the lower area of the screen (the mount request area), the associated spacebar response area provides the options listed in Table 3-3, *Mount Request Options*.

Table 3-3 Mount Request Options

Option	Description
MNT-DONE	Confirms that the action as specified by the ACT field has been performed. If it is a mount, a pop-up window then requests the volume ID. Entry may or may not be mandatory depending on whether the original mount request had an asterisk(*) as a suffix to the MID. The volume ID is used to verify the mount.
MNT-SKIP	Indicates that the ACT is being skipped. A pop-up window then prompts for the reason, as shown in Figure 3-2.
HELP	Displays help information.
EXIT	Exits the request monitor.

Table 3-3 Mount Request Options (Continued)

Option	Description
FORCE	<p>This response is entered by keying f. It should only be used if the MNT-DONE response does not succeed because the volume does not pass identification. If you are sure that the proper volume is mounted, this response overrides the identification error. You must enter the volume ID when prompted.</p> <p>Warning: This is a powerful tool and should be used carefully. Forcing a mount inappropriately can corrupt data and catalog integrity.</p>
NEW	<p>This response is entered by keying n. The NEW response should only be used if the FORCE response did not work. It totally bypasses the identification error. You must enter the volume ID when prompted.</p>



If MNT-SKIP is selected, a popup window will appear; this screen is shown in Figure 3-2. Use the spacebar to select an option. If you select Other, you will be required to enter a message in the Message field. This message is sent to the user who originated the request.

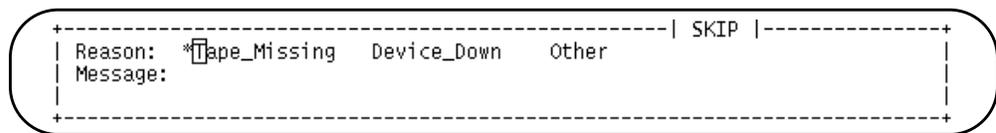


Figure 3-2 MNT-SKIP Popup Window

ACT: Action Requests The ACT field specifies an action to perform. Each of the actions is explained below.

Tape Identity Problems

If the request monitor responds that the mounted tape's electronic identity does not match the requested tape's identity, there are three options.

- Locate the correct tape and mount it.
- If you are certain the correct tape is mounted, use the FORCE option. Follow the description presented earlier.
- If the FORCE option fails, use the NEW option. Follow the description presented earlier.

SMNT: Mount a Scratch Tape

To service a SMNT request, follow these steps.

1. Fetch the tape identified by the rack number displayed on the request line. If the mount cannot be performed, then use the MNT-SKIP option described above.
2. Mount the tape on the tape drive displayed on the request line, with write-protection disabled.
3. Indicate the mount is complete by selecting the MNT-DONE item with the spacebar and keying Return. If the volume ID is requested, obtain it and enter it.

WMNT: Mount for Write Request

To service a WMNT request, follow these steps.

1. Fetch the tape identified by the rack number displayed on the request line. If the mount cannot be performed, then use the MNT-SKIP option described above.
2. Mount the tape on the tape drive displayed on the request line, with write-protection disabled.
3. Indicate the mount is complete by selecting the MNT-DONE item with the spacebar and keying Return. If the volume ID is requested, obtain it and enter it.

RMNT: Mount for Read Request

To service a RMNT request, follow these steps.

1. Fetch the tape identified by the rack number displayed on the request line. If the mount cannot be performed, then use the MNT-SKIP option described above.
 2. Mount the tape on the tape drive displayed on the request line, with write-protection enabled.
 3. Indicate the mount is complete by selecting the MNT-DONE item with the spacebar and keying Return. If the volume ID is requested, obtain it and enter it.
- UMNT: Unmount Request
1. Unmount the volume and return it to its storage slot.
 2. Confirm with the MNT-DONE option.
- DCHK: Density Check Request
- A density check request requires that the density selection on the specified drive be changed. This is usually accomplished via a switch or button on the front panel of the drive.
1. Set the recording format of the tape drive to the support the specified media type (the value under the Type column).
 2. Indicate completion with the MNT-DONE option.
- VADD: Add New Volume
- The VADD request requires that a new tape - a tape not currently known to REELlibrarian - be mounted. REELlibrarian uses VADD requests to dynamically add scratch tapes to pools lacking them so as to satisfy a current mount request needing a scratch tape.
1. Mount a new tape matching the specified media type on the named drive. If a new tape is not available, use MNT-SKIP to reject the request.
 2. When prompted, enter a rack number and volume ID for the tape. Be sure to attach physical labels with these names to the tape when it is unmounted.
 3. Confirm the mount with MNT-DONE.

**Command Line
Operation: rlop**

The command line request monitor is activated by the `rlop` command.

rlop

The `rls` command reports the contents of the request queue. A sample report is shown below.

rls

Request Q: Empty

Device Status:

ADN	Type	Stat	VID	UID Key	Psd
---	---	---	---	---	---
drive1	1600	off	-----		

**WMNT: Mount
Request**

A mount request causes the following type of prompt to display on the operator's screen.

MID	ACT	ADN	Type	Location/Rack	Pool
---	---	---	---	---	---
002*	WMNT	drive1	1600	onsite/101	user/private

The mount request fields are described in Table 3-1, *Request Monitor Field Descriptions*. The operator may either confirm the mount request (`rldone`) or skip the mount request (`rlskip`).

To confirm the mount, follow these steps.

1. Fetch the tape identified by the rack number displayed on the request line. If the mount cannot be performed, then use the `rlskip` command described further below.
2. Mount the tape on the tape drive displayed on the request line, with write-protection disabled.
3. Indicate the mount is complete with the `rldone` command.

rldone mid=*mid* vid

where *vid* is the volume ID found on the volume. (The `mid` keyword can be left out of the command if you are responding to the topmost mount request.)

Tape Identity Problems

If, after confirming the mount, the system responds that the tape has not been properly identified, you should check to make sure that the requested tape was mounted. If you are sure the right tape is mounted, you can use the force option.

```
rldone force=y vid
```

The force= keyword asks that the system ignore the identification error by not requiring the observed fingerprint to coincide with the recorded fingerprint.



Warning: This is a powerful tool and should be used carefully. Forcing a mount inappropriately can corrupt data and catalog integrity.

If the force= keyword fails, then you can use the new= keyword.

```
rldone new=y vid
```

The new= keyword entirely bypasses the volume identification phase.

Skipping the Mount Request

If, for whatever reason, the mount request cannot be satisfied, you should use the rlskip command. The command allows a message to be specified. That explanation is relayed to the user.

```
rlskip mid=mid unable to find volume
```

RMNT: Mount for Read Request

Service the RMNT request same as the WMNT request except that the write protection must be enabled.

UMNT: Unmount Request

An unmount request causes the following message to display on the screen.

```
MID  ACT  ADN      Type   Location/Rack  Pool
---  ---  -----  -----  -----  -----
002* UMNT drive1  1600   onsite/101     user/private
```

1. Unmount the volume and return it to its storage slot.
2. Confirm with the rldone command.

DCHK: Density Check Request

A density check request appears in the form:

```
MID  ACT  ADN      Type   Location/Rack Pool
---  ---  -----  -----  -----
002* DCHK  drive1  1600
```

1. Set the recording format of the tape drive to the specified media type (the value under the Type column). This is usually accomplished via a switch or button on the front panel of the drive.
2. Indicate completion with the `rldone` command.

VADD: Volume Add Request

A VADD request appears in the form:

```
MID  ACT  ADN      Type   Location/Rack Pool
---  ---  -----  -----  -----
002* VADD  drive1  1600   onsite/(2400) user/private
```

1. Mount a new tape matching the specified media type on the named drive. If a new tape is not available, use `rlskip` to reject the request.
2. When prompted, enter a rack number and volume ID for the tape. Be sure to attach physical labels with these names to the tape when it is unmounted.
3. Confirm the mount with `rldone`.

Working with Autoloader Tape Devices

REELlibrarian can be configured to use automatic cartridge loader devices which sequentially mount stacks of tapes.



Note: Tapes must be fingerprinted before use with autoloaders. See *Manually Identifying (Fingerprinting) a Volume* on page 128 for more information.

Stacker Messages

Operator messages regarding autoloader devices that may appear on your monitor are listed below. Recommended actions for each message are also listed.

Problem with site_exit config...	<p>Problem with site_exit config. Manual loading only.</p> <p>This message indicates that the stkmount(8) script is having a problem with the rltapevol command. After satisfying the mount requests manually, check the site_exit logs in the /REEL directory.</p>
Problem with site_exit device...	<p>Problem with site_exit device. Manual loading only.</p> <p>This message indicates that the stkmount script received an invalid device name. After satisfying the mount requests manually:</p> <ol style="list-style-type: none"> 1. Check the configuration of the tape drives. Make sure all defined drives exist and that they are defined on the proper machine.
Can't access tape...	<p>Can't access tape, stacker may be empty. Please reload.</p> <p>This message indicates that the tape drive is not ready. It will be resent to the screen ten times, once every 60 seconds, until a tape is detected or the job is cancelled. The stacker may be empty, or the drive may be disconnected, unplugged, or turned off. If you receive this message:</p> <ol style="list-style-type: none"> 1. Check the stacker. If it is empty, reload it. 2. If it is not empty, check for a loose connection or loss of power. 3. Test if drive can be accessed without using REELlibrarian.
Unknown error encountered...	<p>Unknown error encountered. Manual mode only.</p> <p>This message should not be received. If you receive this message, contact customer support.</p>

Handling the Last Tape in the Magazine

When the last tape in the magazine has been used and unmounted by the system, and there are additional tapes remaining in the stack, then the magazine needs to be replenished with the next tapes in the stack.

1. REELlibrarian signals the next magazine state with a mount request in the monitor. It appears as a standard mount request, but is automatically serviced and confirmed.
2. Remove the tapes currently in the magazine and replace them with the next n in the stack where n is the capacity of the magazine. If there are less than n tapes left, then mount the remaining tapes.



Note: Do not confirm the mount request; this is handled automatically by REELlibrarian.

Tape Management Duties

The Tape Catalog

The tape catalog maintains records for each tape in the library. A tape record includes the contents, status and location of each tape. Information that is tracked in the catalog is listed Table 3-4, : *Cataloged Tape Data*.

Table 3-4 : Cataloged Tape Data

Catalog Item	Description
rack number	Each tape has a unique rack number. It can be the same as either the volume ID and/or the VSN. The rack number represents the physical shelf location of the tape and is used by REELlibrarian when prompting the operator to handle the tape. It can be up to twelve characters long.

Table 3-4 : Cataloged Tape Data

Catalog Item	Description
volume ID	Each tape has a unique volume ID. It can be the same as either the rack number and/or the VSN. The volume ID is an external name for the tape used to confirm the identity of the rack number. Up to twelve characters long.
VSN	The volume serial number (VSN) represents the internal label field recording on the tape. It can be the same as either the rack number and/or the volume ID. Six characters long.
media type	Media types are used to match tapes with tape drives. Usually this item is of no concern unless it is entered wrong so that the tape does not match the media type value of its supporting drive.
status	See the next section on the tape life cycle.
location	The location is the vault where a volume resides. The catalog tracks the location of each tape.

Handling the Tape Life Cycle

REELlibrarian categorizes tape situations into a set of states. The highest level of state for a tape is its state with respect to physical possession. Physical tape states are described Table 3-5, *Physical Tape States*.

Table 3-5 Physical Tape States

Tape State	Description
SUB	Submitted but not yet accepted. Occurs after the user enters <code>rlvsubmit</code> or <code>rlpsubmit</code> but before the operator enters <code>rlaccept</code> .

Table 3-5 Physical Tape States

Tape State	Description
ACC	Accepted but not yet identified (only occurs at sites where <code>rlid</code> is required). Occurs after the operator enters <code>rlaccept</code> but before the operator enters <code>rlid</code> .
Accepted/Identified	It is in the library and available for use.
RET	Retrieved and awaiting return. Occurs after user enters <code>rlvretrieve</code> but before operator enters <code>rlreturn</code> .

The transitions of tapes between these states is known as the tape life cycle. The tape life cycle is shown in Figure 3-3.

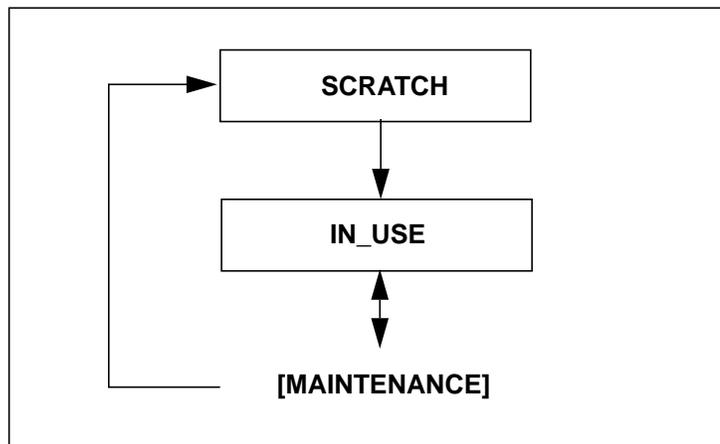


Figure 3-3 The Tape Life Cycle

The tape starts as scratch, it gets drafted into use through either volumeset creation (`rlvscreate`) or volumeset expansion (`rlvwrite`), the volumeset then expires and is scratched (`rlvsscratch`). When scratched, the volume is tested to see if it needs maintenance. If maintenance is required, the tape waits for maintenance before becoming an eligible scratch tape. In addition, it is possible for a tape to have more than one type of maintenance

scheduled (e.g., erasure and cleaning) in which case the each maintenance activity must be completed before the volume is eligible as a scratch volume.

REELlibrarian tape states are listed in Table 3-6, *Tape States*.

Table 3-6 *Tape States*

Tape State	Description
age	If the pool containing the volume prescribes an aging interval, then the volume is held in limbo for the prescribed amount of time.
ctf	Tape requires certification with the <code>rlcertify</code> command to change the tape's status to LIB.
cln	Tape exceeds the cleaning count and requires cleaning and the <code>rlcleaned</code> command to change the tape's status to LIB.
eras	The tape stays in this state until it is run with <code>rlerased</code> .
rem	The tape exceeds either the removal mount count or error limit. The tape stays in this state until it is run with <code>rlremoved</code> (at which time it is removed from the library).
lost	The volume was skipped by the operator during a mount request. The tape stays in this state until the operator runs <code>rlfound</code> with the tape's volume ID. Lost tapes are not allowed to be mounted. This tape state can affect volumes that are either SCRATCH or IN_USE.
move	The volume's current location is different than its scheduled location. The volume stays in this state until the operator runs <code>rlmoved</code> with the tape's volume ID. This tape state can affect volumes that are either SCRATCH or IN_USE.
scr	Available scratch volume. This state indicates that the volume is in the library and has no scheduled maintenance.

Table 3-6 Tape States (Continued)

Tape State	Description
init	Allocated volume scheduled for initialization. The volume will be initialized the next time it is mounted or run with <code>rlid</code> . This state indicates that the volume is in the library and has no scheduled maintenance.
LIB	Allocated volume. This state indicates that the volume is in the library and has no scheduled maintenance.

The `rlr vlist` command generates a report listing all tapes and their states.



Note: When issued by a non-root user, this report lists volumes owned by that user only.

For more information on this or other reports, refer to Chapter 5, *Using Reports*.

Entering Tapes into the Library

User volumes enter the library through a two-step tape submission procedure. First, the user executes either the `rlvsubmit` command or the `rlpsubmit` and identifies the volume and its attributes (or uses the `rl` full-screen program). The program generates a volume ID by which the user identifies the volume in all subsequent REELlibrarian activities. The user then takes the volume and delivers it to the operator relating the volume name also. At that point, the operator performs the second part of the submission procedure.

Full-Screen Operation

To enter a tape into the library, follow these steps.

1. Run the `rlt1m` program; type:

rlt1m

The Library Management menu will appear on screen. It is shown Figure 3-4.

```
-----| LIBRARY MANAGEMENT |-----
SELECTION: 

0  Exit
1  Accept Volume from User
2  Edit Volume Entry
3  Return Volume to User
4  Confirm Volume Movement
5  Confirm Volume Cleaning
6  Confirm Volume Removal
7  Reports

F1 or CNTL-A to Escape          F2 or CNTL-B for Help
-----+-----
REELlibrarian 3.3 (C)opyright 1993-4 by Storage Technology Corporation
```

Figure 3-4 Library Management Menu

2. Select item 1 by pressing 1 followed by Return.
3. A pop-up window prompts for the volume ID; enter the volume ID and press Return.



Note: You should have received the volume ID from the user who submitted the tape. Tapes requiring acceptance are also listed in the maintenance report; for a complete list of tapes requiring acceptance, type:

```
rlr action=accept maint
```

Refer to Chapter 5, *Using Reports*, for a complete description of REELlibrarian reports.

4. Another pop-up window prompts for more information on the tape; this window is shown in Figure 3-5. Enter the appropriate information in each field and press Return. Volume acceptance fields are described in Table 3-7, *Volume Acceptance Fields*.

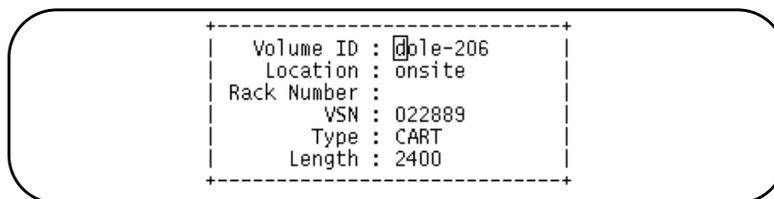


Figure 3-5 Volume Acceptance Pop-up Window

Table 3-7 Volume Acceptance Fields

Field	Description
Volume ID:	The user's name for the tape, up to twelve characters long. This ID should be physically labeled on the tape. The special value <code>vms</code> causes this value to be assigned by the Vault Management System.
Location:	The site where the tape is currently located.
Rack Number:	The volume's rack number, up to twelve characters long. This number will be used in all future references and should represent the storage slot for the tape. The special value <code>vms</code> causes this value to be assigned by the Vault Management System.
VSN:	The volume serial number (VSN), up to six characters long. The special value <code>agen</code> causes REELibrarian to generate a VSN automatically; the special value <code>vms</code> causes this value to be assigned by the Vault Management System.
Type:	The media type of the tape. Media types vary according to each site's configuration.
Length:	If the volume is a reel tape, enter its length in feet. For any other kind of media, enter its capacity in kilobytes (1024).

5. Exit `r1t1m` by pressing 1 followed by Return.
6. If your site requires electronic ID at entry, run the `r1id` program. If it is not required then proceed to step 9.

```
rlid adn=drive vid
```

Where *drive* is a tape drive name and *vid* is the volume ID of the tape. You can name multiple volume IDs and `rlid` handles them sequentially.

The following message should appear on screen:

```
Reserving Device...  
Reservation Complete  
  
Volume 'vid' - mount with type=type hit  
return:
```

7. Mount the tape on the drive and press Return.

The following message should appear on screen:

```
Volume Identification Complete
```

8. Unmount the tape.
9. Store the tape in its rack location.

Command Line Operation

1. Run the command:

```
rlaccept rack=rackno vid
```

vid is supplied by the user when the volume is given to the operator. *rackno* is the rack number for the volume.

2. If your site requires electronic ID at entry, run the `rlid` program. If it is not required, then proceed to step 5.

```
rlid adn=drive vid
```

Where *drive* is a tape drive name and *vid* is the volume ID of the tape. You can name multiple volume IDs and `rlid` handles them sequentially.

The following message should appear on screen:

```
Reserving Device...  
Reservation Complete
```

```
Volume 'vid' - mount with type=type hit  
return:
```

3. Mount the tape on the drive and press Return.

The following message should appear on screen:

```
Volume Identification Complete
```

4. Unmount the tape.
5. Store the tape in its rack location.

Exiting Tapes from the Library

The volume exit procedure permits a user to remove a volume from the library. It is a two-step procedure that begins with the user invoking the `rlvretrieve` command (or the `rl` full-screen program).

Full-Screen Operation

1. To return a tape to its owner select item 3 on the Library Management menu.
2. A pop-window prompts for the receipt number. Enter the number (the owner should have provided this number to you).
3. The rack number and location are displayed. Fetch the tape and input the volume ID.
4. Present the tape to the user.

Command Line Operation

The `rlreturn` command plays the same role as the Return Volume screen in `rlt1m`.

1. Run the `rlreturn` command:

```
rlreturn receiptno
```

Where *receiptno* is the 7-position receipt number supplied by the user.

A message displaying the vault and rack number will appear on screen, as shown below:

```
Location - onsite:rackno
```

2. Fetch the tape from the specified vault and rack.
3. Issue another `rlreturn` command, this time specifying the tape's volume ID:

```
rlreturn vid=VID receiptno
```

`rlreturn` then verifies that the correct volume has been selected and removes it from the catalog. The following message will appear on screen:

```
Database Record Deleted
Return tape to user...
```

4. Present the tape to the user.

Editing a Tape Catalog Entry

Full-Screen Operation

1. Select item 2 on the Library Management menu.
2. At the pop-up window prompt, enter the volume ID.
3. Revise the catalog entry items. The items are described in the earlier section on entering tapes into the library.

Command Line Operation

The command `rledit` modifies the catalog entry for individual volumes.



Note: A complete list of the catalog fields that can be edited can be found in the `rledit(1)` manpage in Appendix A.

The catalog entry for an individual volume can be displayed with the `rlr` command; a sample command is shown below. Sample output is shown in Figure 3-6.

```
rlr vid=bind-423 vinfo
```

Volume Information Report: Wed Sep 21 16:20:00 1994

```
vname: root/filevol:G0000:V00:N001

  vid: bind-423          type: CART          uname: root
  vsn: 000890           ctype: CART         gname: daemon
  rack: 567             length: 2400        vmode: 700
  vsid: bind-423       format: ANSI        passwd:
  vno: 1                ftrack: yes        pool: private
valloc: yes            rformat: u:10240:10240 vexpire: S
  cloc: onsite          conv: data          vacc: ' '
  sloc: onsite          scratch: no         offset: 0
  floc: onsite          maint: 0           ucnt: 1
dispose:              status: LIB         ccnt: 1
  init: no              ftemp: @F17@      app:
ctime: Wed Sep 21 14:31:30 1994
mtime: Wed Sep 21 14:34:19 1994
atime: Wed Sep 21 14:34:19 1994
fingerp: 'A000890CH>j&gcP2smSF70>D!{s'
vcom:
Command: rlr vid=bind-423 full=yes vinfo
```

Figure 3-6 Volume Information Report

The following command changes the volume's rack number.

```
rledit rack=303 bind-423
```

Editing changes may be confirmed by running the `vinfo` report again.



Note: For more information on REELlibrarian reports, refer to Chapter 5, *Using Reports*.

Tape Movement, Cleaning, & Removal

REELlibrarian supports standard tape management practices including movement of volumes to off-site storage, regular, interval based cleaning and end-of-life volume disposal. The following sections describe the operator's part in these activities.

The `rlr` command produces a maintenance report which lists all currently scheduled tape maintenance activities. See the report section later in this chapter.

- | | |
|--------------------|--|
| Moving Tapes | <ol style="list-style-type: none"> 1. Select item 4 on the Library Management menu. 2. A pop-up window appears. Enter the volume ID of the tape, the location to which it has been moved, and the new rack number. |
| Cleaning Tapes | <ol style="list-style-type: none"> 1. Select item 5 on the Library Management menu. 2. A pop-up window appears. Enter the volume ID which has been cleaned. |
| Disposing of Tapes | <ol style="list-style-type: none"> 1. Select item 6 on the Library Management menu. 2. A pop-up window appears. Enter the volume ID which has been disposed of. |

Command Line Operation

- | | |
|--------------------|---|
| Moving Tapes | <p>Confirm the moves via the <code>rlmoved</code> command. For example:</p> <pre style="margin-left: 40px;"><code>rlmoved location=offsite pick-638</code></pre> <pre style="margin-left: 40px;"><code>pick-638: Moved</code></pre> <p>confirms that the volume with volume ID <code>pick-638</code> has been moved to site <code>offsite</code>.</p> |
| Cleaning Tapes | <p>After cleaning the volumes, use the <code>rlcleaned</code> command to confirm the actions.</p> <pre style="margin-left: 40px;"><code>rlcleaned pick-638</code></pre> <pre style="margin-left: 40px;"><code>pick-638: Cleaned</code></pre> |
| Disposing of Tapes | <p>The <code>rlremoved</code> command confirms that a volume have been removed from the library.</p> <pre style="margin-left: 40px;"><code>rlremoved pick-638</code></pre> <pre style="margin-left: 40px;"><code>pick-638: Removed</code></pre> |

Reports Operator reports are available through selection 7, Reports, of the Library Management menu. These reports are described in detail in Chapter 5, *Using Reports*.

Command Summary

The table below summarizes the operator programs.

Table 3-8 REELlibrarian Operator Commands

Command	Description
rl	Master program provides full-screen interface gateway to the request monitor and all other full-screen operator programs.
rlmon	The full-screen request monitor program. Here, the operator receives mount instructions and confirms their execution. Here also, the operator controls system tape drives - taking them "up" or "down" as needed.
rlt1m	The full-screen library management program includes volume acceptance, return, movement, cleaning and removal functions.
RL RLexit RLtest	Start, stop, and test the RL server program.
RLnet RLnexit RLntest	Start, stop, and test the RLnet server program.
rlop	Maintains the active operator roster.
rldone	Confirms tape mounts/unmounts.
rlskip	Cancels tape mounts.
rls	Reports the current request queue and drive status.

Table 3-8 REELlibrarian Operator Commands (Continued)

Command	Description
rldev	Enable/disable tape drives.
rlpremount	Indicate an unsolicited tape mount.
rlunpremount	Indicates an unsolicited tape unmount.
rlaccept rlid	Commands to process a user volume submission.
rlreturn	Command to process a user volume retrieval.
rlr	Reports volumes scheduled for movement, cleaning, and removal.
rlmoved rlcleaned rlremoved rlerased	These programs confirm volume movement, cleaning, removal, and erasure tasks.

Chapter 4. Using REELlibrarian

Overview

This chapter is for the REELlibrarian user. It describes how to store and retrieve data on tapes.

REELlibrarian is a networked tape management system which controls a centrally stored tape library. It allows you to store data on tapes conveniently and securely. All physical tape activities are handled by the tape operator - the person who fetches tapes from the library and mounts them on the system's tape drives.

REELlibrarian users can:

- submit tapes to the library
- retrieve tapes from the library
- create tape pools
- organize files on volumesets
- request tape mounts and unmounts
- write or read tape files
- review volumesets and file contents

The Library REELlibrarian controls tape volumes by placing them in a central library. The library is much like a book library with tape volumes arranged for easy access and a catalog for locating individual volumes.

Entry to and exit from the library is controlled through volume submission and volume retrieval. After your tape has been submitted to the library, you can engage in tape sessions to read and/or write data to your tapes. All tape mounts and unmounts are physically performed by the tape operator according to your requests.

Volumesets You store data on tapes by creating and writing files to volumesets. A volumeset is a group of one or more volumes that constitute one logical volume. When data is written to the first volume of the volumeset and it reaches the volume's capacity limit, REELlibrarian automatically and transparently unmounts the first volume and mounts the second volume so the data can continue to be written.

When a volumeset is created, you specify the pool from which REELlibrarian is to draft volumes into membership of the volumeset when needed. By default this is your private user pool (named `private`).

Volumeset Name A volumeset is identified to REELlibrarian by its volumeset name. This name must be unique to the volumesets a member owns. The full specification of a volumeset name is `[userid/]vname[:Ggno][:Vvno][:Snumber]`. If `userid` is omitted, your `userid` is assumed.



Note: Only operators may submit to a `userid` other than the current effective `userid`.

`vname` is an arbitrary string up to 12 characters long. `Ggno` specifies a volumeset generation number. `Vvno` specifies a volumeset version number. Generation and version numbers are used as subscripts for volumesets with the same `vname`. Refer to the `rlvsubmit(1)` manpage for a full description of generation and version numbers. `Snumber` specifies the section number or the tape number of a volume in a volumeset (index starting with 1).

Unnamed volumesets cannot be created via the full-screen interface.

The Catalog The on-line tape catalog tracks the vital statistics for all volumes in the library. The REELlibrarian catalog can be accessed via both the full-screen and command line interfaces; these interfaces are described in more detail in the next section, *User Interfaces*.

Data the catalog tracks includes:

- volumes that comprise a volumeset
- files on the volumeset
- volume ID
- volume location
- security and ownership information
- volume maintenance records
- volume status

Tape Sessions You store and retrieve data on volumesets by conducting tape sessions at the command line. During a tape session, a volumeset is made available on a tape drive so that you can write or read tape files on it. The tape session commands can:

- have a volumeset mounted on a tape drive
- write one or more files to the volumeset
- read any file from the volumeset
- conclude a session with a volumeset

REELlibrarian employs security checks to let only the permitted users access a volumeset. It also controls the system tape drives, granting users temporary access to tape drives for the volumeset sessions. REELlibrarian allocates the tape drives so that only one user has access to any one tape drive at any point in time.

Pools A pool is a collection of volumes which have a common owner and are constrained by the same access restrictions. All volumes and volumesets in the REELlibrarian library must belong to one and only one pool. You may use tapes from public pools

established for your site, or you may use tapes in your own default private pool (named `private`). You also have the option of creating other private pools and modifying these pools to meet your needs.

You can create new pools and move volumes between the pools. Each pool is defined by its member volumes and its user and group access lists. That is, you can define which users and groups are allowed to use the volumes in your pool if you own that pool. However, a user cannot submit or remove volumes from a pool unless they own it. The user can also specify what other users may access particular volumes within a pool.



Note: The full specification of pool is `userid/pool`. If `userid` is omitted, the current effective user ID is assumed. Only operators may create pools for IDs other than their own.

The system administrator might create a public pool accessible to all users. Individual departments within an organization might create their own pools accessible only to their members.

For example, the system administrator can create a public pool by entering the following commands:

```
rlpcreate uacc=ANY gacc=ANY rllib/oz  
  
rlpsubmit type=XBYTE length=2000  
pool=rllib/oz
```

Users can then access the public pool by entering the following command:

```
rlvcreate pool=rllib/public type=XBYTE \  
length=2000 toto
```

User Interfaces

REELlibrarian provides two user interfaces. The full-screen interface simplifies catalog and maintenance tasks such as volumeset submission, pool creation, and report requests. All REELlibrarian user activities can be accomplished via commands

issued at the command line, including tasks which can also be accomplished via the full-screen interface.

The Full-screen Interface

The REELlibrarian full-screen interface is a menu-driven gateway to the REELlibrarian catalog. It is invoked via the `r1` command.

To bring up the main menu of the full-screen interface, type:

`r1`

and the following screen appears:

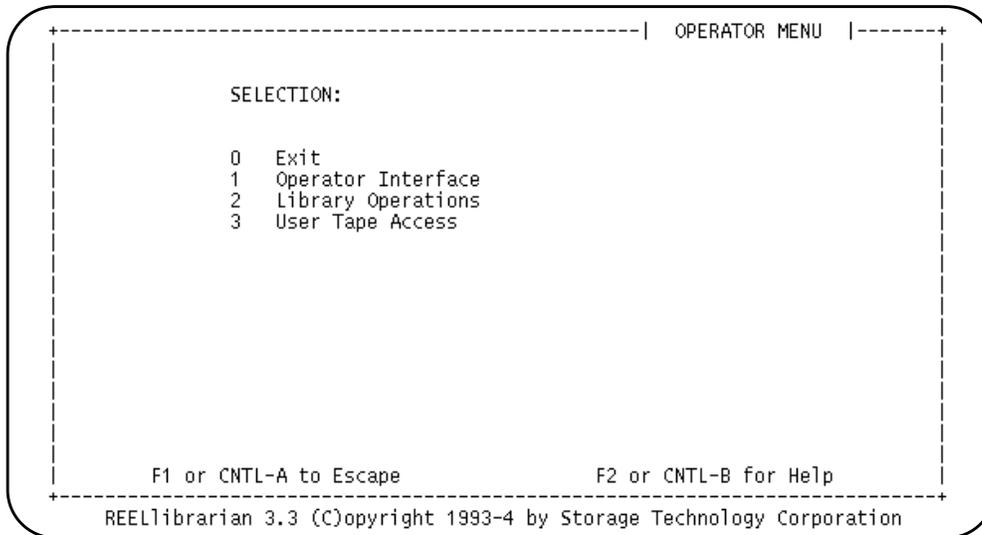


Figure 4-1 Operator Menu

Now press **3**, and the User Tape Access Menu should appear on screen. It is shown in Figure 4-6.

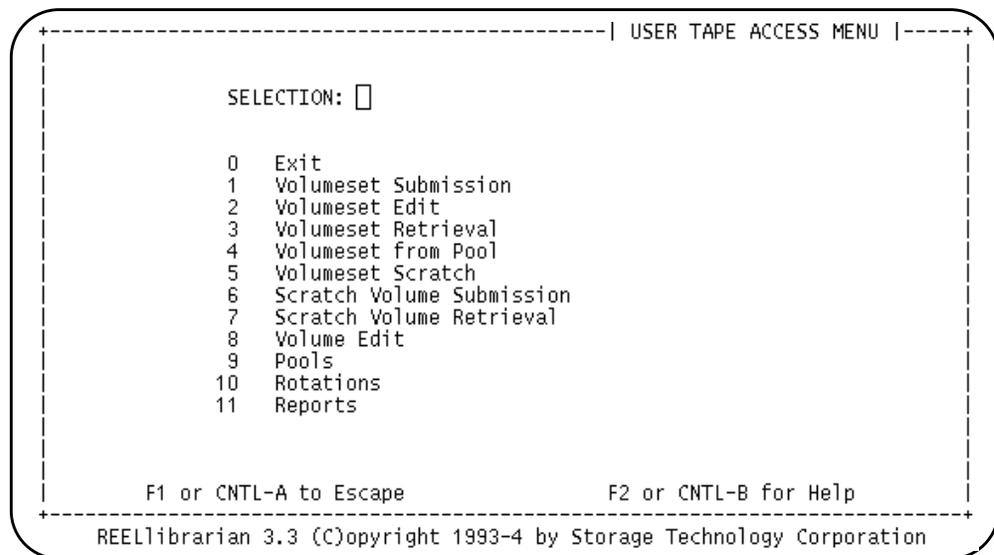


Figure 4-2 User Tape Access Menu

Navigating the Full-screen Interface

REELlibrarian full-screen navigation and input are accomplished by the four methods listed below:

- **Menu Selection** - A list of items is presented; enter the item number of your choice and press [[Return]].
- **Master List** - A small window is displayed with a list of entries. Navigational keys include:
 - a - add
 - e - edit
 - d - delete
 - q - quit the list; return to configuration menu
 - j - move down one item
 - k - move up one item.
- **Field Entry** - Text input is required; type your reply into the field and press [[Return]].
- **Space Bar** - Use the space bar to toggle between choices; the current selection is indicated with reverse video or an asterisk. Press [[Return]] to select.

The following REELlibrarian full-screen functions are accomplished via the key combinations listed below:

- **Escape** - [[F1]] or [[Control-A]]
- **Help** - [[F2]] or [[Control-B]]
- **Process screen** - [[Control-F]]
- **Redraw screen** - [[Control-R]]
- **Previous field** - [[Control-U]]

Submitting a Volumeset



Command Line Equivalent: `rlvsubmit(1)`.



Note: There is a difference between submitting a volumeset and creating a volumeset from scratch volumes already in the pool. For instructions on creating a volumeset from scratch volumes, refer to *Creating a Volumeset from Pool Volumes* on page 124.



Note: You may only submit volumesets to pools that you own, that is, your pool `private` and any other pools you created yourself. For instructions on creating pools, refer to *Creating/Editing/Deleting Pools* on page 131.

A volumeset is a group of one or more physical volumes that constitute one logical volume. Volumesets may contain one or more files. Volumesets can be kept in ANSI or IBM standard label formats, or in IBM, CPIO or TAR unlabeled formats.

REELlibrarian assigns volumes to volumesets as needed and automatically issues mount and unmount instructions to the library operator as required by your tape session. Tape mounts and unmounts during tape sessions should be imperceptible to you; when the end of a physical volume is reached during writing or reading, REELlibrarian instructs the tape operator to unmount the current volume and mount the next volume so that the session can continue.

To enter a volumeset into the library, follow these steps:

1. From the User Tape Access Menu, select Volumeset Submission.
2. Enter **1** and press **[[Return]]**.

The Volumeset Submission Screen should appear. It is shown in Figure 4-7.

```
-----| VOLUMESSET SUBMISSION |-----
  Volumeset Name : 
    Media Type :
    Data Format :
      VSN(s) :
    Length/Capacity :
      Pool :
      Mode :
    Password :
      Expire :
    Location :
    Comment :
    File Tracking :
    File Template :
    Record Format :
    Record Conversion :
      Erase :
      Retain :
    Application :

    F1 or CNTL-A to Escape                F2 or CNTL-B for Help
-----
REELlibrarian 3.3 (C)opyright 1993-4 by Storage Technology Corporation
```

Figure 4-3 Volumeset Submission Screen

3. Enter the attributes of the volumeset, field by field. Volumeset Submission fields are explained in detail in Table 4-1, *Volumeset Submission Fields*.
4. After the volumeset is configured, process the screen by pressing **[[Control-F]]** or by pressing **[[Return]]** from the last field.
5. When the screen is processed, a pop-up window will appear like the one shown in Figure 4-8. This window lists volume serial numbers (VSNs) and volume IDs (VIDs) for each member of the new volumeset.

Table 4-1 Volumeset Submission Fields

Field	Description
Volumeset Name: 	<p>Must be unique among the volumesets you own. The full specification is [<i>userid/</i>] <i>vname</i>[:<i>Ggno</i>][:<i>Vvno</i>]. If <i>userid</i> is omitted, your <i>userid</i> is assumed.</p> <p>Note: Only operators may submit to a <i>userid</i> other than the current effective <i>userid</i>.</p> <p><i>vname</i> is an arbitrary string up to 12 characters long. <i>Ggno</i> specifies a volumeset generation number. <i>Vvno</i> specifies a volumeset version number. Generation and version numbers are used as subscripts for volumesets with the same <i>vname</i>. Refer to the <code>rlvsubmit(1)</code> manpage for a full description of generation and version numbers.</p> <p>Unnamed volumesets cannot be created via the full-screen interface.</p>
Media Type:	<p>The media type of the tapes submitted. Acceptable media types are listed in the <code>rlr type</code> report; refer to the <code>rlr(1)</code> manpage for more information.</p> <p>Example: XBYTE.</p>
Data Format:	<p>Use the [[Spacebar]] to select a label format. IBMU is for IBM unlabeled tapes; RAW is for tapes that do not fit any of the other categories.</p>
VSN(s):	<p>A comma-separated list of serial numbers for each volume in the volumeset. Each tape must be assigned a VSN; a VSN may be up to 6 alphanumeric characters long.</p> <p>The special value <code>agen</code> assigns the volume a unique VSN.</p> <p>Leading slashes (“/”) may be used in place of the comma separators to indicate volumes that belong to the volumeset but do not yet contain data. Refer to the <code>rlvsubmit(1)</code> manpage for more information on specifying VSNs.</p>

Table 4-1 Volumeset Submission Fields (Continued)

Field	Description
Length/Capacity	<p>For reel tapes, enter the length of the tape, in feet. For cartridges, enter the capacity, in megabytes. Most tape reels are 2400 feet in length. QIC-150 tape cartridges have 150 megabytes of capacity. EXABYTE cartridges hold 2200 megabytes of data.</p> <p>Example: 2200</p>
Pool	<p>The pool to submit the volumeset to and to draft scratch volumes from. By default, all volumes are submitted to the user's pool <code>private</code>.</p>
Mode	<p>Security mode for the volumeset, based on UNIX file permissions. Takes the form <code>rwrxwrxrwx</code>, where <code>r</code>=read, <code>w</code>=write, and <code>x</code>=catalog privileges. The first three positions control your privileges; the second three positions control the privileges of the members of your UNIX group (<code>/usr/group</code>); the last three positions control the privileges of all other users. The hyphen (<code>-</code>) in place of any character denies that privilege to the members of that security class. The default string is <code>rwx-----</code>, granting all privileges to the volumeset owner and no privileges to any other user.</p> <p>See the UNIX manpage <code>chmod(1)</code> for more information on permission modes.</p>
Password	<p>Volumeset password (not required); may be up to 12 alphanumeric characters.</p>
Expire	<p>Volumeset expiration date. One of:</p> <ul style="list-style-type: none"> I - never expires S - always expired Rn - expire <i>n</i> days after creation An - expire if not accessed in <i>n</i> days L - expire when all files on the volumeset have expired Xccyyymmdd or Xmm/dd/yy - expire on the given date Gn - expire when there are <i>n</i> newer generations
Location	<p>Storage vault. This sets the scheduled location. This field accepts only defined REELlibrarian storage sites (the command <code>r1r sites</code> lists all currently defined sites). The default is the vault <code>onsite</code>.</p>

Table 4-1 Volumeset Submission Fields (Continued)

Field	Description
Comment	Accepts up to 40 characters (not required). The comment will appear on reports for the volumeset.
File Tracking	Use the [[Spacebar]] to toggle between YES and NO (default). YES causes the catalog to maintain a record for every file written to the volumeset. NO disables file cataloging.
File Template	The file name template is used to dynamically construct names for files written to the volumeset. For further information on specifying a file name template, refer to the <code>rlvsubmit(1)</code> manpage. File templates are not required.
Record Format	Takes the form: <i>fmt:blen:rlen</i> . <i>fmt</i> is the record format, one of: <i>f</i> - fixed-length records; <i>fb</i> - fixed-length, blocked records; <i>v</i> - variable-length records; <i>vb</i> - variable-length, blocked records; <i>vs</i> - variable-length, spanned records; <i>vbs</i> - variable-length, blocked, spanned records; <i>u</i> - unformatted data; <i>blen</i> is the block length in bytes. <i>rlen</i> is the record length in bytes. Example <i>fb:800:80</i> .
Record Conversion	Record conversion specification. Enter one of the following: <i>text</i> - text records <i>etext</i> - EBCDIC text records <i>data</i> - fixed-length ASCII or binary data records (default) <i>edata</i> - fixed-length EBCDIC data records For more information on record conversion, refer to <i>ANSI and IBM Tape Handling</i> on page 143.
Erase	Use the [[Spacebar]] to toggle between YES and NO (default). If set to YES, volumes are erased when they leave the volumeset.
Retain	Use the [[Spacebar]] to toggle between YES and NO (default). If set to YES, the volumeset is never truncated, and all original volumes remain as volumeset members. If set to NO, whenever a file on the volumeset is overwritten, all volumes following the new file are scratched.

Table 4-1 Volumeset Submission Fields (Continued)

Field	Description
Application	Enter the name of an associated application, as you wish it to appear on reports (not required).

Editing a Volumeset



Command Line Equivalent: `rlvedit(1)`.

To edit the catalog record of an existing volumeset, follow these steps:

1. From the User Tape Access menu, select Volumeset Edit.
2. Enter **2** and press `[[Return]]`.

A blank Volumeset Edit screen should appear; note that the fields are very similar to those of the Volumeset Submission screen. The Volumeset Edit screen is shown in Figure 4-9.

```
-----| VOLUMESSET EDIT |-----
Volumeset Name : 
Media Type :
Data Format :
Length/Capacity :
Pool :
Mode :
Password :
Expire :
Location :
Comment :
File Tracking :
File Template :
Record Format :
Record Conversion :
Erase :
Retain :
Initialize :
Application :

F1 or CNTL-A to Escape           F2 or CNTL-B for Help
-----+-----
REELlibrarian 3.3 (C)copyright 1993-4 by Storage Technology Corporation
```

Figure 4-5 Volumeset Edit Screen

3. Enter the name of the volumeset you wish to edit in the Volumeset Name field and press `[[Return]]`; the remaining fields are updated to display the catalog entry for the volumeset.
4. Go to the field(s) you wish to edit by pressing `[[Return]]`.



Note: The Initialize field is a restricted field; only operators may edit this field.

5. Make any changes you wish to make to the volumeset record by over-typing in the appropriate field.
6. Process the screen by pressing `[[Return]]` from the last field or `[[Control-F]]` from any field. A pop-up window similar to the one shown in Figure 4-6 should appear.

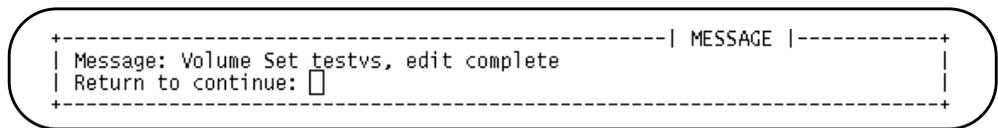


Figure 4-6 Volumeset Edit Pop-up Window

7. Press `[[Return]]` to return to the User Tape Access menu.

Certain Volumeset Edit fields behave differently from their counterparts on the Volumeset Submission menu; these differences are noted in Table 4-2.

Table 4-2 Volumeset Edit Fields (different from Volumeset Submission Fields)

Field	Description
VSN(s):	Once a VSN has been established, it may not be edited; this field does not appear in the Volumeset Edit screen.
Password:	To remove an existing password, enter the special value <code>none</code> in this field.
Initialize:	This is a restricted field; only tape operators may edit this field. <code>YES</code> indicates that the volumeset will be initialized the next time it is mounted.

Retrieving a Volumeset



Command Line Equivalent: `rlvretrieve(1)`.



Note: These instructions are for retrieving volumesets that you own. For instructions on retrieving scratch volumes, refer to *Retrieving a Scratch Volume* on page 129.

To retrieve a volumeset from the tape library, follow these steps:

1. From the User Tape Access menu, select Volumeset Retrieval; type **3** and press `[[Return]]`.

A pop-up window should appear on screen prompting for the name of the volumeset.

2. Type the name of the volumeset you wish to retrieve and press `[[Return]]`.
3. Another pop-up window should appear on screen, like the one shown in Figure 4-11. This window lists Volume IDs and retrieval receipt numbers for all of the volumes in the volumeset. Present these receipt numbers to the library operator to retrieve your volumeset.

```
Retrieving...
+---+-----+
|Vno  VID      Receipt|
+---+-----+
| 1   honk-986  R000218|
+---+-----+
Return to Continue: 
```

Figure 4-7 Volumeset Retrieval Pop-up Window

Creating a Volumeset from Pool Volumes



Command Line Equivalent: `rlvcreate(1)`.

An alternative to submitting scratch volumesets is to create volumesets from volumes that are already on hand in a pool. Volumesets created in this manner are automatically assigned one volume from the specified pool; subsequent volumes are automatically drafted from the same pool and attached to the volumeset as they are needed.



Volumesets cannot be created from pool volumes unless the pool has available, compatible scratch volumes; you may need to submit volumes to the desired pool prior to creating volumesets (a prompt will inform you if no compatible volumes are available). Refer to *Submitting a Scratch Volume* on page 126 for more information.

To create a volumeset from pool volumes, follow these steps:

1. From the User Tape Access menu, select Volumeset from Pool.
2. Enter **4** and press `[[Return]]`.

A blank Volumeset from Pool screen should appear like the one shown in Figure 4-13. Note that the fields are very similar to those of the Volumeset Submission screen.

3. Enter the attributes of the volumeset, field by field. Volumeset Submission fields are explained in detail in Table 4-1, *Volumeset Submission Fields*.
4. After the volumeset is configured, process the screen by pressing `[[Control-F]]` or by pressing `[[Return]]` from the last field.
5. When the screen is processed, a pop-up window will appear like the one shown in Figure 4-13. This pop-up window displays the volume ID of the volume allocated to the new volumeset.

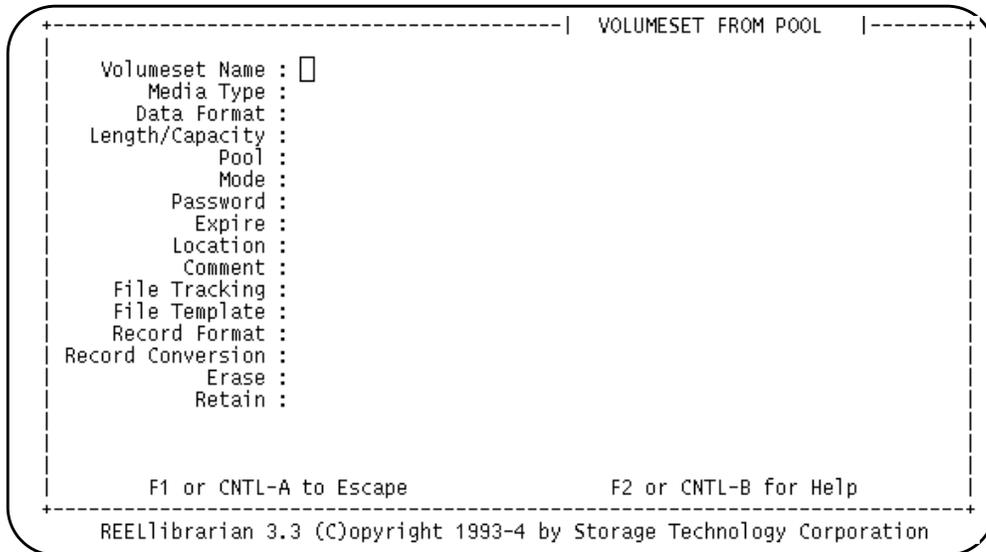


Figure 4-8 Volumeset from Pool Screen

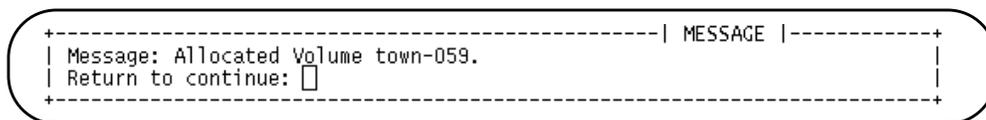


Figure 4-9 Volumeset from Pool Pop-up Window

Deleting a Volumeset



Command Line Equivalent: `rlvscratch(1)`.

Deleting a volumeset means that its member volumes are scratched and the catalog entry for the volumeset is deleted.

To delete a volumeset, follow these steps:

1. From the User Tape Access menu, select Volumeset Scratch.
2. Type **5** and press `[[Return]]`.

A pop-up window should appear on screen like the one shown in Figure 4-14.

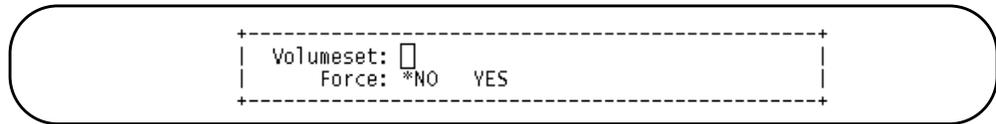


Figure 4-10 Scratch Volumeset Pop-up Window

3. Enter the name of the volumeset you wish to scratch and press `[[Return]]`.
4. If the volumeset has already expired, it is not necessary to force the scratch. Press `[[Return]]` to select `NO` (do not force).
5. To override the system's normal objections to deleting an unexpired volumeset, use the `[[Spacebar]]` to toggle to `YES` (force); press `[[Return]]`.



Note: To determine if a volumeset has expired, refer to the expire field of the Volumeset Attributes report for the volumeset. Refer to Chapter 5, *Using Reports*, for more information.

Submitting a Scratch Volume



Command Line Equivalent: `rlpsubmit(1)`.



Note: You may only submit scratch volumes to pools that you own, that is, your pool `private` and any other pools you created yourself. For instructions on creating pools, refer to *Creating a Pool* on page 131.

To submit a scratch volume, follow these steps:

1. From the User Tape Access menu, select Scratch Volume Submission;
2. Enter **6** and press `[[Return]]`.

A pop-up window should appear on screen like the one shown in Figure 4-15; for field definitions, refer to Table 4-1, *Volumeset Submission Fields*.

```
+--| SCRATCH SUBMIT |-----+
|                               |
|   Pool: private   |
| Media Type: CART              |
|   VSN: agen                  |
| Length: 0                    |
|                               |
+-----+-----+
```

Figure 4-11 Scratch Volume Submission Pop-up Window

- From each field, enter the correct information for the scratch volume and press `[[Return]]`.
- A pop-up window should appear on screen like the one shown in Figure 4-12; this pop-up window displays the new VID.

```
-----| MESSAGE |-----+
| Message: VID 'rift-925', Submission Complete (onsite/) |
| Return to continue:  |
+-----+-----+
```

Figure 4-12 Scratch Volume ID Pop-up Window



Note: You may wish to record the VID for future references to the volume.

- Deliver the volume to the operator for acceptance into the library.

A newly submitted scratch volume cannot be drafted into a volumeset until the operator has accepted it. You may check volume status by requesting a Volume Attributes Report. Volume state `SUB` means that the tape has been submitted, but not accepted. Volume state `SCR` indicates that the volumeset is usable. For more information on reports and tape states, refer to Chapter 5, *Using Reports*.



Note: For submitting large numbers of scratch volumes, it is more practical to incorporate the `rlpsubmit(1)` command into a script.

Manually Identifying (Fingerprinting) a Volume

These instructions are for manually identifying a volume to REELlibrarian, also known as fingerprinting. The fingerprint is a coded identity derived from the data on the volume. This electronic code allows REELlibrarian to recognize volumes in its catalog and prevent data corruption.

Although a fingerprint is automatically generated by REELlibrarian when a volume is first mounted, there is occasionally a need to regenerate a fingerprint in case it has been corrupted or if a fingerprint needs to be generated before the first time the volume is mounted (e.g., use with autoloaders). This is done by using the `r1id` command.



Note: There is no way of using the full-screen interface to fingerprint volumes.

To manually identify or “fingerprint” a volume to REELlibrarian, follow these steps:

1. Mount the volume in a tape device
2. Enter

```
r1id adn=adn vid
```

where *adn* is the Assigned Device Name of the tape device the volume is mounted on, and *vid* is the Volume ID of the tape volume you want to fingerprint.

A message similar to the following prompts you to confirm the identification of the volume.

```
Reserving Device...
```

```
Reservation Complete
```

```
Volume 'tape-101' - mount with type=CART hit  
return:
```

3. Press `[[Return]]` and the following message is returned.

```
Volume Identification Complete
```

Retrieving a Scratch Volume



Command Line Equivalent: `rlretrieve(1)`.



Note: These instructions are for retrieving scratch volumes that you own. For instructions on retrieving active volumesets, refer to *Retrieving a Volumeset* on page 123.

To retrieve a volume and remove its record from the library catalog, follow these steps:

1. From the User Tape Access menu, select Scratch Volume Retrieval; type **7** and press `[[Return]]`.

A pop-up window should appear on screen like the one shown in Figure 4-17.



Figure 4-13 Scratch Volume Retrieval Pop-up Window

2. Type the volume ID in the window and press `[[Return]]`.
3. Your retrieval receipt number should appear on-screen, as shown in Figure 4-18. Take this to the library operator to receive your volume.

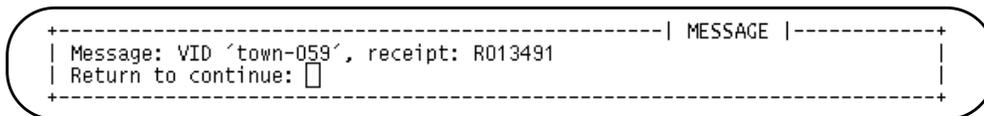


Figure 4-14 Scratch Volume Retrieval Pop-up Window

Editing the Volume Catalog Entry



Command Line Equivalent: `rledit(1)`.



Note: You cannot edit the scheduled volume location and free volume location catalog fields via the full-screen interface; for instructions on modifying these fields at the command line, refer to the `rledit(1)` manpage.

To modify a volume catalog entry, follow these steps:

1. From the User Tape Access menu, select Volume Edit; type **8** and press `[[Return]]`.

A pop-up window should appear on screen like the one shown in Figure 4-19.

```
+-----+
| Volume ID: |
+-----+
```

Figure 4-15 Volume Edit Pop-up Window

2. Type the volume ID in the window and press `[[Return]]`.

A second pop-up window should appear on screen like the one shown in Figure 4-20; for field definitions, refer to Table 4-1, *Volumeset Submission Fields*.

```
+--| VOLUME EDIT |-----+
|
| Pool: private
| Media Type: CART
| VSN: 345678
| Length: 2400
+-----+
```

Figure 4-16 Volume Edit Pop-up Window

3. From each field, enter the new information for the scratch volume and press `[[Return]]`.
4. Another pop-up window should appear on screen like the one shown in Figure 4-21; press `[[Return]]` to exit the window.

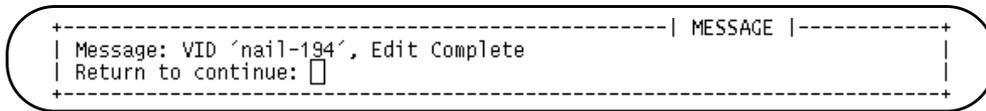


Figure 4-17 Edit Complete Pop-up Window



Never modify the pool value of volumes that belong to volumesets in this manner. Instead, use the Volumeset Edit function, as described in *Editing a Volumeset* on page 121, to modify the pool field. This way all volumeset members will belong to the same pool.

Creating/Editing/Deleting Pools

This section provides instructions for creating, editing, and deleting pools. For more information on pools, refer to *Pools* on page 111 of this chapter.

Creating a Pool



Command Line Equivalent: `rlpcreate(1)`.

To create a new pool, follow these steps:

1. From the User Tape Access Menu, select Pools; type **9** and press `[[Return]]`.

The Pools screen should appear; it is shown in Figure 4-22. Screen fields are defined in Table 4-3, *Pool Creation Fields*.

8. Press **[q]** to move to the Group List.
9. Press **[a]** to add a name to the Group List; type the name in the Enter field and press **[Return]** to add the name to the list. Repeat this step for every group name you wish to enter.
10. When the pool is configured as you would like it, press **[q]** or **[Control-f]** to process the screen.



Note: After you create a tape pool, you probably will want to submit volumes to it. For instructions, refer to *Submitting a Scratch Volume* on page 126.

Table 4-3 Pool Creation Fields

Field	Description
ACTION	Use the [Spacebar] to toggle among the options; press [Return] to select the highlighted option.
Pool Name	Must be unique among all of your pool names; may be up to 12 alphanumeric characters.
Cleaning Regime	Use the [Spacebar] to toggle between the options. YES schedules your volumes for periodic cleaning at an interval determined by the administrator. Volumes are only cleaned when they are in the scratch state. Default: NO.
End-of-Life Disposal	Use the [Spacebar] to toggle between the options. YES schedules volumes for disposal after a number of mounts, determined by the administrator, have been logged for the volume. Volumes are only discarded when they are in the scratch state. Default: NO.
Days before Recycle:	The number of days before a member of a newly scratched volumeset may be drafted into a new volumeset. This grace period protects data on volumesets that you may have scratched by mistake. Default: 0.

Table 4-3 Pool Creation Fields (Continued)

Field	Description
User List	<p>The list of user IDs permitted to access the tape pool. This field accepts the key commands listed in <i>Navigating the Full-screen Interface</i> on page 114.</p> <p>The add and edit functions activate the Enter field at the bottom of the list; type the user ID to include and press <code>[[Return]]</code>.</p> <p>To explicitly prohibit a user from the pool, prefix the user ID with an exclamation mark; this excludes members of groups permitted under the Group List.</p> <p>The special entry ANY grants access all users not explicitly prohibited with exclamation mark-prefixed user or group list entries.</p>
Group List	<p>The list of group IDs (as defined in <code>/etc/group</code>) permitted to access the tape pool. This field functions identically to the User List field, defined above.</p>

Editing a Pool



Command Line Equivalent: `rlpedit(1)`.

To edit a pool, follow these steps:

1. From the User Tape Access Menu, select Pools; type **9** and press `[[Return]]`.

The Pools screen should appear; it is shown in Figure 4-22. Screen fields are defined in Table 4-3, *Pool Creation Fields*.
2. From the ACTION field, press `[[Spacebar]]` to toggle to EDIT; press `[[Return]]` to select.
3. Enter the name of the pool you wish to edit in the Pool Name field; press `[[Return]]`. This populates the fields of the screen with the attributes of that pool.
4. Make any desired modifications to the screen fields according to steps 4 through 10 of *Creating a Pool* on page 131.

Deleting a Pool



Command Line Equivalent: `rlpdelete(1)`.

To delete a pool, follow these steps:

1. From the User Tape Access Menu, select Pools; type **9** and press `[[Return]]`.

The Pools screen should appear; it is shown in Figure 4-22. Screen fields are defined in Table 4-3, *Pool Creation Fields*.

2. From the ACTION field, press `[[Spacebar]]` to toggle to DELETE; press `[[Return]]` to select.
3. Enter the name of the pool you wish to delete in the Pool Name field; press `[[Return]]`.

When deleting a pool, you will not get any prompt.

Creating/Editing/Deleting Rotations

This section provides instructions for creating, editing, and deleting rotations. For more information on rotations, refer to the `rlrcreate(1)` and `rlrdelete(1)` manpages in Appendix A.

Creating a Rotation



Command Line Equivalent: `rlrcreate(1)`.

To create a rotation, follow these steps:

1. From the User Tape Access Menu, select Rotations; type **10** and press `[[Return]]`.

The Rotations screen should appear; it is shown in Figure 4-19. Rotation fields are defined in Table 4-4, *Rotation Fields*.

2. From the ACTION field, press `[[Spacebar]]` to toggle to CREATE; press `[[Return]]` to select.

3. Enter the name of the rotation you wish to create in the Rotation Name field; press [[Return]].
4. The cursor should now be in the empty window in the center of the screen. Press [[a]] to add the rotation definition. A pop-up window will appear on screen; it is shown in Figure 4-24.
5. Enter the first vault location on the rotation list in the Location field and press [[Return]].
6. Enter the duration to associate with the location in the Duration field and press [[Return]].
7. The location and duration pair will now appear in the window in the center of the screen. To add more items to the rotation list, repeat steps 4 through 6. A sample rotation list is shown in Figure 4-25.
8. When the rotation is configured as you would like it, press [[q]] or [[Control-f]] to process the screen.

Table 4-4 Rotation Fields

Field	Description
ACTION:	Use the [[Spacebar]] to toggle among the options; press [[Return]] to select the highlighted option.
Rotation Name:	Must be unique among all of your pool names; may be up to 12 alphanumeric characters.
Location:	Storage Vault. This field accepts only defined REELlibrarian storage sites.
Duration:	The length of time, in either days (R) or generations (G), that volumesets assigned to this rotation will reside at this location before rotating to the next location on the rotation list. Example: R30 (volumeset will remain in the vault for 30 days)

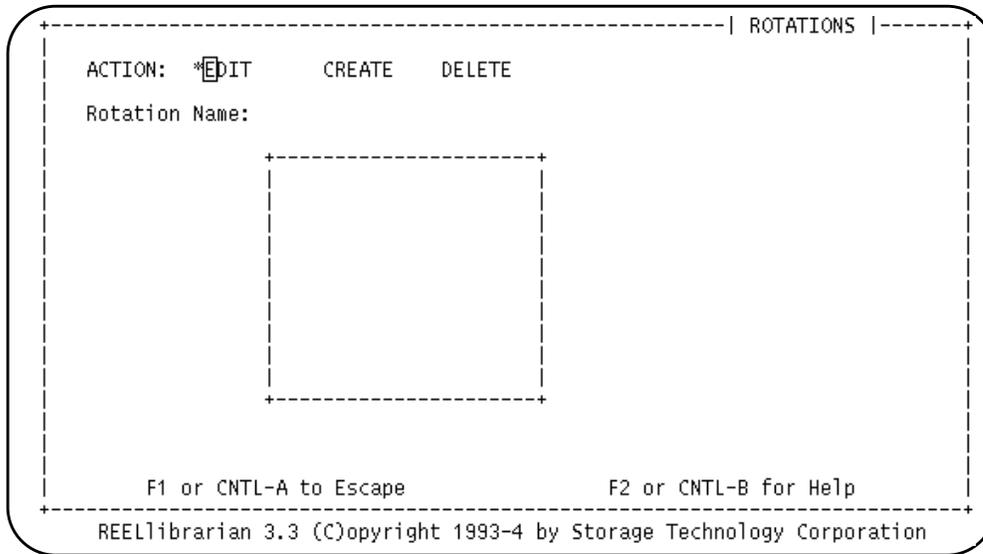


Figure 4-19 Rotations Screen

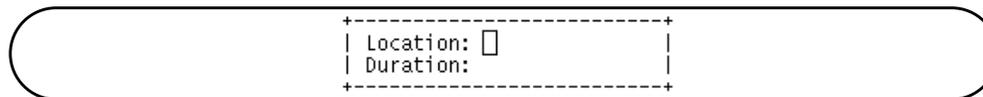


Figure 4-20 Rotations Pop-up Window

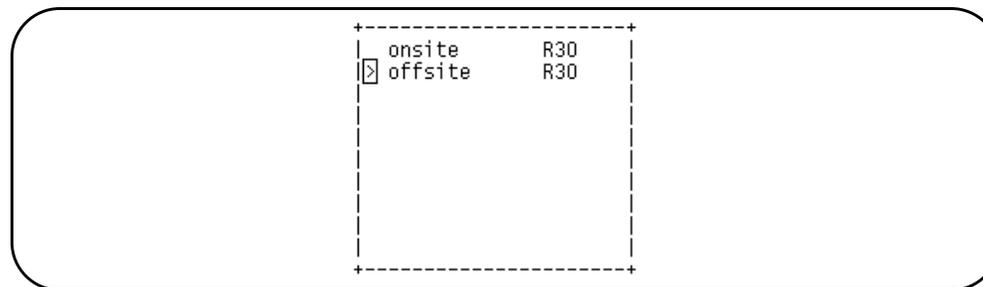


Figure 4-21 Rotation List

Editing a Rotation

To edit a rotation, follow these steps:

1. From the User Tape Access Menu, select Rotations; type **10** and press `[[Return]]`.

The Rotations screen should appear; it is shown in Figure 4-19. Rotation fields are defined in Table 4-4, *Rotation Fields*.

2. From the ACTION field, press `[[Spacebar]]` to toggle to EDIT; press `[[Return]]` to select.
3. Enter the name of the rotation you wish to edit in the Rotation Name field; press `[[Return]]`.
4. The rotation list will now appear in the window in the center of the screen. A sample rotation list is shown in Figure 4-25. Press `[[a]]` to add, `[[e]]` to edit, and `[[d]]` to delete items from the rotation list. If adding or editing items on the rotation list, a pop-up window will appear on screen; it is shown in Figure 4-24.
5. If adding or editing an item from the rotation list, enter the changes in the popup window.
6. When the rotation is configured as you would like it, press `[[q]]` or `[[Control-f]]` to process the screen.

Deleting a Rotation



Command Line Equivalent: `r1rdelete(1)`.

To delete a rotation, follow these steps:

1. From the User Tape Access Menu, select Rotations; type **10** and press `[[Return]]`.

The Rotations screen should appear; it is shown in Figure 4-19. Rotation fields are defined in Table 4-4, *Rotation Fields*.

2. From the ACTION field, press `[[Spacebar]]` to toggle to DELETE; press `[[Return]]` to select.

3. Enter the name of the rotation you wish to delete in the Rotation Name field; press [[Return]].

A Volumeset Session

Volumeset sessions are conducted via the commands listed in Table 4-5, *Volumeset Session Commands*.

Table 4-5 Volumeset Session Commands

list:input:1	Description
<code>rlvaccess</code>	Initiate access to a volumeset.
<code>rlvrelease</code>	Complete access to a volumeset.
<code>rlvwrite</code>	Write a file to a volumeset.
<code>rlvread</code>	Read a file from a volumeset.
<code>rlvdisplay</code>	Display file attributes.

For a complete description of these commands, refer to the manpages in Appendix A.

Storing Files on a Volumeset

The following command creates a volumeset named `forecast`:

```
rlvcreate pool=private forecast
```

To initiate a session where you will store files on the volumeset, you must use the `rlvaccess` command. It verifies that you have access rights to the volumeset and requests an appropriate tape drive from the MRS. A volume in the volumeset is not actually mounted until a read or write command is executed. For volumesets containing more than one volume, this reduces operator activity and delays. An example is shown below.

```
rlvaccess write=y forecast
```

The `rlstatus` command provides a status report which shows the drive reserved by the request. An example is shown below.

```
rlstatus
```

Request Q: Empty
Device Status:

ADN	Type	Stat	VID	UID	Key	Psd
---	----	----	----	----	---	---
DB1	CART	off	-----			
DB2	CART	off	-----			
DV1	1600	off	-----			
DV2	1600	off	-----			
DV3	1600	off	-----	djg	djg	default
drive1	CART	off	-----			

To write the file data onto the volumeset, issue the command:

```
rlvwrite if=data
```

rlvwrite writes at the current position in the volumeset. If you follow the first rlvwrite command with another command writing out the file data2, it will follow the file data. An example is shown below.

```
rlvwrite if=data2
```

Finish the session by releasing the volumeset with the following command:

```
rlvrelease  
Device drive1 Freed
```

Reading Files from a Volumeset

To read in the files written onto the forecast volumeset, use the commands shown below.

```
rlvaccess forecast  
Access complete  
rlvread of=data.new  
rlvread of=data2.new  
rlvrelease  
Device drive1 Freed
```

Table of Contents Report

When volumesets are submitted to the library, the catalog does not have a record of the volumeset's files. So files cannot be referred to by name, nor can any detailed information be reported by the `rlr` report program.

However, the `rlvdisplay` command can be used to perform a scan on the volumeset and so, build a table of contents in the catalog. For example:

```
rlvaccess myimport  
  
rlvdisplay report=scan  
  
rlvrelease
```

Transient Volumesets

A transient volumeset is a volumeset that is in the REELlibrarian catalog but has been updated outside of REELlibrarian (e.g., a different system). Thus, when the volumeset gets mounted again by REELlibrarian it has a corrupted fingerprint. The `rlvtran` command will reinitialize the fingerprint so that REELlibrarian can recognize the volumeset again when it is mounted.

For more information on using the `rlvtran` command, see the `rlvtran(1)` manpage on page 284.

Using the Catalog

The catalog maintains entries for each of your volumesets and for the files on the volumesets (this may be disabled for a volumeset).

Maintaining file catalogs (file tracking set to yes for the volumeset) makes it possible to generate a volumeset table of contents without having to mount a tape. It also improves the efficiency of accessing files on multi-volume volumesets.

Volumeset Format Issues

TAR Tape Handling

There are two ways to use TAR tapes under REELlibrarian. The first way is to use the direct volume access route and use the `tar` command on the tape directly. The second way is to assemble the tape volumes into a TAR format volumeset. By using the

volumeset method, you gain the file tracking capability provided by the catalog.

TAR Tape Handling Qualifications

REELibrarian writes TAR tapes with a block size of 10240 bytes. This means that these tapes can be read by the standard `tar` command. REELibrarian supports tar tapes via the TAR format type with the following exceptions:

- REELibrarian cannot read multiple file, multi-volume tar volumesets that it has not created itself.
- REELibrarian can fail on multi-volume TAR format volumesets. The reason for possible failure is the inability to distinguish when a `tar` file ends from when the end of volume occurs. This is an inherent problem with most UNIX tape drivers.

Creating a TAR Volumeset

The `rlvcreate` command allows you to create a TAR volumeset.

```
rlvcreate format=TAR tarset
```

If you have existing TAR format tapes which you want to use via REELibrarian, submit the individual volumes to the library using the command with the volume names in the correct order that they were written.

Reading TAR Tapes

To read a TAR volumeset, access it and use the command:

```
rlvaccess tarset
```

```
rlvread | tar xf -
```

The `tar` command must use the “`f -`” option as that indicates input is coming from standard input.

If the tape has a block size other than the standard 10240 bytes, set the `rformat` keyword with the read command. For example, this command sets the block size to 20480 bytes.

```
rlvread rformat=:20480 | tar xf -
```

If a multi-volume volumeset is being read, REELlibrarian automatically handles the unmount and mount to successive tape volumes.

Writing TAR Tapes

Writing a TAR tape is accomplished via the `rlvwrite` command:

```
tar cf - /usr/* | rlvwrite
```

Be sure to use the “f -” option as that indicates to `tar` that output should go to standard output.

As with reading TAR tapes, use the `rformat` keyword to make any adjustments in block size.

CPIO Tape Handling

REELlibrarian writes CPIO tapes with a block size of 5120 bytes. This means that these tapes can be read by the standard `cpio` command with the appropriate options: `cpio -iB`. REELlibrarian supports CPIO tapes in the same manner as TAR tapes. Refer to the *TAR Tape Handling Qualifications* on page 142.

RAW Tape Handling

RAW format volumesets are read and written via the `rlvread` and `rlvwrite` commands.

REELlibrarian, by default, uses a block size of 32768 bytes when reading or writing RAW tapes. If a different block size is appropriate, use the `rformat` keyword to set it. This example shows the block size being set to 1024 bytes.

```
rlvwrite rformat=u:1024:1024 < data
```

ANSI and IBM Tape Handling

REELlibrarian supports the ANSI and IBM tape label standards. This section discusses the manipulation of these standards via the REELlibrarian commands.

Volume Labels

REELlibrarian fully supports the volume labels found at the beginning of each IBM and ANSI tape.

When an IBM or ANSI volumeset is accessed, REELlibrarian automatically reads the volume labels to verify tape conformance to the label standard.

Volume labels are automatically written on IBM or ANSI tapes when they are accessed for the first time and their initialize flag is on. The owner field is written with the user's UNIX ID.

Creating an IBM or ANSI Labeled Tape

Use the `rlvcreate` command to create an ANSI or IBM volumeset. As volumes are added to the volumeset, they are given appropriate volume and file labels.

File Labels

Each file on an ANSI or IBM volumeset has file labels that contain descriptive information of the file. File label information is available via the `rlvdisplay` command. An example command and output is shown below.

```
rlvdisplay fseq=1 report=labels

REELlibrarian by StorageTek

File file1 (fseq=1, fsect=1)

    owner   : root
    fid     : file1
    gname   : daemon
    fmode   : 700
    vsid    : bind-423
    vname   : filevol:G0000:V00
    rformat : vbs:4096:512
    conv    : text
    offset  : 0
    passwd  :
    blocks  : 8
    vno     : 1
    fno     : 0
    mask    : 3f 1c
    fexpire : S
    app     :
    ctime   : Wed Sep 21 14:34:12 1994
    atime   : Wed Sep 21 14:34:12 1994
    fcom    :
```

When a volumeset file is written, file labels are written along with it. REELlibrarian keeps a set of default values for the label fields as part of the catalog. For any individual file, the field values can be modified via keywords with the `rlvwrite` command.

IBM File Formats IBM tape files can be stored in a variety of formats. REELlibrarian automatically detects and accommodates a file's format when read from tape. When writing a tape file, the format can be selected in one of two ways. REELlibrarian provides a special keyword for the `rlvwrite` command called `rformat`. IBM file formats are described in Table 4-6, *IBM File Format Descriptions*.

Table 4-6 IBM File Format Descriptions

rformat	Description
f	Fixed-length records, one per block.
fb	Fixed-length records, blocked.
u	Unformatted data.
v	Variable-length records, one per block. Records that exceed the record length are truncated.
vb	Variable-length records, blocked.
vs	Variable-length records, spanned.
vbs	Variable-length records, blocked and spanned.

To set a file to be of format fixed block, use this command:

```
rlvwrite rformat=fb:800:80 < data
```

The `rformat` keyword represents more than the record format. It has in order, separated by colons: the record format, the block length, and the record length.

ANSI File Formats ANSI tape files can be stored in three formats. REELlibrarian automatically detects and accommodates a file's format when read from tape. When writing a tape file, the format is selected via the `rformat` keyword. The ANSI standard supports three record formats: `F` (fixed-length records), `D` (variable-length records), and `S` (variable-length, spanned records). Fixed-length records are automatically blocked together if the file block length is a multiple of the record length. ANSI formats are selected via the

corresponding rformat values listed in Table 4-7, *ANSI File Formats*.

Table 4-7 ANSI File Formats

ANSI format	rformat value
F	f fb
D	v vb
S	vs vbs

To select fixed-length records, use this command:

```
rlvwrite rformat=f:Vblen:Vrlen < data
```

Record Conversion

One of the difficulties of reading or writing IBM and ANSI tapes on a UNIX system is translating between IBM and ANSI records and UNIX data. The UNIX operating system does not support records. On UNIX, the notion of a record only has meaning to individual applications. For example, the program `vi` considers a record to be all the characters it finds between two ASCII newline characters (`\n`). Database applications define records to be arbitrary fixed or variable-length chunks of data.

When REELlibrarian reads or writes a file, it cannot guess the most appropriate way to perform record translation. The `convert` keyword for the `rlvread` and `rlvwrite` commands controls how record translation is performed. Possible values for `convert` are listed in Table 4-8, *Convert Keyword Values*.

Table 4-8 Convert Keyword Values

convert=	Description
text	Assumes ASCII data. If the file is in fixed-length record format, record conversion is performed. The record padding character is assumed to be a blank and the record termination character is the newline character ('\n'). So, when a file is written to tape, short records are padded with blanks. A UNIX "record" for tape writing purposes is assumed to be terminated by the newline character ('\n'). When reading a tape file, trailing blanks are stripped from the record and a newline is attached to the end of the record.
etext	The same as the text description above except it assumes EBCDIC data on the tape. All data written to tape is translated from ASCII to EBCDIC. All data read from tape is translated from EBCDIC to ASCII.
data	The data is not modified in any way.
edata	The data is not modified in any way except for ASCII/EBCDIC translation as described in the etext description above.

Text and Data File Formats

There are special considerations to be addressed when using the IBM and ANSI formats. The UNIX command `file` is fairly accurate in characterizing files along the text/data lines. If you are in doubt, you can always write a file out as a data file.

ANSI Text Files

A text file consists of readable data and has many newlines throughout its body. To write text files on an ANSI volumeset, the following is advised.

- Set `convert=text`.
- Set `rformat=vbs:32768:32760`.

The block size can be adjusted up or down depending on the attributes of the tape drive. However, these settings should suffice in most cases.

IBM Text Files To write text files on an IBM volumeset, the following is advised.

- Set `convert=etext`.
- Set `rformat=vbs:32768:32760`.

The block size can be adjusted up or down depending on the attributes of the tape drive. However, these settings should suffice in most cases.

ANSI Data Files A data file consists of unreadable data. To write data files on an ANSI volumeset, the following is advised.

- Set `convert=data`.
- Set `rformat=u:32768:32768`.

The block size can be adjusted up or down depending on the attributes of the tape drive. However, these settings should suffice in most cases.

IBM Data Files To write data files on an IBM volumeset, the following is advised.

- Set `convert=edata`.
- Set `rformat=u:32768:32768`.

The block size can be adjusted up or down depending on the attributes of the tape drive. However, these settings should suffice in most cases.

**IBMU (Unlabeled)
Tape Handling**

IBMU tapes are the same as IBM tapes only without the volume and file labels. The IBM standard defines unlabeled tapes as being restricted to only one volume - no multivolume tapesets are allowed. However, REELlibrarian does support single file, multivolume IBMU tapesets.

Record Conversion Record translation is handled just as for IBM tapes. For more information, refer to *Record Conversion* on page 146.

Volumeset Defaults Default volumeset attributes are stored in user-created default definition volumesets. A default definition volumeset is a volumeset template that is created just like a volumeset, but contains no physical volumes. Default volumesets are distinguished by the special volumeset name prefix *DF*.

When a volumeset is created, default settings for its attributes are inherited from the most closely associated default definition volumeset. Any or all of the default settings can be amended by explicitly specifying a value for a particular catalog field.

The benefit of these volumeset defaults is that it allows the user to have a standardized set of attributes used automatically whenever a volumeset is submitted to REELlibrarian.

Default Volumeset Priority The REELlibrarian catalog can store both user-defined defaults and system-wide defaults. A user's own defaults always take priority over system-wide defaults.

User and system wide default definitions are divided into three priority levels. The more-specific default definitions take priority over the less-specific definitions. These levels are listed below, in priority order.

- media type and data format
- media type only
- generic (neither data format nor media type specified)

When selecting default settings for a newly created volumeset, REELlibrarian checks the user-defined default definitions first. If a default definition exists that matches the specified media type and data format of the new volumeset, then the values of that default definition are used. If there is no match, then REELlibrarian checks to see if a default definition exists that matches the media type only. If there is still no match, the generic default definition is used, if it exists. REELlibrarian exhausts all of

the user-defined defaults before it employs any of the system-wide defaults.



Note: Defaults are not effective until you exit and re-enter the `r1` program.

Volumeset Default Formats

The following table illustrates the format of default definition volumesets used by REELlibrarian and their priority.

Table 4-9 Volumeset Default Definitions

Volumeset Name	Description
<code>DF.generic</code>	Specification of last resort. Used only if there is not a more-specific default as described by the remaining entries below.
<code>DF.mtype</code>	Used unless there is a more-specific description for the particular tape format as described below. <code>mtype</code> represents the particular media type.
<code>DF.mtype.A</code> <code>DF.mtype.I</code> <code>DF.mtype.U</code> <code>DF.mtype.T</code> <code>DF.mtype.C</code>	These are the most-specific default setting allowed. The suffixes represent defaults for: A - ANSI volumesets; I - IBM volumesets; U - Unlabeled (IBMU) volumesets; T - TAR volumesets; and C - CPIO volumesets.

Again, the more-specific entries are consulted first. Less-specific entries are used if the more specific entries do not exist for the particular type of volumeset being created.

Creating and Using Default Definition Volumesets

As mentioned earlier, a default volumeset is created just like a normal volumeset except it is distinguished with the special `DF.` prefix. Creating such a default definition will allow you eliminate repetitive work when submitting volumesets to REELlibrarian.

For example, suppose you want to create a volumeset default called `DF.3480` for all 3480 media tapes in your `work1` pool to use and have the non-physical volume named `test-1` (with VSN of 000890) assigned to it.



Note: Before creating a default definition volumeset, be sure that the “Volumes must be Accepted” flag in the Miscellaneous Screen (accessed through `rlconfig`) is set to YES. Otherwise, the default definition volumeset will not be accessible.

To do this using the command-line interface, you must first submit the new volumeset to REELlibrarian. Type:

```
rlvsubmit type=3480 pool=work1 lvsn=000890
DF.3480
```

where 3480 is the media type for volumes in the volumeset to us, `work1` is the pool that the volumes will belong to, and 000890 is the VSN of the first volume assigned to the new volumeset default.



Note: The volumeset will be submitted but not accepted by REELlibrarian.

Now, any volumesets submitted with a media type of 3480 volumeset will automatically inherit the characteristics specified within the default volumeset `DF.3480`.



Note: A volumeset’s attributes can be edited using the `rlvedit` command. Please see the `rlvedit(1)` manpage in Appendix A for more information.

File Name Templates

File name templates are used with the `rlvwrite` command to dynamically construct names for files written to a volumeset. This is particularly useful when used with default volumeset definitions.

The file name constructed by the template is limited to seventeen (17) characters. The template consists of constant text and substitution patterns. Substitution patterns have the general form of: `@sublen@` where the “@” symbol delimits the beginning and end of the substitution.specification. *sub* is a character indicating

what to substitute and *len* can be any number between 1 and 17, however, for some values of *sub* only a particular value of *len* is reasonable.

The values recognized by the template are shown in Table 4-10, *File Name Template Values*.

Table 4-10 File Name Template Values

Value	Description
@Y4@	Numeric year
@C2@	Numeric month (e.g., January = 1)
@E3@	Month name
@D2@	Day of Month
@J3@	Julian Day
@W3@	Day of Week (e.g., Sunday = 1)
@H2@	Hour in 24-hour time (e.g., 3:00 p.m. = 15:00)
@M2@	Minute
@S2@	Second
@F9@	Value assigned <i>fid</i> = keyword on the <i>rlvwrite</i> command
@U9@	Base name of file assigned to the <i>if</i> = keyword on the <i>rlvwrite</i> command
@G4@	File generation
@V2@	File version

For example, if you entered a volume into a volumeset on January 27, 1989 specifying *ftemplate=@D2@@E3@@Y2@@G@G4@V@V2@* in the *rlvwrite* command syntax, it would result in a file name

of: 25Jan89G0000V01. Please see the example in the *Test Run* in Chapter 1 for additional guidance on using file name templates.



Note: Numeric values are truncated on the left and character values are truncated on the right. If a numeric value is shorter than len it is padded on the left with zeros to ensure consistency.

Direct Device and Volume Access

When direct access to a tape drive's filesystem device names is needed, the `rlvmount` command should be used.

Like `rlvaccess`, `rlvmount` requests a volume mount on an appropriate drive. Once mounted though, the `rlvread` and `rlvwrite` commands cannot be used with the volumeset. The volumeset can only be accessed via the filesystem device names.

An example session is shown below.

```
rlvmount .book-299
REW='rliq -r'
NREW='rliq -n'
export REW NREW
find . -print | cpio -oc > $REW
rlvunmount
```

The second command line using `rliq` sets up two environment variables named `REW` and `NREW` equated to the rewind and no-rewind device names, respectively. Note the use of the `REW` variable in the third command.

The `rliq` command can also be used with reserved devices. Here is a revised mounting sequence using the `rliq` command to set environment variables.

```
rlvmount .work-321 on dev1
REW='rliq -r dev1'
```

```

NREW='rliq -n dev1'

export REW NREW

rlvmount .book-730 on dev2

REW2='rliq -r dev2';NREW2='rliq -n dev2'

export REW2 NREW2

```

The `-r` option requests the rewind device name, and the `-n` option does the same for the no-rewind device name.

Using Multiple Tape Drives

REELibrarian supports simultaneous use of multiple tape drives. The reservation program `rlreserve` allows you to request several tape drives and determines whether there are free drives which can satisfy those requirements. If there are, it allocates them to you. If there are not any drives available, the reservation request can be queued until it can successfully reserve the appropriate drives.

REELibrarian completely prevents deadlock situations so that you are never waiting for a device that can never become available due to allocation conflicts.

The `rlreserve` command reserves multiple tape drives. The following command reserves two drives of format 1600.

```

rlreserve type=1600,1600 as dev1,dev2

```

The above command also assigns them the pseudonyms `dev1` and `dev2`. Order is important, so if the type list had been `1600,6250` then `dev1` would refer to the 1600 drive and `dev2` to the 6250 drive.

The user can optionally specify the specific drives required by including the tape drive names (ADNs). The following command requests `drive1` and `drive3`:

```

rlreserve adn=drive1,drive3 as dev1,dev2

```

Like the `rlvaccess` command, if `rlreserve` cannot immediately satisfy the user's request, it will ask the user if the request should be queued.

Once devices are reserved, access requests can be issued at any time. The following example uses the devices reserved in the previous example.

```
rlvaccess forecast on dev1
```

```
rlvaccess history on dev2
```

In the above example, the `on` argument distinguishes between the two drives.

Deadlock Situations

In general, multiple resource requests may fail due to deadlock situations. A deadlock situation occurs when user A requests a resource allocated to user B and user B may be requesting a resource under control of user A. These requests can never be satisfied and so the situation is hopelessly deadlocked. REELlibrarian prevents these possibilities by disallowing the requests that can cause them.

You can only have one session requested and in use at a time. That is, you cannot use `rlreserve` to request three drives and then use a separate request to reserve two additional drives. The second request is immediately rejected because of the deadlock possibility. The deadlock/rejection message appears as shown below:

```
rlreserve adn=drive1,drive3 as dev1,dev2
```

```
rlreserve failed: Deadlock possible
```

Freeing Reserved Drives

Drives reserved via the `rlreserve` command can be released one at a time or altogether. The example below releases one of the drives reserved in the previous example.

```
rlfree dev1
```

To free all drives simultaneously, use the following command:

```
rlfree
```

Freeing a drive automatically unmounts any volumeset still on the drive.

Command Summary

REELlibrarian user commands are listed in Table 4-11, *REELlibrarian User Commands*. Refer to Appendix A for manpage descriptions of each command.

Table 4-11 REELlibrarian User Commands

Command	Description
rl	Full-screen interface to the request monitor and all other full-screen operator programs.
rledit	Modify a volume's catalog entry.
rlfedit	Modify the catalog entry of a volume.
rlfree	Free previously reserved drives.
rlinq	Display the filesystem pathnames for the tape drive.
rlpmsg	Send a message to the REELlibrarian operator.
rlpcreate rlpdelete rlpedit	Create, delete and edit a pool.
rlpsubmit rlpretrieve	Submit to and retrieve from the library a scratch volume.
rlr	Generates REELlibrarian reports.
rlrcreate rlrdelete	Create and delete rotations.
rlreserve	Reserve one or more tape drives.
rlstatus	Reports the current request queue.

Table 4-11 REELlibrarian User Commands (Continued)

Command	Description
rlunq	unqueue REELlibrarian requests
rlvaccess rlvrelease	Initiate and finish a session with a volumeset.
rlvread rlvwrite	Read and write files to the accessed volumeset.
rlvdisplay	Display contents of the accessed volumeset.
rlvcreate rlvedit rlvscratch	Create, edit, and delete a volumeset.
rlvsubmit rlvretrieve	Submit and retrieve a volumeset from the library.
rlvmove	Request a volumeset to be moved from its current site.
rlvmount rlvunmount	Mount and unmount a single volume for direct access.
rlvtran	Reinitialize a transient volumeset.
rlvtruncate	Truncate a volumeset

Chapter 5. Using Reports

Overview

REELibrarian offers extensive reporting capabilities. The complete set of REELibrarian reports are available via the `r1r(1)` command; a subset of these reports is also available via the full-screen interface.

Default report output includes descriptive field names, column headings, and date stamps, but reports can also be requested in a parsable format, with colons as field delimiters. The scope of many of the REELibrarian reports can be limited via keyword arguments.

User Reports

The most commonly requested REELibrarian user reports are available through selection 11, Reports, of the User Tape Access Menu. The full set of user reports are available via the command-line interface using the `r1r` command.



Note: Refer to the `r1r(1)` manpage in Appendix A for a complete syntactical description of the `r1r` command.

Full-Screen User Reports

The Reports Menu is shown in Figure 5-16. To request a report:

1. Start the `r1t1m` program.
2. Press 7 for Reports.

3. Type the number of the report you want to run and press Return.

If more information is required to process the request, a pop-up window will appear on screen prompting for the necessary data. Enter the data at the prompt and press Return.

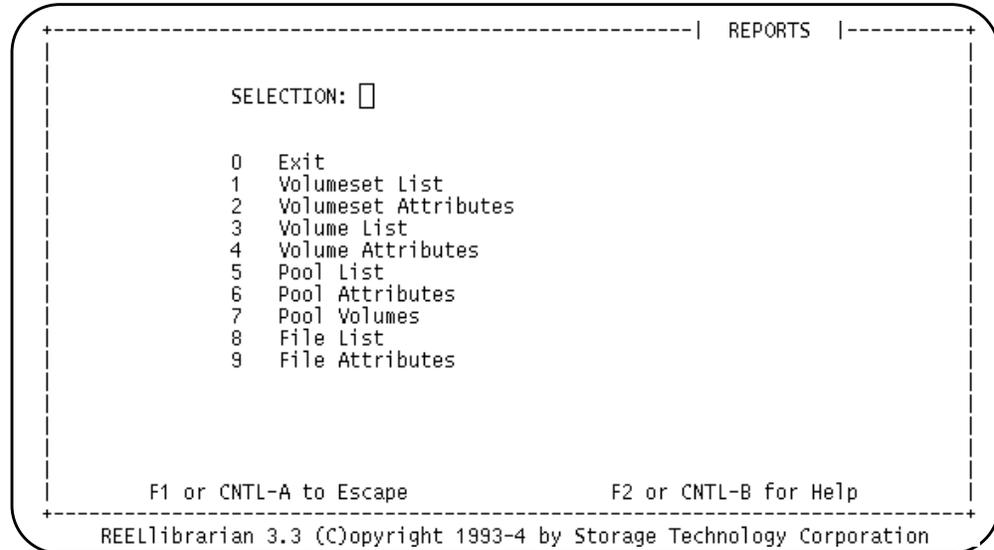


Figure 5-1 Reports Menu

Volumeset List



Command Line Equivalent: `rlr vslist`. Accepted keywords: `app`, `full`.

The Volumeset List report lists all the user's volumesets. It is available through selection 1 of the Reports Menu. A sample Volumeset List report is shown in Figure 5-2. Report fields are described in Table 5-1, *Volumeset List Report Fields*.

```

Volumeset List Report: Fri Sep 16 14:02:07 1994

Vname          STAT  Vexpire      Expires      Vcomment
-----
testvs         RET   I            99999 D
volset1        LIB   S            -99999 G
volset2        LIB   S            -99999 G
volset3        LIB   S            -99999 D

Command: rlr user=bob full=yes vslist

```

Figure 5-2 Volumeset List Report

Table 5-1 Volumeset List Report Fields

Field	Description
Vname	The volumeset name.
STAT	The volumeset status. One of: move - scheduled for movement LIB - cataloged RET - scheduled for return.
Vexpire	Volumeset expiration date. One of: I - never expires S - always expired Rn - expire n days after creation An - expire if not accessed in n days L - expire when all files on the volumeset have expired Xmm/dd/yy - expire on the given date Gn - expire when there are n newer generations
Expires	Expiration scheme. The volumeset expires when the specified number of newer generations (G) are created or when the specified number of days (D) elapse.
Vcomment	Always blank.

Volumeset Attributes



Command Line Equivalent: `rlr volset=volset vinfo`. Other accepted keywords: `full`, `vid`.

The Volumeset Attributes report displays the catalog record for the first volume in the named volumeset. It is available through selection 2 of the Reports Menu. A sample Volumeset Attributes report is shown in Figure 5-3. Report fields are described in Table 5-2, *Volumeset Attributes Report Fields*.

```
Volume Information Report: Fri Sep 16 14:04:54 1994

vname: bob/volset2:G0000:V00:N001

  vid: bunk-779          type: CART          uname: bob
  vsn: 345678           ctype:             gname: sceptre
  rack: 000             length: 2400       vmode: 700
  vsid: bunk-779       format: ANSI       passwd:
  vno: 1                ftrack: no        pool: private
valloc: yes            rformat: u:10240:10240 vexpire: S
  cloc: onsite          conv: data         vacc: ' '
  sloc: onsite          scratch: no        offset: 0
  floc: onsite          maint: 0           ucnt: 0
dispose:              status: LIB        ccnt: 0
  init: no             ftemp:            app:
  ctime: Thu Sep 22 20:35:30 1994
  mtime: Thu Sep 22 20:35:30 1994
  atime: Thu Sep 22 20:35:30 1994
  fingerp: 'X          Uninspected'
  vcom:
Command: rlr vid=bunk-779 full=yes vinfo
```

Figure 5-3 Volumeset Attributes Report

Table 5-2 Volumeset Attributes Report Fields

Field	Description
<code>vname:</code>	The volumeset name.
<code>vid:</code>	The volume ID of the first volume in the volumeset.

Table 5-2 Volumeset Attributes Report Fields (Continued)

Field	Description
vsn	The volume serial number.
rack	The rack number.
vsid	The volume ID of the first volume in the volumeset.
vno	For the volumeset report, this is always 1.
valloc	Set to <i>yes</i> , if the first tape currently contains active data - that is, the tape is not scratched.
cloc	The volumeset's current location.
sloc	The volumeset is scheduled to move to the named site.
floc	When the volumeset expires, the volumes are scratched and scheduled for movement to the named site.
dispose	Always blank.
init	Set to <i>yes</i> if the tape is scratched. Set to <i>no</i> if the tape contains active data.
type	The volumeset's media type.
ctype	The media type the tape is employed as. This can differ from the type value if the media type has media aliases.
length	The length in feet or capacity in megabytes of the tapes.
format	The tape label standard used by the volumeset.
ftrack	Always set to <i>yes</i> .
rformat	The file format: <i>fmt:blen:rlen</i> . <i>fmt</i> is the record type - REEL librarian always sets this to <i>u</i> for unformatted. <i>blen</i> is the block length. <i>rlen</i> is the record length.

Table 5-2 Volumeset Attributes Report Fields (Continued)

Field	Description
conv	Always set to data.
scratch	Set to yes if the tape is scratched.
maint	Internal flags representing maintenance events.
status	One of: SUB - submitted to library LIB - already in library RET - scheduled to be returned to scratch pool
ftemp	Always set to blank.
uname	The volumeset owner. Always set to root.
gname	The volumeset group owner.
vmode	The permission mask similar to the UNIX file permission mask. Always set to 755.
passwd	Always unset.
pool	The pool the volumeset belongs to.
vexpire	Volumeset expiration date. One of: I - never expires S - always expired Rn - expire n days after creation An - expire if not accessed in n days L - expire when all files on the volumeset have expired Xmm/dd/yy - expire on the given date Gn - expire when there are n newer generations
vacc	Always set to ' '.
offset	Always set to 0.
ucnt	Total number of mounts for the first volume.
ccnt	Total number of mounts for the first volume since its last cleaning.

Table 5-2 Volumeset Attributes Report Fields (Continued)

Field	Description
app	Application name. The cataloged application for the volumeset, if assigned.
ctime	Creation time. The time the volumeset was created.
mtime	Modification time. The time the volumeset was last modified.
atime	Access time. The last time the volumeset was accessed.
fingerp	Fingerprint. The electronic fingerprint for the first tape. Used to identify the tape.
vcom	Always unset.

Volume List



Command Line Equivalent: `rlr vlist`. Keyword accepted: `full`.

The Volume List report lists all volumes owned by the user. It is available through selection 3 of the Reports Menu. A sample Volume List report is shown in Figure 5-4. Report fields are described in Table 5-3, *Volume List Report Fields*.

```

Volumes Report: Fri Sep 16 14:03:24 1994

VID          STAT Type      Length  Location      Pool
---          -
town-059    RET  CART      2400    onsite        >R013491
honk-986    RET  CART      2400    onsite        R000218
bunk-779    LIB  CART      2400    onsite        private
nail-194    LIB  CART      2400    onsite        private
ally-001    LIB  CART      2400    onsite        private

Command: rlr user=bob full=yes vlist

```

Figure 5-4 Volume List Report

Table 5-3 Volume List Report Fields

Field	Description
VID	Volume ID.
STAT	Tape status. One of: ctf - tape requires certification cln - tape requires cleaning lost - tape is lost move - tape is scheduled for movement LIB - tape is cataloged rem - tape should be removed stc - tape is in a stack RET - tape is ready to be retrieved init - tape requires initialization
Type	Media type.
Length	Length in feet or capacity in megabytes.
Location	Tape resides at the named site.

Table 5-3 Volume List Report Fields (Continued)

Field	Description
Pool	Tape belongs to the named pool. If the pool name is preceded by a ">" symbol, then the tape is in the scratch state. If the tape is in the retrieval state (RET), the retrieval receipt number is displayed in this field.

Volume Attributes



Command Line Equivalent: `rlr vid=vid vinfo`. Other accepted keywords: `full`, `volset`.

The Volume Attributes report displays the catalog record for the named volume. It is available through selection 4 of the Reports Menu. This report is identical to the Volumeset Attributes report; a sample Volumeset Attributes report is shown in Figure 5-3 on page 162. Report fields are described in Table 5-2, *Volumeset Attributes Report Fields*.

Pool List



Command Line Equivalent: `rlr plist`. Keyword accepted: `full`.

The Pool List report names all the pools owned by the user. It is available through selection 5 of the Reports Menu. A sample Pool List report is shown in Figure 5-5. Report fields are described in Table 5-4, *Pool List Report Fields*.

```
Pool List Report: Fri Sep 16 14:05:13 1994

Pool name          Cln Rem Add Hold
-----          --- --
private           no  no  no  0
safepool          no  no  no 30
cleanpool         no  no  yes 0

Command: rlr user=bob full=yes plist
```

Figure 5-5 Pool List Report

Table 5-4 Pool List Report Fields

Field	Description
Cln	Indicates if cleaning maintenance is enabled for the pool.
Rem	Indicates if end-of-life tape disposal is enabled for the pool.
Add	Indicates if dynamic volume addition is enabled for the pool.
Hold	The number of days a tape is held after its data has expired. After the holding period, the tape enters the scratch state and can be drafted into a volumeset.

Pool Attributes



Command Line Equivalent: `rlr pool=pool pinfo`. Keyword accepted: `full`.

The Pool Attributes report displays all information about a pool except for its member tapes. It is available through selection 6 of the Reports Menu. A sample Pool Attributes report is shown in Figure 5-6. Report fields are described in Table 5-5, *Pool Attributes Report Fields*.

```
Pool Information Report: Fri Sep 16 15:58:46 1994

      cleaning: no
      removal: no
dynamic add: no
      hold: 0
User List: bob          dpm          pes
Group List: sceptre    wheel
Command: rlr pool=private full=yes pinfo
```

Figure 5-6 Pool Attributes Report

Table 5-5 Pool Attributes Report Fields

Field	Description
cleaning	If cleaning maintenance is enabled, this is the number of mounts between cleanings. Tapes are cleaned only when they are in the scratch state. If the displayed value is no, then cleaning maintenance is off.
removal	If end-of-life tape disposal is enabled, the displayed value represents the cumulative number of mounts that cause automatic tape disposal. If the displayed value is no, then tapes are not automatically disposed.
dynamic add	If set to yes, then REELlibrarian prompts the operator to add new tapes to the pool when there are no more scratch tapes. REELlibrarian prompts via the Request Monitor and only if a pending mount request requires a scratch tape from the pool. If set to no, then this feature is disabled.
hold	The number of days a tape is held after its data has expired. After the holding period, the tape enters the scratch state and is eligible for regular usage and allocation.
User List	The list of user names allowed to use the pool. A user name prefixed with a “!” indicates that the user is specifically prohibited from using the pool. The special name ANY means all users have access to the pool.
Group List	The list of group names allowed to use the pool. A group name prefixed with a “!” in front of indicates that the group is specifically prohibited from using the pool. The special name ANY means all groups have access to the pool.

Pool Volumes



Command Line Equivalent: `rlr pool=pool pvolumes.`
Keyword accepted: `full.`

The Pool Volumes report lists all of the tapes belonging to the named pool. It is available through selection 7 of the Reports Menu. A sample Pool Volumes report is shown in Figure 5-7. Report fields are described in Table 5-6, *Pool Volumes Report Fields*.

```
Pool Volumes Report: Fri Sep 16 14:06:40 1994

Vname          Scratch  Type      VID
-----        -
SCRATCH        yes      CART      town-059
testvs         no       CART      honk-986
volset2        no       CART      bunk-779
volset3        no       CART      nail-194
volset1        no       CART      ally-001

Command: rlr pool=private full=yes pvolumes
```

Figure 5-7 Pool Volumes Report

Table 5-6 Pool Volumes Report Fields

Field	Description
Vname	If the tape is active, then this is the name of the volumeset to which it belongs. If the tape is scratch, then the name SCRATCH is displayed.
Scratch	Scratch status. If the tape is active, then no is displayed here. If the tape is scratched, then <code>maint</code> is displayed.
Type	The media type of the volume.
VID	The volume ID.

File List



Command Line Equivalent: `rlr flist` Keywords accepted: `app, full`.

The File List report lists all files owned by the user. It is available through selection 8 of the Reports Menu. A sample File List report is shown in Figure 5-8. Report fields are described in Table 5-7, *File List Report Fields*.

```
File List Report: Fri Sep 16 16:58:04 1994

Vname          File Name          Expires
-----          -
filevol        file1              S
filevol        file2              S
filevol        file3              S
filevol        file4              S
filevol        file5              S
tempvs        file1              S
tempvs        file2              S

Command: rlr user=bob full=yes flist
```

Figure 5-8 *File List Report*

Table 5-7 *File List Report Fields*

Field	Description
Vname	The name of the volumeset on which the file resides.
File Name	The file name.

Table 5-7 File List Report Fields (Continued)

Field	Description
fexpire:	File expiration date. One of: I - never expires S - always expired Rn - expire n days after creation An - expire if not accessed in n days Xccyyymmdd or Xmm/dd/yy - expire on the given date Gn - expire when there are n newer generations. This data is only considered if volumeset expiration is set to L. The expiration scheme for the file.

File Attributes



Command Line Equivalent: `rlr volset=volset fid=fid finfo`. Keywords accepted: `fseq`, `fsect`, `full`.

The File Attributes report lists the catalog information for the specified file. It is available through selection 9 of the Reports Menu. A sample File Attributes report is shown in Figure 5-9. Report fields are described in Table 5-8, *File Attributes Report Fields*.

```

File Information Report: Fri Sep 16 14:31:36 1994

owner   : bob
fid     : file2
gname   : daemon
fmode   : 700
vsid    : teak-960
vname   : tempvs:G0000:V00
rformat : vbs:4096:512
conv    : text
offset  : 0
passwd  :
blocks  : 8
vno     : 1
fno     : 1
mask    : 3f 1c
fexpire : S
app     :
ctime   : Tue Sep 13 10:55:34 1994
atime   : Tue Sep 13 10:55:34 1994
fcom    :

Command: rlr fid=file2  fsect=1 full=yes finfo

```

Figure 5-9 File Attributes Report

Table 5-8 File Attributes Report Fields

Field	Description
owner	The user name which owns the file.
fid	The file name.
gname	The owner's group name.
fmode	The file permission mask.
vsid	The volume ID which contains the file.

Table 5-8 File Attributes Report Fields (Continued)

Field	Description
vname	The name of the volumeset on which the file resides.
rformat	The file format: <i>fmt:blen:rlen</i> . Where <i>fmt</i> is the record format. <i>blen</i> is the block length. <i>rlen</i> is the record length.
conv	Data type. One of: text for ASCII text, data for ASCII binary data, etext for EBCDIC text, or edata for EBCDIC binary data.
offset	Offset byte. The offset byte value in the label.
passwd	Password.
blocks	Number of blocks in the file.
vno	The ordinal number of the volume in the volumeset.
fno	The ordinal number of the file within the volumeset.
mask	For internal use only.
fexpire	File expiration date. One of: I - never expires S - always expired R <i>n</i> - expire <i>n</i> days after creation A <i>n</i> - expire if not accessed in <i>n</i> days X <i>ccyyymmdd</i> or X <i>mm/dd/yy</i> - expire on the given date G <i>n</i> - expire when there are <i>n</i> newer generations. This data is only considered if volumeset expiration is set to L.
app	Application name associated with the file.
ctime	Creation time. Set to the time the file was created.
atime	Access time. Set to the time the file was last accessed.
fcom	File comment.

Command-Line User Reports

Rotation Information The Rotation Information report displays the schedule of the specified rotation. It is requested with the command:

```
rlr rotation=rotation_name rinfo
```

Keyword accepted: full.

A sample Rotation Information report is shown in Figure 5-10. Report fields are described in Table 5-9, *Rotation Information Report Fields*.

```
Rotation Information Report: Fri Sep 16 14:39:12 1994

      Location      Type      Count
      -----      -
      onsite        R         30
      offsite       R         30
      onsite        R         30

Command: rlr rotation=rotation2 rinfo
```

Figure 5-10 Rotation Information Report

Table 5-9 Rotation Information Report Fields

Field	Description
Location	The storage vault for a single step in the rotation itinerary.
Type	The rotation duration type. R indicates a days-based rotation; G indicates a generation-based itinerary.
Count	The length of time in days (for rotation type R) or generations (for rotation type G) that volumesets assigned to this rotation will reside at this location before rotating to the next location on the rotation list.

Rotation List The Rotation List report lists all the rotations owned by the user issuing the command and the number of steps included in each rotation. It is requested with the command:

rlr rlist

Keyword accepted: full.

A sample Rotation List report is shown in Figure 5-11. Report fields are described in Table 5-10, *Rotation List Report Fields*.

```
Rotation List Report: Fri Sep 16 14:35:44 1994

Rotation          Locations
-----          -
rotation1         2
rotation2         3
rotation3         4

Command: rlr user=bob full=yes rlist
```

Figure 5-11 Rotation List Report

Table 5-10 Rotation List Report Fields

Field	Description
Rotation	The name of the rotation.
Locations	The number of steps in the rotation itinerary.

Storage Location The Storage Location List report lists all defined storage vaults. It is requested with the command:

rlr sites

Keyword accepted: full.

A sample Storage Location List report is shown in Figure 5-12. Report fields are described in Table 5-11, *Storage Location List Report Fields*.

```

Storage Location List: Fri Sep 16 14:41:35 1994

      Name           Mountable
      ----           -
      onsite          yes
      offsite         no

Command: rlr full=yes sites

```

Figure 5-12 Storage Location List Report

Table 5-11 Storage Location List Report Fields

Field	Description
Name	The name of the vault.
Mountable	Indicates whether tapes at that location are mountable. If no, mount requests for tapes stored in this location are refused.

Media Type The Media Type List report lists all configured media types. It is requested with the command:

rlr types

Keyword accepted: full.

A sample Media Type List report is shown in Figure 5-13. Report fields are described in Table 5-12, *The Media Type List Report Fields*.

```

Media Type List: Fri Sep 16 14:42:49 1994

Type          Clean  Remove
----          -
CART          100    999
DAT           100    999
DATHD        100    999
XBYTE        100    999
XBYTEHD      100    999
1600         100    999
800          100    999
6250         100    999
3200         100    999
3480         100    999
C20          100    999
C40          100    999
C60          100    999
C80          100    999
C100         100    999
C120         100    999
C150         -1     -1

Command: rlr full=yes types

```

Figure 5-13 Media Type List Report

Table 5-12 The Media Type List Report Fields

Field	Description
Type	The media type name.
Clean	The cleaning cycle. Each time a tape accumulates the specified number of mounts, REELlibrarian schedules it for cleaning. Note that REELlibrarian only schedules the tape when it is in the scratch state. A value of -1 means that the cleaning cycle is disabled.

Table 5-12 The Media Type List Report Fields (Continued)

Field	Description
Remove	The end-of-life disposal limit. When a tape accumulates the specified number of mounts, REELlibrarian schedules it for disposal. A tape can only be disposed when it is in scratch state. A value of -1 means that the disposal limit is disabled.

Volumeset File List The Volumeset File List report lists all of the files on the specified volumeset. It is requested with the command:

```
rlr volset=volset vsflist
```

Keyword accepted: full.

A sample Volumeset File List report is shown in Figure 5-14. Report fields are described in Table 5-13, *Volumeset File List Fields*.

```
Volumeset File List: Fri Sep 16 14:48:27 1994

Fid          SC Blocks Fexpire   Expires   Fcomment
---          - - - - - - - - - - - - - - - - - - - - - -
file1        1  8      - - - - - - - - - - - - - - - - - -
file2        1  8      - - - - - - - - - - - - - - - - - -
file3        1  8      - - - - - - - - - - - - - - - - - -

End of Tape

Command: rlr volset=filevol full=yes vsflist
```

Figure 5-14 Volumeset File List Report

Table 5-13 Volumeset File List Fields

Field	Description
Fid	The name of the file.

Table 5-13 Volumeset File List Fields

Field	Description
SC	File section. If the tape spans more than one tape, then the first tape has section 1, the second tape has section 2, etc.
Blocks	Number of blocks in the file.
Vexpire	File expiration date. One of: I - never expires S - always expired Rn - expire n days after creation An - expire if not accessed in n days Xccyyymmdd or Xmm/dd/yy - expire on the given date Gn - expire when there are n newer generations. This data is only considered if volumeset expiration is set to L.
Expires	Expiration scheme. The file expires when the specified number of newer generations (G) are created or when the specified number of days (D) elapse.
Fcomment	File comment.

Volumeset Volume List

The Volumeset Volume List report lists all of the volumes that constitute the specified volumeset. It is requested with the command:

```
rlr volset=volset vsvlist
```

Keyword accepted: full.

A sample Volumeset Volume List report is shown in Figure 5-15. Report fields are described in Table 5-14, *Volumeset Volume List Report Fields*.

```

Volumeset Volume List: Fri Sep 16 15:08:31 1994

Volset      Vno  Vid      Vsn      Location      Alloc  Media Type
-----
tempvs      1    teak-960  020113  onsite        yes    CART
tempvs      2    dole-206  049567  onsite        yes    CART
tempvs      3    bind-423  098765  onsite        yes    CART

Command: rlr volset=tempvs full=yes vsvlist

```

Figure 5-15 Volumeset Volume List Report

Table 5-14 Volumeset Volume List Report Fields

Field	Description
Volset	The name of the volumeset.
Vno	Volume Number. The order in which the volume occurs on the volumeset.
Vid	The volume ID.
Vsn	The volume serial number.
Location	Current location of the tape.
Alloc	If set to yes, the tape contains active data. If set to no, the tape is scratched.
Media Type	The media type of the volume.

Operator Reports

The reports listed in *User Reports* on page 159 are a subset of the REELlibrarian operator reports. Operators may also use the keyword `user` to limit the search to objects owned by a specific library user. This keyword is not available to library users without operator or root permissions.

The reports detailed in this section are available to REELlibrarian operators and administrators only. The most commonly requested REELlibrarian operator reports are available through selection 7, Reports, of the Library Management Menu. The full set of operator reports are available via the `r1r` command.



Note: Refer to the `r1r(8)` manpage in Appendix D for a complete syntactical description of the `r1r` command.

Full-Screen Operator Reports

The Operator Reports Menu is shown in Figure 5-16. To request a report, type the number of the report and press Return. If more information is required to process the request, a pop-up window will appear on screen prompting for the necessary data. Enter the data at the prompt and press Return.



Note: Reports 1 through 9 are documented in *User Reports* on page 159.

```
+-----+ | REPORTS | +-----+
|
| SELECTION: 
|
| 0  Exit
| 1  Volumeset List
| 2  Volumeset Attributes
| 3  Volume List
| 4  Volume Attributes
| 5  Pool List
| 6  Pool Attributes
| 7  Pool Volumes
| 8  File List
| 9  File Attributes
|10  Volume Maintenance
|11  Volume Inventory
|
| F1 or CNTL-A to Escape          F2 or CNTL-B for Help
+-----+
REELlibrarian 3.3 (C)opyright 1993-4 by Storage Technology Corporation
```

Figure 5-16 Operator Reports Menu

Volume Maintenance



Command Line Equivalent: rlr maint. Keywords accepted: action, full, location.

The Volume Maintenance report lists all currently scheduled maintenance activities. It is available through selection 10 of the Operator Reports Menu. A sample Volume Maintenance report is shown in Figure 5-17. Report fields are described in Table 5-15, *Volume Maintenance Report Fields*.

```
Volume Maintenance Report: Fri Sep 16 11:52:58 1994

Action  From                      To                      VID
-----
Move    onsite/303                      offsite/                bind-423

Erase:  none
Certify: none
Remove: none

Action Location      Rack      ID
-----
Lost  onsite      555      text-166

Action Location      Rack      ID
-----
Accept onsite      ??????  lead-399

Action Location      Rack      ID
-----
Id    onsite      ??????  calm-265

Action Location      Rack      ID
-----
Return onsite      234      town-059
Return              444      honk-986

Command: rlr full=yes maint
```

Figure 5-17 Volume Maintenance Report

Table 5-15 Volume Maintenance Report Fields

Action	Description
Move	Lists the tapes scheduled for movement. The From and To columns describe the origin and destination sites. Confirm with <code>rlmoved</code> .
Erase	Lists the tapes scheduled for erasure. Confirm with <code>rlerased</code> .
Certify	Lists all tapes requiring certification due to tape errors. Confirm with <code>rlcertify</code> .
Remove	Lists the tapes scheduled for disposal as part of life-cycle management. The tapes do not contain current data. Confirm with <code>rlremoved</code> .
Lost	Lists all tapes marked as lost in the catalog. If a tape is truly lost, then it should be removed from the catalog with <code>rlremoved</code> . If a tape can be located, then use <code>rlfound</code> to update the catalog.
Clean	Lists all tapes scheduled for cleaning. Confirm with <code>rlcleaned</code> .
Accept	Lists all tapes awaiting acceptance. Conduct acceptance with <code>rlaccept</code> .
Id	Lists all tapes requiring electronic identification. Conduct ID inspection with <code>rlid</code> .
Return	Lists all tapes to be returned to users. Conduct return with <code>rlreturn</code> .

Volume Inventory



Command Line Equivalent: `rlr vinventory`. Keywords accepted: `full`, `location`, `pool`, `rack`.

The Volume Inventory report lists all tapes in the library sorted by vault location and rack number. It is available through selection 11 of the Operator Reports Menu. A sample Volume Inventory report is shown in Figure 5-18. Report fields are described in Table 5-16, *Volume Inventory Report Fields*.

```
Volume Inventory Report: Fri Sep 16 12:26:15 1994

Location      Rack      VID      CLN  USG  Pool
-----      -
onsite        000      bunk-779  12  12  bob/private
onsite        001      town-059   0   0  >bob/private
onsite        002      ally-001   0   0  bob/private
onsite        003      honk-986   8  16  bob/private
onsite        004      nail-194   0   0  bob/private
onsite        999      teak-960   0   1  root/private
onsite        ACC      calm-265   0   0  >root/private
onsite        ACC      hose-676   0   0  >root/private
onsite        SUB      lead-399   0   0  >root/private
onsite        SUB      sign-685   0   0  >root/private
onsite        SUB      town-024   0   0  >root/private

Command: rlr      full=yes vinventory
```

Figure 5-18 Volume Inventory Report

Table 5-16 Volume Inventory Report Fields

Field	Description
Location	The vault where the volume is stored.
Rack	The rack where the volume is stored.
VID	The volume ID of the volume.
CLN	The number of times the volume has been mounted since it was last cleaned.

Table 5-16 Volume Inventory Report Fields (Continued)

Field	Description
USG	The total number of times the volume has been mounted.
Pool	The pool the volume belongs to.

Command-Line Operator Reports

Device Information

The Device Information report displays the configuration of the specified drive. It is requested with the command:

```
r1r adn=device_name dinfo
```

Keyword accepted: full.

A sample Device Information report is shown in Figure 5-19. Report fields are described in Table 5-17, *Device Information Report Fields*.

```

Device Information Report: Fri Sep 16 13:43:37 1994

Type Software selectable: no
Type Software detectable: yes
      Machine: larch

Supported Media Types:

      CART
      Tapecap: SDISK.CART
      Rewind Name: /oak1/pi/reel/lib/tmp/Dev/CART.r
      No-Rewind Name: /oak1/pi/reel/lib/tmp/Dev/CART
      KBytes/second: 100
      Efficient Size: 32768

Other devices:

      none...

Command: rlr adn=drive1 full=yes dinfo

```

Figure 5-19 Device Information Report

Table 5-17 Device Information Report Fields

Field	Description
Type Software selectable	yes indicates that the drive switches between densities under software control.
Type Software detectable	yes indicates the drive automatically adjusts to the recording density of each mounted volume.
Machine	The name of the computer on which the drive resides.
Supported Media Types	The drive can support volumes of this type.
Tapecap	The tapecap entry for the drive.
Rewind Name	The rewind device filename for the drive .

Table 5-17 Device Information Report Fields (Continued)

Field	Description
No-Rewind Name	The no-rewind device filename for the drive .
K bytes/second	The maximum data throughput the drive delivers when writing a tape in units of kilobytes (1024 bytes) per second.
Efficient Size	The buffer size in kilobytes (1024 byte units). The system uses this buffer size when writing data to the tape drive.

Tapecap Report The Tapecap report lists all of the site's configured tapecap entries. It is requested with the command:

```
rlr tapecap
```

Keyword accepted: full, entry.

When issued with the `entry` keyword, this report displays configuration details for the specified tapecap entry. An example is shown below.

```
rlr entry=3480 tapecap
```

A sample Tapecap report is shown in Figure 5-20; a sample Tapecap Entry report is shown in Figure 5-21.

```
SDISK.800
SDISK.1600
SDISK.6250
SDISK.3480
SDISK.3480
SDISK.XBYTE
800
1600
3200
6250
3480
DAT
XBYTE
XBYTEB
XBYTEC
XBYTEM
```

Figure 5-20 TapeCap Report

```
Fixed block type device
Normally closed
Not Seekable
Not Byte Swapped
Min/Mod/Max Buffer:           512/512/32768
Inter block gap (mils):      0
Tapemark size (mils):        6
Density (bpi):               0
End of Tape Detection:       Calculated
Ioctl support (FSF/BSF/REW/EOD/EOF): 0/0/0/0/0
Non-standard 'capabilities' (0x14):
    Append at EOT only
```

Figure 5-21 TapeCap Entry Report

Each of the report's items are described in the following table:

Table 5-18 TapeCap Report Fields

Item	Description
Min/Mod/Max Buffer:	Minimum buffer size/Block Modular Requirement/Maximum Write Buffer Size—all shown in bytes.
Inter block gap (mils):	Interrecord Block Gap. Determines size in which to distinguish between blocks in mils.
Tapemark size (mils)	Length of the tapemark in mils.
Density (bpi):	Density of media in bpi.
End of Tape Detection:	Shows if EOT detection is enabled. “Calculated” means that EOT is disabled. An error code number will appear if EOT is enabled.
Ioctl Support ...	Displays if the following Ioctl calls are available (1) or unavailable (0): <ul style="list-style-type: none"> • FSF (Forward Space File) • BSF (Block Space Forward) • REW • EOD (End of Data) • EOF (End of File)
Non-standard ..	Not applicable.



Note: Displaying the tapecap report in this manner allows the user to see if End of Tape detection is active or inactive. Please see *EOT Detection* on page 51 for more information.

Vault Content

The Vault Content report lists all of the volumes in the onsite vault or the vault specified by the `location` keyword. It is requested with the command:

```
r1r vcontent
```

Keyword accepted: `full, location`.

A sample Vault Content report is shown in Figure 5-22. Report fields are described in Table 5-19, *Vault Content Report Fields*.

```

Vault Contents Report: Fri Sep 16 14:09:20 1994

Rack          VID          Volset          vno Created Expires
----          -
              calm-265     SCRATCH
              hose-676     SCRATCH
              lead-399    SCRATCH
              sign-685    SCRATCH
              tilt-403    SCRATCH
              town-024    SCRATCH
000          bunk-779     bob/volset2     1  9/16/94  10 G
234          town-059     SCRATCH
303          bind-423     filevol         1  9/16/94  9 D
345          ally-001     bob/volset1     1  9/16/94  9 D
444          honk-986     bob/testvs      1  9/16/94  9 D
555          text-166     test2           1  9/16/94  9 D
789          dole-206     SCRATCH
998          dent-781     newvol          1  9/16/94  9 D
999          teak-960     tempvs          1  9/16/94  9 D
Command: rlr location=onsite full=yes vcontent

```

Figure 5-22 Vault Content Report

Table 5-19 Vault Content Report Fields

Field	Description
Rack	The rack where the volume is stored.
VID	The volume ID.
Volset	If the tape is active, then this is the name of the volumeset to which it belongs. If the tape is scratch, then the name SCRATCH is displayed.
vno	If the tape is active, this is the position of the tape in the volumeset.
Created	If the volume belongs to a volumeset, the date the volumeset was created.

Table 5-19 Vault Content Report Fields

Field	Description
Expires	Volumeset expiration date. One of: I - never expires S - always expired Rn - expire n days after creation An - expire if not accessed in n days L - expire when all files on the volumeset have expired Xccyyymmdd or Xmm/dd/yy - expire on the given date Gn - expire when there are n newer generations

Chapter 6. Programming Interface

Overview

The REELlibrarian programming interface is provided through a C function library named `librltape.a`. It provides, at the C function level, the same tape access and control available to users at the command level.

The tape functions are described and demonstrated by example here, and are further detailed in the manpages in Appendix B. The manpages are delivered in electronic form with the product - see the *REELlibrarian Installation Guide* for details.

Tape Programming Requirements

Compiling With the Library: `librltape.a`

All programs written with the programmer library functions must include the library as part of the linking process. For example:

```
cc -o test1 test1.c \  
/usr/local/lib/librltape.a
```

The library may be located elsewhere depending on where it was installed.

Tape Formats

As with the REELlibrarian commands, the programmer library supports all of the tape formats: ANSI, IBM, TAR, CPIO, IBMU, and RAW. When a tape is accessed, the standard blocking and record processing defaults are set. See Chapter 4 for more information on standard processing of each tape format.

Data Structures All pertinent data structures are maintained in include files named `reel_struct.h`, `reel_defs.h`, `reel_err.h`, `reel_types.h`, located in the RLL directory. A single include file named `reel.h` includes all of the others. They must be included in each C source file that uses the REELlibrarian functions.

Sample Programs

Three programs are documented here and are provided in source code form in the Library Directory `example/`. Each program demonstrates the use of the programmer's interface.

Example 1 The first program, `dev.c` manipulates a single volume that is not a member of a volumeset.

A listing of the program (`dev.c`) is provided below.

```
#include <fcntl.h>
#include <stdio.h>
#include "reel.h"

/*
** sample program demonstrating accessing a tape
** using the tape device interface
**
** The argument specifies the name of an existing
** volume
*/

error_x()
{
    fprintf(stderr, "error: %s\n", reel_error());
    exit(1);
}

main(argc, argv)
int    argc;
char   *argv[];
{
    ADN    *rl_inq();
    RLIST  rlist;
    VREC   vrec;
```

```

ADN      *adn;
char     *psd = "rltest";
char     *vname = argc > 1 ? argv[1] : "vol001";

vrec_null(&vrec); /* init vrec fields */

/*
** reserve a device
*/
bzero(&rlist, sizeof(RLIST));
strcpy(rlist.type, "C150");
strcpy(rlist.psd, psd);

if( rl_reserve(0, 5, &rlist) < 0 )
    error_x();

printf("Reservation succeeded\n");

/*
** request a tape mount
*/
if( rl_vmount(psd, vname, "", T_RW, 0, &vrec)
< 0 )
    {
        if(terrno != RL_MWAIT)
            error_x();
        if(rl_wait(WAIT_MNT, psd) < 0)
            error_x();
    }

printf("mount succeeded\n");

/*
** Find out about the device
** We can now open(), read(), write() and
close()
**      the rewind and no-rewind devices.
*/
if( !(adn = rl_inq(psd)) )
    error_x();

printf("\nTape mounted on devices:\n");
printf("  rewind: %s\n", adn->devs->rew);
printf("no rewind: %s\n\n", adn->devs->nrew);

```

```

/* unmount the tape
*/
if( rl_vunmount(psd) < 0 )
    error_x();

/* free the device
*/
if( rl_free(psd) < 0 )
    error_x();

printf("Free succeeded\n");
exit(0);
}

```

Example 2 This second example program writes an ANSI format volumeset. A listing of the program (ansi_write.c) is provided below.

```

#include <sys/types.h>
#include <stdio.h>
#include "reel.h"

/*
** Sample program demonstrating writing a file to an
** ANSI-format
** volumeset. For the test to work, there must be a
** submitted
** ANSI volumeset named 'tansi' or as specified in
** ARG1.
**/

main(argc, argv)
int    argc;
char   *argv[];
{
    int vsd;
    RLIST rlist;
    ADN *adn;
    FSPEC fspec;
    FILE *fp;
    char buf[512];
    char *psd = "test";
    char *volset = argc > 1 ? argv[1] : "tansi";

```

```

setbuf(stdout, (char *)0);

/* reserve a device
*/
printf("Reserving device...");
bzero(&rlist, sizeof(RLIST));
strcpy(rlist.type, "C150");
strcpy(rlist.psd, psd);
if(rl_reserve(0, 5, &rlist) < 0)
    error_x();
printf("complete\n");

/* Access the Volumeset
*/
printf("Accessing Volumeset...");
if((vsd = vs_vaccess(psd, volset, "", T_RW,
0)) < 0)
    error_x();
printf("complete\n");

/* open the proper file
*/
bzero((char *)&fspec, sizeof(FSPEC));
fspec.fselect = FS_FSEQ;          /* select file
by sequence on tape */
strcpy(fspect.fid, "test_file");
fspec.fseq = 1;                  /* first file
on tape */
fspec.fsect = 1;                /* first
section of file */
fspec.rtype = RET_SCRATCH;
fspec.facc = ' ';

fspec.rspec.rfmt = RF_FB;       /* Fixed Block
Record Format */
fspec.rspec.blen = 1000;       /* Block length
*/
fspec.rspec.rlen = 100;        /* Record
length */

fspec.cspec.pflg = '1';        /* enable pad
processing */
fspec.cspec.pchar = ' ';      /* pad character
- blank */

```

```

        fspec.cspec.tflg = '1';           /* enable
termination processing */
        fspec.cspec.tchar = '\n';       /* termination
character - newline */
        fspec.cspec.tran = '0';        /* no
translation */
        fspec.fmode = 0777;

        fspec.fvalid = FV_HDR1 | FV_HDR2 | FV_HDR3;

        printf("Opening Tape file...\n");
        if(vs_open(vsd, T_WRITE, &fspec) < 0)
            error_x();
        printf("complete\n");

        /* write out the file
        */
        if(!(fp = fopen("/etc/passwd", "r")))
        {
            perror("fopen");
            exit(1);
        }

        printf("Writing File...");
        while(fgets(buf, 512, fp))
            if(rec_put(vsd, buf, strlen(buf)) < 0)
                error_x();
        printf("complete\n");

        printf("Closing File...");
        if(vs_close(vsd) < 0)
            error_x();
        printf("complete\n");

        printf("Releasing Volumeset...");
        if(vs_vrelease(vsd, RET_UNSPEC, 0) < 0)
            error_x();
        printf("complete\n");

        printf("Freeing Device...");
        if(rl_free(psd) < 0)
            error_x();
        printf("complete\n");

```

```

        printf("\nTest Complete\n");
        exit(0);
    }

error_x()
{
    printf("error: %s\n", reel_error());
    exit(1);
}

```

Example 3 This third example program reads the volumeset created by the `ansi_write` program. A listing of the `ansi_read.c` program file is provided below.

```

#include <sys/types.h>
#include <stdio.h>
#include "reel.h"

/*
** sample program demonstrating reading a file from
** an ANSI-format
** cartridge tape. ARG1 is the volumeset name.
**
*/

main(argc, argv)
int    argc;
char   *argv[];
{
    int    vsd, n;
    RLIST  rlist;
    ADN    *adn;
    FSPEC  fspec;
    FILE   *fp;
    char   buf[512];
    char   *psd = "test";
    char   *volset = argc > 1 ? argv[1] : "tansi";

    setbuf(stdout, (char *)0);

    /* reserve a device
    */
    printf("Reserving device...");

```

```

bzero(&rlist, sizeof(RLIST));
strcpy(rlist.type, "C150");
strcpy(rlist.psd, psd);

if(rl_reserve(0, 5, &rlist) < 0)
    error_x();
printf("complete\n");

/* Access the Volumeset
*/
printf("Accessing Volumeset...");
if((vsd = vs_vaccess(psd, volset, "", T_READ,
0)) < 0)
    error_x();
printf("complete\n");

/* open the proper file
*/
bzero((char *)&fspec, sizeof(FSPEC));
fspec.fselect = FS_FSEQ;          /* select file
by sequence on tape */
fspec.fseq = 1;                  /* first file
on tape */
fspec.fsect = 1;                 /* first
section of file */

printf("Opening Tape file...\n");
if(vs_open(vsd, T_READ, &fspec) < 0)
    error_x();
printf("complete\n");

/* read the file
*/
while((n = rec_get(vsd, buf)) >= 0)
    fwrite(buf, n, 1, stdout);

printf("\nClosing File...");
if(vs_close(vsd) < 0)
    error_x();
printf("complete\n");

printf("Releasing Volumeset...");
if(vs_vrelease(vsd, RET_UNSPEC, 0) < 0)
    error_x();
printf("complete\n");

```

```
printf("Freeing Device...");
if(rl_free(psd) < 0)
    error_x();
printf("complete\n");

printf("\nTest Complete\n");
exit(0);
}

error_x()
{
printf("error: %s\n", reel_error());
exit(1);
}
```


User Commands (UNIX Section 1)

This appendix includes the UNIX style manual pages for each of the REELibrarian user commands.

Name	Intro - introduction to REELlibrarian
Description	<p>REELlibrarian facilitates the use of tapes and tape drives. It allows you to conduct ad-hoc tape sessions and to store and retrieve files on tape. REELlibrarian keeps an on-line catalog which tracks tapes and their contents.</p> <p>All user tapes are submitted to a central library which is under the control of the operator. You can create new volumesets, via the <code>rlvcreate(1)</code> command, and access them with the <code>rlvaccess(1)</code> command. Files are read and written with the <code>rlvread(1)</code> and <code>rlvwrite(1)</code> commands. The REELlibrarian operator will receive instructions to mount and unmount tapes according to the commands you issue.</p> <p>The following REELlibrarian user commands are described in their own manpages:</p> <ul style="list-style-type: none"><code>reel</code> - REELlibrarian server summary<code>r1</code> - REELlibrarian full screen interface<code>rledit</code> - modify the catalog entry of a volume<code>rlfedit</code> - modify the catalog entry of a file<code>rlfree</code> - free a previously reserved device<code>rlinq</code> - display the path names for the special files associated with a pseudo device<code>rlpmsg</code> - send a message to the REELlibrarian operator<code>rlpcreate</code> - create a new volume pool<code>rlpdelete</code> - delete a volume pool<code>rlpedit</code> - edit a volume pool<code>rlpretrieve</code> - initiate retrieval of a scratch volume<code>rlpsubmit</code> - submit a scratch volume to REELlibrarian<code>rlr</code> - REELlibrarian report generator

`rlrcreate` - create a rotation schedule
`rlrdelete` - delete a rotation schedule
`rlreserve` - reserve one or more devices
`rlstatus` - display device reservation status
`rlunq` - unqueue REELlibrarian requests
`rlvaccess` - initiate access to a volumeset
`rlvcreate` - create a volumeset
`rlvdisplay` - display information about an accessed volumeset
`rlvedit` - modify the catalog entries for volumes in a volumeset
`rlvmount` - mount a volume
`rlvmove` - schedule a volumeset for movement to a different site
`rlvread` - read a file from a volumeset
`rlvrelease` - terminate access to a volumeset
`rlvretrieve` - initiate volumeset retrieval
`rlvscratch` - scratch a volumeset
`rlvsubmit` - submit a volumeset
`rlvtran` - reinitialize a transient volumeset
`rlvtruncate` - truncate a volumeset
`rlvunmount` - unmount a volume
`rlvwrite` - write a file to a volumeset

See Also *REELlibrarian Master Guide.*

reel(1)

Name	reel - REELlibrarian server summary
Synopsis	reel
Description	reel prints a summary of the environment the REELlibrarian servers are running in. It also pings each server and reports whether the server is responding.
See Also	reel(8), <i>REELlibrarian Master Guide</i> .

Name r1 - REELlibrarian full screen interface

Synopsis r1

Description Many features of REELlibrarian are accessible from a full screen interface. r1 starts the full screen interface at the main menu level. Online help is available.

See Also *REELlibrarian Master Guide.*

Name	<code>rledit</code> - modify the catalog entry of a volume
Synopsis	<code>rledit</code> [<i>keywords</i>] <i>vid</i>
Description	<code>rledit</code> modifies the catalog entry of volume <i>vid</i> to reflect the given keyword assignments. A volume may only be edited by its owner.
Options	<i>vid</i> Volume ID. A unique identifier assigned to each volume when it is submitted. It can be up to twelve characters long. <i>keywords</i> One or more keyword=value assignments. Keywords listed are optional with no default specified. Keywords recognized by this command are: <i>app=</i> Application prefix. This prefix is used with file reports to limit selections to those with the specified <i>app=</i> value. <i>app</i> may be up to 12 characters long. <i>capacity=</i> Volume capacity in Mbytes. Specifies the volume capacity for cartridge Media types. Note: for nine-track (or reel) media, use the <i>length</i> keyword. <i>conv=</i> Record conversion specification. This value controls the conversion of records to and from tape. <i>conv</i> may be one of: <i>text</i> for text records; <i>etext</i> for EBCDIC text records, <i>data</i> for fixed length ASCII or binary data records; <i>edata</i> for fixed length EBCDIC data records. Warning: Under ordinary circumstances, this field should not be edited; if the file resides on a labeled tape, and the database value for record conversion does not match the value recorded in the tape label, all accesses to the file will be aborted. <i>dispose=</i> Volume disposition. <i>dispose</i> controls when volumes leave

the volumeset and what happens when they do. `dispose` may be set to `erase` and/or `retain`. If `erase` is specified, volumes that leave the volumeset (when it is truncated or scratched) are erased before they can be reallocated. If `retain` is specified the volumeset is never truncated. (Truncation occurs when the first file on a volumeset is overwritten and the new file is not large enough to span all the volumes currently in the volumeset.) To select both, specify `dispose=erase&retain`.

`ecnt`= Error count. Set error count as specified.

`finger`=File fingerprint. Set fingerprint field as specified.

`flocation`=

Free volume location. When a volumeset is truncated or scratched, the newly unattached volumes will return to the location assigned to `flocation`. The value given must be defined as a REELibrarian storage site (the command `rlr sites` will produce a list of currently defined storage sites).

`format`=Label Format. `format` may be set to one of `ANSI`, `IBM`, `IBMU`, `TAR`, `CPIO` or `RAW`.

`ftemplate`=

File Name template. The filename template is used to dynamically construct names for files written to the volumeset. The name constructed by the template is limited to seventeen (17) characters. The template consists of constant text and substitution patterns. Substitution patterns have the general form: `@sublen@` where the “@” symbol delimits the beginning and end of the substitution specification. `sub` is a character indicating what to substitute. `len` gives the length in characters of the substituted value. In principle can be any number between 1 and 17, however, for some values of `sub` only a particular value of `len` is reasonable. The following

values are recognized.

@Y4@ - numeric year,

@C2@ - numeric month (Jan = 1),

@E3@ - month name,

@D2@ - day of month,

@J3@ - Julian day,

@W3@ - day of week (Sun = 1),

@H2@ - hour (24 hour clock),

@M2@ - minute,

@S2@ - second,

@F9@ - value assigned `fid` keyword on the `rlvwrite` command,

@U9@ - base name of file assigned to the `if` keyword on the `rlvwrite` command,

@G4@ - file generation,

@V2@ - file version.

Numeric values are truncated on the left. Character values are truncated on the right. If a numeric value is shorter than *len* it is padded on the left with zeros. Example on January 25 1987: `ftemplate=@D2@@E3@@Y2@G@G4@V@V2@`, results in a file name of: `25Jan89G0000V01`

`ftrack`=File Tracking Flag. `yes` causes the catalog to maintain a record for every file written to the volumeset. `no` disables file cataloging. Maintaining file catalogs makes it possible to generate a volumeset table of contents without having to mount a tape. It also improves the efficiency of accessing files on multi-volume volumesets.

`group`=File group. The group (from `/etc/group`) to which the file belongs, up to 12 characters long.

`initialize`=

Volumeset Initialization. `yes` indicates the volumeset requires initialization. The first time each volume is mounted,

REELlibrarian will initialize it. `no` indicates the volumeset contains existing data. The first time each volume is mounted REELlibrarian checks to make sure the Volume Serial Number on the volume label matches the Volume Serial Number in the catalog. Default: `yes`.

`length`=Length of tape in feet. Specifies the length of nine-track or reel media. Default: 2400

Note: For 8mm, 4mm, and QIC media, use the `capacity` keyword.

`offset`=Record offset. The number of bytes reserved at the beginning of each block for additional information. Offset only effects the ANSI tape format.

WARNING: Under ordinary circumstances, this field should not be edited.

`passwd`=Volume Access Password. An optional password which, if specified, must be given before the volumeset can be read or written. Example: `passwd=undertow`.

`pool`= Pool membership. Every volume must belong to a pool. By default, volumes belong to the user's private pool. Volumes may only be assigned to pools owned by the current effective user ID.

Note: The full specification of pool is `userid/pool`. If `userid` is omitted, the current effective user ID is assumed. Only operators may create pools for IDs other than their own.

`rformat`=

Record format. The new record format for the file. `rformat`

rledit(1)

takes the form: *fmt:blen:rlen*. *fmt* is one of:

f - fixed length records;

fb - fixed length, blocked records;

v - variable length records;

vb - variable length, blocked records;

vs - variable length, spanned records;

vbs - variable length, blocked, spanned records.

u - unformatted data.

blen is the block length in bytes. *rlen* is the record length in bytes. Example `rformat=fb:800:80`.

scratch=

Scratch status. Sets volume scratch status (and entire if volume is the first member of the volumeset).

user= File owner. The user ID (from `/etc/passwd`) to which the file belongs, up to 12 characters long.

vaccess=

Volume Access Byte. The character assigned to `vaccess` is written as byte 11 (Volume Accessibility) in the VOL1 label. REELlibrarian attaches no particular significance to the Volume Access byte; control is provided for export of volumes to sites that require certain values. Applies to IBM and ANSI formats only. Default: 0x00.

vexpire=

Volumeset Expiration Date. One of:

I - infinite (never expires),

S - scratch (immediately expired),

RN - expires *N* days after creation,

AN - expires if not accessed in *N* days,

L - expires when all files on the volumeset have expired,

Orotsched - follows the rotation schedule *rotsched* (see

`rlrcreate(1)`,

`Xccyyymmdd` or `Xmm/dd/yy` - expires on given date

`GN` - expire when there are *N* newer generations.

Please note that upon expiration the volumeset is not disbanded until it is scratched. The `rlvscratch` program disbands volumesets. Example: `vexpire=R30`. Default: `S`.

`vmode`=Volumeset Permission Mask. Three octal digits controlling owner, group and others (respectively) permission to the volumeset (similar to the UNIX file mode). Bit 4 controls read access. Bit 2 controls write access. Bit 1 permits viewing of the volume's catalog entry. Example: `vmode=744`. Default: `700`.

Note: The `vmode` keyword is only applicable to the first volume in a volumeset.

`vsn`= Volume Serial Number. The volume serial number (VSN) associated with the volume. The VSN is written in the VOL1 label for IBM and ANSI tape formats. Even unlabeled volumes (IBMU, TAR, CPIO and RAW) must be given a VSN. In general REELlibrarian does not require each volume to have a unique VSN. Up to six characters long.

`slocation`=

Scheduled Volume Location. If a volume's current location (`clocation`) does not equal its scheduled location. The volume will show up on the Volume maintenance report as wanting to move from `clocation` to `slocation`. The value given must be defined as a REELlibrarian storage site (the command `rlr sites` will produce a list of currently defined storage sites).

`type`= Volume Media Type. Media Type is an arbitrary name assigned during REELlibrarian configuration to describe Volume formats supported by devices under REELlibrarian control. For 9-track devices, this is typically the tape density (i.e. 1600, 6250, etc.). For cartridge devices, it is typically the cartridge standard (i.e. QIC11, QIC24, etc.). The given value must be a define Media Type. The command `rlr mtype`

rledit(1)

produces a list of defined Media Types.

The following keywords may only be used by REELlibrarian operators:

`clean`=Volume cleaning count. The number of times the volume has been mounted since it was last cleaned. This value is normally maintained by REELlibrarian.

`clocation`=

Current Volume Location. The value given must be defined as a REELlibrarian storage site (the command `rlr sites` produces a list of currently defined storage sites).

`ctype`=Current Media Type. For initialized volumes, this field contains the Media Type REELlibrarian has verified is the correct media type for the volume. This field is normally maintained by REELlibrarian.

`rack`= Rack Number. The rack number identifying the volume's storage slot. The rack keyword is provided here so operators can submit volumes in a single step. Up to twelve characters long.

Note: If volume acceptance is required, the rack number is normally assigned when the operator accepts the volume.

`vid`= Volume ID. A unique volume identifier. For sites that maintain unique Volume Serial Numbers, `vid` is normally identical to `vsn`. By default, REELlibrarian, assigns each volume a unique volume ID by combining a four character word with a three digit number (Example: `root-496`). Up to twelve characters long.

`maintenance`=

Scheduled Maintenance. `maintenance` may be set to one or more of `move`, `age`, `erase`, `remove` and `clean`. Multiple items are selected by separating them with "&" (i.e. `maint="erase&clean"`). `move` indicates the volume is awaiting movement. `age` causes the volume to wait *N* days after it is scratched before becoming free (*N* is defined by the pool the tape belongs to). `erase` schedules the volume for erasure.

`clean` schedules the volume for cleaning.

Note: Normally the maintenance value is maintained by REELlibrarian.

`remove`=Volume Usage Count. The number of times the volume has been mounted since it entered the catalog. This value is normally maintained by REELlibrarian.

`status`=Volume submission status. Status may be assigned the following integer values.

- 1 - submitted awaiting acceptance.
- 2 - accepted awaiting identification.
- 3 - fully submitted.
- 4 - retrieved awaiting return.

Note: Submission status is normally maintained by REELlibrarian.

See Also `rlpsubmit(1)`, `rlvaccess(1)`, `rlvsubmit(1)`.

Name	rlfedit - modify the catalog entry of a file
Synopsis	<pre>rlfedit keywords fname</pre> <pre>rlfedit vol=volset keywords fname</pre> <pre>rlfedit vol=volset fseq=fsn keywords</pre>
Description	<p>rlfedit modifies the catalog entry of the specified file. A file may be specified by <i>fname</i> only if <i>fname</i> is unique in the library. A file must be specified with <i>fname</i> and the <i>vol=</i> keyword if <i>fname</i> occurs on other volumesets in the library. If the file is unnamed, or if <i>fname</i> occurs more than once on the specified volumeset, the file must be specified with the <i>vol=</i> and <i>fseq=</i> keywords.</p> <p>A file may only be edited by its owner. Files on volumesets that are in use cannot be edited.</p>
Options	<p>fname</p> <p>The cataloged name of the file to edit. If <i>fname</i> is not unique in the library, the <i>vol=</i> keyword is required. If <i>fname</i> is not unique on the volumeset, the <i>fseq=</i> keyword is required.</p> <p>Note: You may include file generation and version numbers when specifying <i>fname</i>; refer to the <i>fid=</i> keyword, below, for proper syntax.</p> <p>vol= Volumeset name. The cataloged name of the volumeset, assigned by the volumeset owner. Volumeset names take the form [<i>userid</i>/]<i>vname</i>[:<i>Ggno</i>][:<i>Vvno</i>]. If <i>userid</i> is omitted, the effective user ID is assumed. <i>vname</i> is an arbitrary string up to 12 characters long. <i>Ggno</i> specifies a volumeset generation number. <i>Vvno</i> specifies a volumeset version number.</p> <p>Note: You may not edit this field. Use it for file identification in cases where <i>fname</i> is not unique in the library.</p> <p>fseq= File sequence number. Selects a file to edit by its relative position on the volumeset. <i>fseq=1</i> selects the first file on the</p>

volumeset, `fseq=2` selects the second file, and so on.

Note: You may not edit this field. Use it for file identification in cases where `fname` does not exist, or is not unique on the volumeset.

keywords

One or more *keyword=value* assignments. Keywords recognized by this command are listed below.

`fid=` File ID. The new file ID to associate with the specified file. File IDs take the form `fname[:Ggno][:Vvno]`. `fname` is the file name, an arbitrary string up to 17 characters long. `Ggno` is the file generation number. `Vvno` is the file version number. Generation and version numbers are used as subscripts for files with the same `fname`. This is explained below.

`gno` can be an integer or a signed integer. If `gno` is an integer, it references the given generation number. If `gno` is a signed integer, it references the highest existing generation number offset by `gno`. For example, if the highest generation number for file forecast is 10, `:G-1` refers to generation 9 and `:G+1` refers to generation 11. If the generation specification is omitted, the highest existing generation is referenced. If there are no existing generations, generation 0 is selected. The highest allowable generation number is 9999.

`vno` specifies a file version number. Version numbers behave like generation numbers. The highest allowable version number is 99.

WARNING: Under ordinary circumstances, this field should not be edited; if the file resides on a labeled tape, the database value for file ID must match the value recorded on tape in the HDR2 label. If these values do not match, all accesses to the file will be aborted.

`fexpire=`

File expiration date. The new expiration date for the file. One

of:

I - infinite (never expires)

S - scratch (always expires)

RN - expires *N* days after creation (maximum: 999)

AN - expires if not accessed in *N* days

Xccyyymmdd **or** Xmm/dd/yy - expires on given date

GN - expire when there are *N* newer generations.

Note: file expiration is only considered if volumeset expiration is set to L.

rformat=

Record format. The new record format for the file. rformat takes the form: *fmt:blen:rlen*. *fmt* is one of:

f - fixed length records;

fb - fixed length, blocked records;

v - variable length records;

vb - variable length, blocked records;

vs - variable length, spanned records;

vbs - variable length, blocked, spanned records.

u - unformatted data.

blen is the block length in bytes. *rlen* is the record length in bytes. Example rformat=fb:800:80.

offset=Record offset. The number of bytes reserved at the beginning of each block for additional information. Offset only effects the ANSI tape format.

WARNING: Under ordinary circumstances, this field should not be edited; if the file resides on a labeled tape, and the database value for file offset does not match the value recorded in the tape label, all accesses to the file will be aborted.

conv= Record conversion specification. This value controls the

conversion of records to and from tape. `conv` may be one of: `text` for text records; `etext` for EBCDIC text records, `data` for fixed length ASCII or binary data records; `edata` for fixed length EBCDIC data records.

WARNING: Under ordinary circumstances, this field should not be edited; if the file resides on a labeled tape, and the database value for record conversion does not match the value recorded in the tape label, all accesses to the file will be aborted.

`app=` Application prefix. This prefix is used with file reports to limit selections to those with the specified `app=` value. `app` may be up to 12 characters long.

`fcomment=`

File comment. A comment to associate with the file, up to 40 characters long. If the comment includes spaces, it must be enclosed in quotes.

`fpasswd=`

File access password. A password to associate with the file, up to 14 characters long. Files with passwords cannot be accessed unless the password is provided.

`user=` File owner. The user ID (from `/etc/passwd`) to which the file belongs, up to 12 characters long.

`group=` File group. The group (from `/etc/group`) to which the file belongs, up to 12 characters long.

`fmode=` File permission mask. Three octal digits that control access to the file. Example: `fmode=744` sets the UNIX file permission string to `rwxr--r--`.

Examples

The following command edits the expiration date of a file. Because the filename `payroll` is not unique in the library, the volumeset on which it resides is specified.

```
rlfedit vol=accounting fexpire=X12/31/95 payroll
```

The following command edits the permission mask of the file to extend read, write, and execute privileges to all users, groups, and

rlfedit(1)

others. Because the file does not have a cataloged name, it is specified with the `fseq=` keyword.

```
rlfedit vol=project37 fseq=3 fmode=777
```

See Also `rlr(1)`, `rledit(1)`, `rlvedit(1)`.

Name `rlfree` - free a previously reserved device

Synopsis `rlfree [psd]`

Description `rlfree` frees devices previously allocated by `rlreserve(1)`.
If `psd` is not specified, all devices under the user's control are freed.

Options `psd` Pseudo Device Name. An arbitrary name assigned during device reservation; used to distinguish between devices when multiple devices are reserved.

See Also `rlreserve(1)`, `rlstatus(1)`.

Notes If the timeout period between `rlreserve` and `rlvmount` expires and you lose your device reserved via a pseudo device name, it is necessary to issue an `rlfree` command to free the pseudo device name before attempting the request again.

rlink(1)

Name	<code>rlink</code> - display the path names for the special files associated with a pseudo device name
Synopsis	<code>rlink [-n] [-r] [<i>psd</i>]</code>
Description	<p><code>rlink</code> without any options displays the rewind and no-rewind special file names associated with the given pseudo device name.</p> <p>With the <code>-n</code> option, <code>rlink</code> displays the no-rewind device name in a format suitable for assignment to a shell variable. Similarly, the <code>-r</code> option displays the rewind device name.</p>
Options	<p><code>-n</code> Display the no-rewind device name in a format suitable for assignment to a shell variable.</p> <p><code>-r</code> Display the rewind device name in a format suitable for assignment to a shell variable.</p> <p><i>psd</i> Pseudo Device Name. An arbitrary name assigned during device reservation; used to distinguish between devices when multiple devices are reserved.</p>
See Also	<code>rlstatus(1)</code> .
Notes	If a device reservation is not made by ADN, <code>rlink</code> will not display anything until there is actually a mounted volume. The reason for this is the actual device assigned to a request can change up until the point where a requested volume is actually mounted.

Name rlopmsg - send a message to the REELlibrarian operator

Synopsis rlopmsg *message*

Description rlopmsg sends the string message to all currently active REELlibrarian operators.

Notes Message can be no larger than 65 characters.

Name	<code>rlpcreate</code> - create a new volume pool
Synopsis	<code>rlpcreate</code> [<i>keywords</i>] <i>pool</i>
Description	<p><code>rlpcreate</code> creates a new volume pool. Pools are used to make volumes available to select users or applications. Any user can create a pool. The creator of a pool specifies what user IDs are permitted to allocate volumes from the pool. By default, only the creator of a pool may allocate pool volumes.</p> <p>The full specification of pool is <i>userid/pool</i>. If <i>userid</i> is omitted, the current effective user ID is assumed. Only operators may create pools for IDs other than their own.</p> <p>The pool <code>private</code> is implicitly created the first time a user submits a volume to REELlibrarian. If a user wishes to designate a set of volumes for exclusive use by an application, or, wishes to make a set of volumes available to a select group of users, it is appropriate to create a new pool to contain the volumes.</p> <p>Often a central pool <code>root/public</code> is provided by the Media Administrator for use by the general user population. Providing volumes centrally saves time spent on submission and retrieval.</p> <p>Allowing users to allocate volumes from another user's pool creates the situation where one person owns a physical volume while another person owns the data it contains. To maintain data security, REELlibrarian does not give the volume owner any special privilege with respect to accessing the volume. Further, the volume owner may not retrieve the volume until it is scratched by the data owner.</p> <p>In general, a user may only retrieve a volume if it belongs to one of their pools. The only way to retrieve a volume from someone else's pool is to have an operator edit the <code>volumeset</code> into a pool owned by the user.</p>
Options	<p><i>pool</i> Volume Pool Name. To reference a pool that belongs to someone else pool is prefixed with the user ID of the pool's owner (i.e. <code>root/public</code>). Up to twelve characters long.</p> <p><i>keywords</i></p> <p>One or more <code>keyword=value</code> assignments. There are no</p>

specified defaults associated with these keywords. Keywords recognized by this command are:

padd= Automatic Volume Addition Flag. If set to *yes*, and a volume allocation request cannot be satisfied by an existing volume, the operator is prompted to provide a “new” volume. If the operator refuses, the allocation request fails. If the operator does provide a new volume, the volume is dynamically added to the catalog. (This keyword is available to operators only). Example: `padd=yes`.

pclean= Pool Cleaning Flag. If *yes*, whenever a pool volume is scratched, REELlibrarian checks to see if the volume’s cleaning count exceeds the cleaning threshold configured for the volume’s Media Type. If the cleaning count is too high, the volume is scheduled for cleaning before being scratched. Example: `pclean=yes`.

gacc= Group Access List. Same as the “*uacc*” keyword only with group IDs. Example: `gacc=ANY, -guest`.

phold= Volume Hold Period. Set to an integer representing the number of days to hold a volume in limbo between when a user scratches it and when it is available for allocation again. This feature allows users to change their mind about scratching a volume. Example: `phold=10`.

uacc= User Access List. A comma-separated list of user IDs identifying who can allocate volumes from the pool. The special value *ANY* specifies any user. User IDs may be explicitly excluded by prefixing them with “-”. Example: `uacc=ANY, -hacker`.

premove=

Pool Removal Flag. If *yes*, whenever a pool volume is scratched, REELlibrarian checks to see if the volume’s usage count exceeds the usage threshold configured for the volume’s Media Type. If the usage count is too high, the volume is scheduled for removal from the catalog Example: `premove=yes`.

See Also `rlpedit(1)`, `rlpdelete(1)`, `rlr(1)`.

rlpdelete(1)

Name	<code>rlpdelete</code> - delete a volume pool
Synopsis	<code>rlpdelete pool</code>
Description	<code>rlpdelete</code> deletes the specified pool. A pool cannot be deleted if it contains volumes. Only the pool owner is permitted to delete it.
Options	<i>pool</i> Volume Pool Name. To reference a pool that belongs to someone else pool is prefixed with the user ID of the pool's owner (i.e. <code>root/public</code>).
See Also	<code>rlpcreate(1)</code> , <code>rlpedit(1)</code> , <code>rlr(1)</code> .

Name	<code>rlpedit</code> - edit a volume pool
Synopsis	<code>rlpedit</code> [<i>keywords</i>] <i>pool</i>
Description	<code>rlpedit</code> modifies the attributes of the specified pool.
Options	<p><i>pool</i> Volume Pool Name. To reference a pool that belongs to someone else pool is prefixed with the userid of the pool's owner (i.e. <code>root/public</code>).</p> <p>keywords</p> <p>One or more keyword=value assignments. There are no specified defaults associated with these keywords. Keywords recognized by this command are:</p> <p>padd= Automatic Volume Addition Flag. If <code>yes</code>, and a volume allocation request cannot be satisfied by an existing volume, the operator is prompted to provide a "new" volume. If the operator refuses, the allocation request fails. If the operator does provide a new volume, the volume is dynamically added to the catalog. (This keyword is available to operators only). Example: <code>padd=yes</code>.</p> <p>pclean= Pool Cleaning Flag. If <code>yes</code>, whenever a pool volume is scratched, REELlibrarian checks to see if the volume's cleaning count exceeds the cleaning threshold configured for the volume's Media Type. If the cleaning count is too high, the volume is scheduled for cleaning before being scratched. Example: <code>pclean=yes</code></p> <p>agacc=A comma separated list of group IDs to add to the group access list.</p> <p>rgacc=A comma separated list of group IDs to remove from the group access list.</p> <p>phold=Volume Hold Period. Set to an integer representing the number of days to hold a volume in limbo between when a user scratches it and when it is available for allocation again. This feature allows users to change their mind about scratching a volume. Example: <code>phold=10</code></p> <p>auacc=A comma-separated list of user IDs to add to the user access</p>

rlpedit(1)

list.

ruacc=A comma-separated list of user IDs to remove from the user access list

premove=

Pool Removal Flag. If yes, whenever a pool volume is scratched, REELlibrarian checks to see if the volume's usage count exceeds the usage threshold configured for the volume's Media Type. If the usage count is too high, the volume is scheduled for removal from the catalog Example:

premove=yes

See Also rlpcreate(1), rlpdelete(1), rlr(1).

Name	rlretrieve - initiate retrieval of a scratch volume
Synopsis	rlretrieve <i>vid</i> [<i>nvol=n</i>]
Description	<p>rlretrieve assigns the named volume a receipt number. To complete retrieval, present the receipt number to the operator in exchange for the volume.</p> <p>If REELlibrarian is not configured to require volume acceptance, or, <i>vid</i> has not yet been accepted, the corresponding volume record is deleted from the catalog, and retrieval is complete. A receipt number is not generated.</p> <p>If acceptance is required and <i>vid</i> has been accepted, REELlibrarian assigns the volume a receipt number.</p> <p>Only the owner may retrieve a volume.</p>
Options	<p><i>vid</i> Volume ID. A unique identifier assigned to each volume when it is submitted.</p> <p><i>nvol=</i> Number of tapes. Specifies the number of tapes to be selected. Default: 1.</p>
See Also	rlpsubmit(1).

rlpsubmit(1)

Name	rlpsubmit - submit a scratch volume to REELlibrarian
Synopsis	rlpsubmit <i>keywords</i>
Description	<p>rlpsubmit initiates submission of a scratch volume to REELlibrarian. Scratch volumes are volumes that do not belong to a volumeset. Scratch volumes join volumesets either when allocated by the rlvcreate(1) command or when they are used to extend an existing volumeset when rlvwrite(1) drafts a new volume from a pool.</p> <p>The rlpsubmit keywords describe attributes of the volume such as length, media type, etc. If a keyword is not given, the corresponding attribute is given a default value from the Default Definition Record that matches the indicated media type.</p> <p>Upon successful completion, rlpsubmit displays the volume ID assigned to the volume. The volume ID is a unique name for the volume within the database, it should be written on an external label affixed to the volume.</p> <p>After rlpsubmit completes, the catalog contains a record for the volume. If REELlibrarian is configured to require volume acceptance and/or volume identification, the volume is not available for use until it is handed over to an operator who completes the submission process.</p>
Options	<p>keywords</p> <p>One or more keyword=value assignments. There are no specified defaults associated with the following keywords. Keywords recognized by this command are:</p> <p>app= Application prefix. Used with the volume and file listing reports, app limits selection to items that begin with the given string. Example: app=89</p> <p>capacity=</p> <p>Volume capacity in Mbytes. Specifies the volume capacity for cartridge media types.</p> <p>Note: For nine-track (or reel) media, use the length keyword.</p>

`ecnt`= Error count. Set error count as specified.

`finger`=File fingerprint. Set fingerprint field as specified.

`flocation`=

Free volume location. When a volumeset is truncated or scratched, the newly unattached volumes will return to the location assigned to `flocation`. The value given must be defined as a REELibrarian storage site (the command `rlr sites` will produce a list of currently defined storage sites).

`length`=Length of tape in feet. Specifies the length of nine-track or reel media. Default: 2400

Note: For 8mm, 4mm, and QIC media, use the `capacity` keyword.

`nvols`= Number of tapes. Specifies the number of tapes to submit to the library. The `vid`, `rack` and `vsns` values (if given) are incremented for each repetition. Default: 1.

`vsns`= Volume Serial Number. The volume serial number (VSN) associated with the volume. The VSN is written in the VOL1 label for IBM and ANSI tape formats. Even unlabeled volumes (IBMU, TAR, CPIO and RAW) must be given a VSN. In general REELibrarian does not require each volume to have a unique VSN. Up to six characters long. The special value `vms` means that this value is selected by the Vault Management System.

`type`= Volume Media Type. Media type is an arbitrary name assigned during REELibrarian configuration to describe Volume formats supported by devices under REELibrarian control. For 9-track devices, this is typically the tape density (i.e. 1600, 6250, etc.). For cartridge devices, it is typically the cartridge standard (i.e. QIC11, QIC24, etc.). The given value must be a defined Media Type. The command `rlr mtype` produces a list of defined Media Types.

The following keywords may only be used by REELibrarian operators:

`clean`=Volume cleaning count. The number of times the volume has

been mounted since it was last cleaned. This value is normally maintained by REELlibrarian.

`clocation=`

Current Volume Location. The value given must be defined as a REELlibrarian storage site (the command `rlr sites` produces a list of currently defined storage sites).

`ctype=`Current Media Type. For initialized volumes, this field contains the media type REELlibrarian has verified is the correct media type for the volume. This field is normally maintained by REELlibrarian.

`rack=` Rack Number. The rack number identifying the volume's storage slot. The rack keyword is provided here so operators can submit volumes in a single step. Up to twelve characters long. The special value `vms` means that this value is selected by the Vault Management System.

Note: If volume acceptance is required, the rack number is normally assigned when the operator accepts the volume.

`vid=` Volume ID. A unique volume identifier. For sites that maintain unique Volume Serial Numbers, `vid` is normally identical to `vsn`. By default, REELlibrarian assigns each volume a unique volume ID by combining a four character word with a three digit number (Example: `root-496`). Up to twelve characters long. The special value `vms` means that this value is selected by the Vault Management System. The VID may not begin with a period (".") or contain a colon (":").

`maintenance=`

Scheduled Maintenance. `maintenance` may be set to one or more of `move`, `age`, `erase`, `remove` and `clean`. Multiple items are selected by separating them with "&" (i.e. `maint="erase&clean"`). `move` indicates the volume is awaiting movement. `age` causes the volume to wait *N* days after it is scratched before becoming free (*N* is defined by the pool the tape belongs to). `erase` schedules the volume for erasure.

`clean` schedules the volume for cleaning.

Note: Normally the maintenance value is set by REELlibrarian.

`pool`= Pool membership. Every volume must belong to a pool. By default, volumes belong to the user's private pool. Volumes may only be assigned to pools owned by the current effective user ID.

Note: The full specification of pool is *userid/pool*. If *userid* is omitted, the current effective user ID is assumed. Only operators may create pools for IDs other than their own.

`remove`=Volume Usage Count. The number of times the volume has been mounted since it entered the catalog. This value is normally maintained by REELlibrarian.

`status`=Volume submission status. Status may be one of:

- 1 - submitted awaiting acceptance.
- 2 - accepted awaiting identification.
- 3 - fully submitted.
- 4 - retrieved awaiting return.

Note: Submission status is normally maintained by REELlibrarian.

See Also `rledit(1)`, `rlretrieve(1)`.

Name	rlr - REELlibrarian report generator
Synopsis	rlr <i>keywords</i> { <i>finfo</i> <i>flist</i> <i>pinfo</i> <i>plist</i> <i>pvolumes</i> <i>rinfo</i> <i>rlist</i> <i>sites</i> <i>types</i> <i>vinfo</i> <i>vlist</i> <i>vsflist</i> <i>vslist</i> <i>vsvlist</i> }
Description	rlr generates the requested REELlibrarian report. REELlibrarian user reports are a subset of the REELlibrarian operator reports; the operator reports are listed in the rlr(8) manpage.
Options	<p>keywords</p> <p>Keyword Parameters. One or more keyword=value definitions. Some reports require keywords to identify the item being reported. Listed below are the allowed keywords and their descriptions.</p> <p>app= Application prefix. Used with the volume and file listing reports, app limits selection to items that begin with the given string. Example: app=89 .</p> <p>fid= File ID. Identifies the file for the finfo report.</p> <p>fsect=File section number. Identifies the file by the file section number for the finfo report.</p> <p>fseq= File sequence number. Identifies the file by order in the volumeset for the finfo report.</p> <p>full= Specifies full report format or parsable format (with fields delimited by colons). Default is full=yes which specifies full format; full=no specifies parsable format.</p> <p>old= Displays reports in pre-REELlibrarian 3.3 format (old=yes). Default is no.</p> <p>pool= Pool Name. Used with the pinfo, vinventory, and pvolumes reports. Example: pool=private</p> <p>rotation=</p> <p>Rotation schedule. Used with the rinfo report.</p> <p>vid= Volume ID. Identifies the volume for the vinfo report.</p> <p>volset=Volumeset name. Identifies the volumeset for the vinfo, finfo, vsflist, and vsvlist report. Example:</p>

volset=volset1

The following reports are available:

- dinfo** Displays the configuration for the drive specified with the **adn** keyword. Keywords accepted: **adn** and **full**.
- finfo** Displays the full catalog entry for the tape file identified by the **fid** keyword. Keywords accepted: **fid**, **fseq**, **fsect**, **full**, **volset**.
- flist** Lists all the files owned by the user. Keywords accepted: **app**, **full**.
- rinfo** Lists rotation schedule information. Keywords accepted: **full** and **rotation**.
- rlist** Lists rotation schedules. Keyword accepted: **full**.
- sites** Lists all library sites. Keyword accepted: **full**.
- pinfo** Displays the catalog entry for the pool named with the **pool** keyword. Keywords accepted: **pool**, **full**.
- plist** Lists all pools owned by the user. Keyword accepted: **full**.
- pvolumes**
Lists a summary of volumes currently assigned to the specified pool. Keywords accepted: **full**, **pool**.
- types** Lists all defined media types. Accepted keyword is: **full**.
- vinfo** Displays the catalog entry for the volume named by the **volset** or **vid** keywords. Keywords accepted: **full**, **vid**, **volset**.
- vlist** Lists a summary of all selected volumes. Keyword accepted: **full**.
- vsflist** Lists all files on the volumeset named by the **volset** keyword. Keywords accepted: **full** and **volset**.
- vslist** Lists all volumesets belonging to the user. Keywords accepted: **app**, **full**.
- vsvlist** Reports the member volumes of the volumeset named by the

rlr(1)

volset keyword. Keywords accepted: full and volset.

See Also Chapter 5, *Reports*, *REELlibrarian Master Guide*.

Name	<code>rlrcreate</code> - create a rotation schedule
Synopsis	<code>rlrcreate</code> <i>rotsched</i>
Description	<p><code>rlrcreate</code> creates a rotation schedule named <i>rotsched</i>. A rotation schedule is a list of one or more “location-type-count” triples. Volumesets can be assigned rotation schedules.</p> <p>When a volumeset is created and assigned a rotation schedule, it moves to the first location in the list, remaining there for the time designated by the type-count. After that period has passed, the volumeset then moves to the second location. This process continues until the volumeset has proceeded through the entire list—at which point the volumeset expires.</p> <p><code>rlrcreate</code> conducts a dialog with the user repeating requests asking for the location-type-count specification for each step of the schedule. When the entire schedule has been entered, the user enters nothing at the next prompt to indicate completion.</p> <p>Durations can be specified two ways: <code>R</code> or <code>G</code>. The first format indicates that the count is a fixed number of days the volumeset should stay at the location - example: “<code>onsite R 30</code>”. After the “count” days have occurred, the volumeset moves to the next step of the schedule.</p> <p>The second format indicates that the count represents the number of generations of the volumeset which should be at this step of the schedule. That is, only “count” generations may reside at the step. When a new generation arrives at the step, the oldest generation leaves the step for the next step in the schedule. When the oldest generation leaves the last step, it expires.</p> <p>Please note that you cannot use both <code>R</code> and <code>G</code> types in the same schedule.</p> <p>Consider the following example.</p> <pre>rlrcreate sched1 Enter {G R} :onsite G 1 Enter {G R} :offsite G 2 Enter {G R} :</pre>

rlrcreate(1)

The “`sched1`” schedule has two steps. The first step has one generation residing at the location `onsite`. The second step has two generations residing at the location `offsite`. The chronology is as follows. Generation 1 is born and resides at `onsite` until Generation 2 is born. At that point, Generation 1 moves to `offsite` and Generation 2 takes up residence at `onsite`. When Generation 3 is born, Generation 2 moves to `offsite` to reside with Generation 1 since this step can accommodate two generations. Finally, when Generation 4 is born, Generation 3 moves to `offsite`, and Generation 1 expires since there is only room for two generations at `offsite` - in this case Generation 2 and Generation 3.

The `rlvcreate(1)` command describes how to create multiple generations of the same `volumeset`.

See Also `rlrdelete(1)`.

Name rlrdelete - delete a rotation schedule

Synopsis rlrdelete *rotsched*

Description rlrdelete deletes the named rotation schedule. Any volumesets under the schedule henceforth behave as if they have no rotation schedule.

See Also rlrcreate(1).

rlreserve(1)

Name	<code>rlreserve</code> - reserve one or more devices
Synopsis	<code>rlreserve</code> [<i>keywords</i>] [[<i>as</i>] <i>psd</i> [, <i>psd</i> ...]]
Description	<p><code>rlreserve</code> asks the resource manager to reserve one or more devices. When <code>rlreserve</code> completes, the requested devices are available for use. If the request cannot be satisfied, <code>rlreserve</code> prints an appropriate error message.</p> <p>To avoid deadlocks, a user who currently has a reserved device may not reserve a second device. Users who require simultaneous access to multiple devices, must request all devices with a single <code>rlreserve</code> command.</p> <p>The type of device to reserve may be specified a variety of ways. <code>adn=<i>adn_list</i></code> allows specification of the exact names of the device(s) to reserve. <code>type=<i>type_list</i></code> is a less specific (and therefore easier to satisfy) request for any device capable of supporting the indicated Media Type(s). <code>volset=<i>volset_list</i></code> requests device(s) compatible with the given volumeset(s). Only one of <code>adn</code>, <code>type</code> or <code>volset</code> may be specified.</p> <p>Unless a device is reserved by <code>adn</code> the actual device assigned to a reservation may change. For example, the second volume of a volumeset may be mounted on a different device while the first volume is rewinding.</p>
Options	<p>[<i>psd</i>[,<i>psd</i>...]]</p> <p>Pseudo Name List. Comma separated list of Pseudo Device Names. Reservation requests for more than one device must provide each device a Pseudo Device Name. Pseudo Device Names are used to distinguish the different devices after they are reserved. If no Pseudo Device Name is given on a single device reservation request, the name “default” is assumed. Each name can be up to twelve characters long.</p> <p>keywords</p> <p>One or more keyword=value assignments. Keywords recognized by this command are:</p> <p><code>adn=</code> ADN List. Comma separated list of ADNs to reserve. There is</p>

no specified default.

`type=` Media Type List. Comma separated list of Media Types to reserve. There is no specified default.

`volset=` Volumeset List. Comma separated list of volumeset names. The resource allocator checks the Media Type associated of each volumeset and reserves a compatible device. There is no specified default.

`mach=` Machine. Use a drive on the designated machine. If the machine is unspecified and the user's machine has an appropriate drive, then the request is, by default, issued for service by the user's machine. If there is no drive available on the user's machine, then the request will fail and an error message is returned. A value of any will issue a request for any appropriate drive in the network.

`queue=` Reservation Queue Flag. If `yes`, the device reservation request is queued if it cannot be satisfied immediately. The default is to return an error.

`priority=`

Reservation Queue Priority. An integer value from 0 to 9 used to order reservation requests in the queue. Lower priority requests are serviced first. The default priority is 5.

`idle=` Idle Device Flag. By default, if a reserved device is idle for a configurable length of time, REELlibrarian cancels the reservation and make the device available for use by others. Setting `idle` to `yes` prevents REELlibrarian from canceling a reservation due to excessive idle time. Default: `no`.

Note: The timeout period between `rreserve` and `rvmount` is short; therefore setting `idle=yes` is recommended.

See Also `rfree(1)`, `rlstatus(1)`, `rvmount(1)`.

Name rlstatus - display device reservation status

Synopsis rlstatus

Description If a user has a queued resource request, `rlstatus` displays the resource request queue. If a user has reserved devices, `rlstatus` displays information about the devices under the user's control.

The report has two sections, Request Q and Device Status.

Request Q columns:

UID The user's name.

Prio The request's priority - a digit '0' through '9'. Lower digits have higher priority.

Key The request's key name - usually this is the same as the user's name. It can be different, see the `rlvaccess(1)` command description.

Machine The request originated on the named computer.

Format The media type involved in the request.

Adn If a specific drive was requested, it is listed here.

Psd The pseudonym assigned to the drive - see the `rlvaccess(1)` command description.

Device Status columns:

Pseudo The pseudonym assigned to the drive - see the `rlvaccess(1)` command description.

Type The media type involved in the request.

Stat The status indicator.

`vrfy` - the mounted volume is being electronically identified.

`on` - a volume is mounted but not under user control.

`off` - the drive is off-line; there is not a mounted volume.

`dchk` - the drive is undergoing a density check.

`user` - the mounted volume is under user control.

`rew` - the mounted volume is being rewound prior to being

unmounted.

prem - a volume has been premounted - it is unknown the volume's disposition for write protection.

wprem - a volume has been premounted for reading or writing.

rprem - a volume has been premounted for reading only.

oct1 - the drive is about to enter the off state.

VID The volume ID currently mounted on the drive.

Volumeset

The name of the volumeset in use.

See Also rlvaccess(1), rlvrelease(1).

rlunq(1)

Name	rlunq - unqueue REELlibrarian requests
Synopsis	rlunq [key= <i>keyname</i>] [user= <i>uname</i>]
Description	rlunq cancels the outstanding user request. User's can review their outstanding requests with the rlstatus(1) command.
Options	<p>key= The keyname under which the request was made. Usually this is the user's name and so, rlunq assumes the same unless this keyword is specified.</p> <p>user= This keyword directs the program to work with the request associated with the specified user name. This keyword can only be used by operators. The default is the current effective user ID.</p>
See Also	rlstatus(1).

Name	<code>rlvaccess</code> - initiate access to a volumeset
Synopsis	<code>rlvaccess</code> [<i>keywords</i>] <i>volset</i> [[<i>on</i>] <i>psd</i>]
Description	<p><code>rlvaccess</code> initiates access to volumeset <i>volset</i> on device <i>psd</i>. If no device is reserved as <i>psd</i>, <code>rlvaccess</code> reserves one.</p> <p>Accessing a volumeset does not immediately cause a mount, it just confirms the volumeset is not locked (in use elsewhere), the user has the necessary permission, and a suitable device is available. A mount request is generated the first time the volumeset is read or written.</p>
Options	<p><i>volset</i> Volumeset Name. Either the name given to the volumeset when it was submitted or '.' followed by the volume ID of the first volume in the volumeset.</p> <p><i>psd</i> Pseudo Device Name. An arbitrary name assigned during device reservation; used to distinguish between devices when multiple devices are reserved.</p> <p><i>keywords</i></p> <p>One or more keyword=value assignments. Keywords recognized by this command are:</p> <p><i>machine</i>=</p> <p>Machine. Use a drive on the designated machine. If the machine is unspecified and the user's machine has an appropriate drive, then the request is, by default, issued for service by the user's machine. If there is no drive available on the user's machine, then the request will fail and an error message is returned. A value of <i>any</i> will issue a request for any appropriate drive in the network.</p> <p><i>priority</i>=</p> <p>Reservation Queue Priority. An integer value from 0 to 9 used to order reservation requests in the queue. Lower priority requests are serviced first. The default priority is 5.</p> <p><i>queue</i>=Reservation Queue Flag. If <i>yes</i>, the device reservation request is queued if it cannot be satisfied immediately. The default is to return an error.</p> <p><i>passwd</i>=Volume Access Password. An optional password which, if</p>

rlvaccess(1)

specified, must be given before the volumeset can be read or written. Example: `passwd=undertow`. There is no specified default.

`write`=Write Access. If `yes`, a volumeset can be accessed in write mode. Default is `no`.

See Also `rlvcreate(1)`, `rlvrelease(1)`.

Name	<code>rlvcreate</code> - create a volumeset
Synopsis	<code>rlvcreate</code> [<i>keywords</i>] [<i>volset</i>]
Description	<p><code>rlvcreate</code> allocates a scratch volume from the pool indicated by the <code>pool</code> keyword. The volume is converted into a volumeset with attributes specified by the command keywords. If a keyword is not given, the corresponding attribute is given a default value. See <code>rlvsubmit(1)</code> for a description of how default values are selected.</p> <p>Upon successful completion, <code>rlvcreate</code> displays the volume ID of the allocated volume.</p>
Options	<p><i>volset</i> Volumeset Name. A meaningful name assigned by the volumeset owner. The general form of <i>volset</i> is [<i>userid</i>/]<i>vname</i>[:<i>Gno</i>][:<i>Vvno</i>]. If <i>userid</i> is omitted, the effective user ID is assumed. <i>vname</i> is an arbitrary string up to 12 characters long. <i>Gno</i> specifies a volumeset generation number. <i>Vvno</i> specifies a volumeset version number. Generation and version numbers are used as subscripts for volumesets with the same <i>vname</i>.</p> <p>Note: Only an operator may submit to a <i>userid</i> other than the current effective user ID.</p> <p><i>gno</i> can be an integer or a signed integer. If <i>gno</i> is an integer, it references the given generation number. If <i>gno</i> is a signed integer, it references the highest existing generation number offset by <i>gno</i>. For example, if the highest generation number for volumeset <code>test</code> is 10, <code>:G-1</code> refers to generation 9 and <code>:G+1</code> refers to generation 11. If the generation specification is omitted, the highest existing generation is referenced. If there are no existing generations, generation 0 is selected. The highest allowable generation number is 9999.</p> <p><i>Vvno</i> specifies a volumeset version number. Version numbers behave like generation numbers. The highest allowable version number is 99.</p> <p>If <i>volset</i> is omitted, an unnamed volumeset is created. Unnamed volumesets are referenced by a <code>'.</code> followed by the volume ID of the first volume in the volumeset. For example,</p>

If the first volume has volume ID `lump-992` the volumeset is referenced as `.lump-992`.

keywords

One or more keyword=value assignments. There are no specified defaults associated with these keywords except where noted. Keywords recognized by this command are:

`app=` Application prefix. This prefix is used with file reports to limit selections to those with the specified `app=` value. `app` may be up to 12 characters long.

`capacity=`

Volume capacity in Mbytes. Specifies the volume capacity for cartridge media types.

`conv=` Record conversion specification. `conv` controls conversion of records to/from tape. `conv` may be set to one of `text` for text records, `etext` for EBCDIC text records, `data` for fixed length ASCII or binary data records, `edata` for fixed length EBCDIC data records.

`dispose=`

Volume disposition. `dispose` controls when volumes leave the volumeset and what happens when they do. `dispose` may be set to `erase` and/or `retain`. If `erase` is specified, volumes that leave the volumeset (when it is truncated or scratched) are erased before they can be reallocated. If `retain` is specified the volumeset is never truncated. (Truncation occurs when the first file on a volumeset is overwritten and the new file is not large enough to span all the volumes currently in the volumeset.) To select both, specify `dispose="erase&retain"` (include the quotation marks).

`ecnt=` Error count. Set error count as specified.

`finger=` File fingerprint. Set fingerprint field as specified.

`flocation=`

Free volume location. When a volumeset is truncated or scratched, the newly unattached volumes will return to the location assigned to `flocation`. The value given must be defined as a REELlibrarian storage site (the command `rlr`

sites will produce a list of currently defined storage sites).
Up to twelve characters long.

`format=Label Format`. `format` may be set to one of ANSI, IBM, IBMU, TAR, CPIO or RAW.

`ftemplate=`

File Name template. The filename template is used to dynamically construct names for files written to the volumeset. The name constructed by the template is limited to seventeen (17) characters. The template consists of constant text and substitution patterns. Substitution patterns have the general form: `@sublen@` where the “@” symbol delimits the beginning and end of the substitution specification. `sub` is a character indicating what to substitute. `len` gives the length in characters of the substituted value. In principle can be any number between 1 and 17, however, for some values of `sub` only a particular value of `len` is reasonable. The following values are recognized.

@Y4@ - numeric year,

@C2@ - numeric month (Jan = 1),

@E3@ - month name,

@D2@ - day of month,

@J3@ - Julian day,

@W3@ - day of week (Sun = 1),

@H2@ - hour (24 hour clock),

@M2@ - minute,

@S2@ - second,

@F9@ - value assigned `fid` keyword on the `rlvwrite` command,

@U9@ - base name of file assigned to the `if` keyword on the

rlvcreate(1)

rlvwrite command,
@G4@ - file generation,
@V2@ - file version.

Numeric values are truncated on the left. Character values are truncated on the right. If a numeric value is shorter than *len* it is padded on the left with zeros. Example on January 25 1987: `ftemplate=@D2@@E3@@Y2@G@G4@V@V2@`, results in a file name of: 25Jan89G0000V01

`ftrack`=File Tracking Flag. `yes` causes the catalog to maintain a record for every file written to the volumeset. `no` disables file cataloging. Maintaining file catalogs makes it possible to generate a volumeset table of contents without having to mount a tape. It also improves the efficiency of accessing files on multi-volume volumesets.

`length`=Length of tape in feet. Specifies the length of 9-track media.
Default: 2400

Note: For 8mm, 4mm, and QIC media, use the `capacity` keyword.

`offset`=Record offset. The number of bytes reserved at the beginning of each block for additional information. Offset only effects the ANSI tape format.

`passwd`=Volume Access Password. An optional password which, if specified, must be given before the volumeset can be read or written. Up to fourteen characters long. Example:
`password=undertow`.

`pool`= Pool Source. The name of the pool from which to draft tapes into the volumeset. Default: the user's private pool is used:
`username/private`.

`psd`= Use a scratch tape located in the stack on the device identified by the provided pseudo device name.

`rformat`=

Record Format. `rformat` has the general form:

fmt:blen:rlen. Where *fmt* is the record format:

f - fixed length records;

fb - fixed length, blocked records;

v - variable length records;

vb - variable length, blocked records;

vs - variable length, spanned records;

vbs - variable length, blocked, spanned records.

u - unformatted data.

blen is the block length in bytes. *rlen* is the record length in bytes. Example *rformat=fb:800:80*. Default: u.

scratch=

Scratch status. Sets volume scratch status (and entire *volumeset* if volume is the first member of the *volumeset*).

slocation=

Scheduled Volume Location. If a volume's current location (*clocation*) does not equal its scheduled location, the volume will show up on the Volume maintenance report as wanting to move from *clocation* to *slocation*. The value given must be defined as a REELlibrarian storage site (the command *rlr sites* will produce a list of currently defined storage sites). Up to twelve characters long.

type= Volume Media Type. Media type is an arbitrary name assigned during REELlibrarian configuration to describe volume formats supported by devices under REELlibrarian control. For 9-track devices, this is typically the tape density (i.e. 1600, 6250, etc.). For cartridge devices, it is typically the cartridge standard (i.e. QIC11, QIC24, etc.). The given value must be a defined media type. The command *rlr mtype* produces a list of currently defined media types. Up to eight characters long. Default: configurable (the first site in the *rlconfig* site list).

vaccess=

Volume Access Byte. The character assigned to *vaccess* is written as byte 11 (Volume Accessibility) in the VOL1 label.

rlvcreate(1)

REELlibrarian attaches no particular significance to the Volume Access byte; control is provided for export of volumes to sites that require certain values. Applies to IBM and ANSI formats only. Default: 0x00.

vcomment=

Volumeset comment. A comment about the volumeset. If the comment includes spaces, it must be enclosed in quotes. Up to forty characters long. Example: vcom="my favorite volumeset"

vexpire=

Volumeset Expiration Date. One of:

I - infinite (never expires),

S - scratch (immediately expired),

RN - expires *N* days after creation,

AN - expires if not accessed in *N* days,

L - expires when all files on the volumeset have expired,

Orotsched - follows the rotation schedule *rotsched* (see `rlrcreate(1)`),

Xccyyymmdd or *Xmm/dd/yy* - expires on given date

GN - expire when there are *N* newer generations.

Please note that upon expiration the volumeset is not disbanded until it is scratched. The `rlvscratch` program disbands volumesets. Example: `vexpire=R30`. Default: S.

vmode=Volumeset Permission Mask. Three octal digits controlling owner, group and others (respectively) permission to the volumeset (similar to the UNIX file mode). Bit 4 controls read access. Bit 2 controls write access. Bit 1 permits viewing of the volume's catalog entry. Example: `vmode=744`. Default: 700.

The following keywords may only be used by REELlibrarian operators:

group=Volumeset Group. The group (from `/etc/group`) to which the volumeset belongs. Up to twelve characters long. Default:

the current effective group ID.

`initialize=`

Volumeset Initialization. `yes` indicates the volumeset requires initialization. The first time each volume is mounted, REELlibrarian will initialize it. `no` indicates the volumeset contains existing data. The first time each volume is mounted REELlibrarian checks to make sure the Volume Serial Number on the volume label matches the Volume Serial Number in the catalog. Default: `yes`.

`maintenance=`

Scheduled Maintenance. Maintenance May be set to one or more of `move`, `age`, `erase`, `remove` and `clean`. Multiple items are selected by separating them with “&” (i.e. `maint=“erase&clean”`). `move` indicates the volume is awaiting movement. `age` causes the volume to wait *N* days after it is scratched before becoming free (*N* is defined by the pool the tape belongs to). `erase` schedules the volume for erasure. `clean` schedules the volume for cleaning. Default: `no maintenance`.

Note: Normally the maintenance value is maintained by REELlibrarian.

`user=` Volumeset Owner. The user ID (from `/etc/passwd`) to which the volumeset belongs. Up to twelve characters long. Default: the current effective user ID.

See Also `rlvaccess(1)`, `rlvscratch(1)`, `rlvsubmit(1)`.

rlvdisplay(1)

Name	<code>rlvdisplay</code> - display information about an accessed volumeset
Synopsis	<code>rlvdisplay rep[ort]={info volumes toc scan labels {fseq=fseq fid=fid}} [ADN]</code>
Description	<code>rlvdisplay</code> produces a variety of reports about the volumeset currently accessed (<code>rlvaccess(1)</code>). Do not interrupt; if the process is killed, <code>rlvrelease</code> the referenced volumeset.
Options	<p><i>ADN</i> Assigned Device Name. When REELibrarian is installed, each device is assigned a name. This name is used to distinguish between devices on the system. If no name is specified, <code>drive1</code> is assumed.</p> <p><code>rep[ort]=</code></p> <p>Report Name. Selects which report to generate. One of:</p> <ul style="list-style-type: none"><code>info</code> - displays volumeset default values.<code>volumes</code> - lists the volumes that comprise the volumeset.<code>toc</code> - lists a table of contents for files on the volumeset. Only files that have been previously read or written are included.<code>scan</code> - provides a table of contents like <code>toc</code> but <code>scan</code> actually mounts and scans each volume until the end of the volumeset is reached.<code>labels</code> - displays label information for a particular volumeset file. Either <code>fseq</code> or <code>fid</code> must be given to identify which file to display. <p><code>info</code> Displays volumeset default values.</p> <p><code>labels</code> Displays label information for a particular volumeset file. Either <code>fid=</code> or <code>fseq=</code> keywords must be given to identify the file.</p> <p><code>scan</code> Provides a table of contents like <code>toc</code> but <code>scan</code> actually mounts and scans each volume until the end of the volumeset is reached.</p> <p><code>toc</code> Lists a table of contents for files on the volumeset. Only files that have been previously read or written are included.</p> <p><code>volumes</code> Lists the volumes that comprise the volumeset.</p> <p><code>fseq=</code> File Sequence Number. Selects a file by its relative position on</p>

the volumeset. `fseq=1` selects the first file, `fseq=2` selects the second, etc.

`fid=` File ID. Specifies a tape file name. The assigned value has the general form: `fname[:Ggno][:Vvno]`. `fname` is the file name (or File ID), an arbitrary string up to 17 characters long. `Ggno` gives a file generation number. `Vvno` gives a file version number. Generation and version numbers are used as subscripts for files with the same `fname`.

Both `gno` and `vno` are non-negative integer values. The maximum `gno` is 9999. The maximum `vno` is 99.

See Also `rlr(1)`.

rlvedit(1)

Name `rlvedit` - modify the catalog entries for volumes in a volumeset

Synopsis `rlvedit [keywords] volset`

Description `rlvedit` modifies the catalog entry associated with each volume in the specified volumeset to reflect the given keyword assignments.

Only the *volset* owner may edit the volumeset.

Options *volset* Volumeset Name. Either the name given to the volumeset when it was submitted or '.' followed by the volume ID of the first volume in the volumeset.

keywords

One or more keyword=value assignments. The defaults associated with the following keywords are the current volume settings. Keywords recognized by this command are:

app= Application prefix. This prefix is used with file reports to limit selections to those with the specified `app=` value. `app` may be up to 12 characters long.

capacity=

Volume capacity in Mbytes. Specifies the volume capacity for cartridge media types.

Note: for 9-track media, use the `length` keyword.

convert=

Record conversion specification. `convert` controls conversion of records to/from tape. `convert` may be set to one of: `text` for text records. `etext` for EBCDIC text records. `data` for fixed length ASCII or binary data records. `edata` for fixed length EBCDIC data records.

dispose=

Volume disposition. `dispose` controls when volumes leave the volumeset and what happens when they do. `dispose` may be set to `erase` and/or `retain`. If `erase` is specified, volumes that leave the volumeset (when it is truncated or scratched) are erased before they can be reallocated. If `retain` is specified the volumeset is never truncated.

(Truncation occurs when the first file on a volumeset is overwritten and the new file is not large enough to span all the volumes currently in the volumeset.) To select both, specify `dispose="erase&retain"` (include the quotation marks).

`ecnt=` Error count. Set error count as specified.

`finger=`File fingerprint. Set fingerprint field as specified.

`flocation=`

Free volume location. When a volumeset is truncated or scratched, the newly unattached volumes will return to the location assigned to `flocation`. The value given must be defined as a REELibrarian storage site (the command `rlr sites` will produce a list of currently defined storage sites). Up to twelve characters long.

`format=`Label Format. `format` may be set to one of ANSI, IBM, IBMU, TAR, CPIO or RAW.

`ftemplate=`

File Name template. The filename template is used to dynamically construct names for files written to the volumeset. The name constructed by the template is limited to seventeen (17) characters. The template consists of constant text and substitution patterns. Substitution patterns have the general form: `@sublen@` where the "@" symbol delimits the beginning and end of the substitution specification. `sub` is a character indicating what to substitute. `len` gives the length in characters of the substituted value. In principle can be any number between 1 and 17, however, for some values of `sub` only a particular value of `len` is reasonable. The following

values are recognized.

@Y4@ - numeric year,

@C2@ - numeric month (Jan = 1),

@E3@ - month name,

@D2@ - day of month,

@J3@ - Julian day,

@W3@ - day of week (Sun = 1),

@H2@ - hour (24 hour clock),

@M2@ - minute,

@S2@ - second,

@F9@ - value assigned `fid` keyword on the `rlvwrite` command,

@U9@ - base name of file assigned to the `if` keyword on the `rlvwrite` command,

@G4@ - file generation,

@V2@ - file version.

Numeric values are truncated on the left. Character values are truncated on the right. If a numeric value is shorter than *len* it is padded on the left with zeros. Example on January 25 1987: `ftemplate=@D2@@E3@@Y2@G@G4@V@V2@`, results in a file name of: 25Jan89G0000V01

`ftrack`=File Tracking Flag. `yes` causes the catalog to maintain a record for every file written to the `volumeset`. `no` disables file cataloging. Maintaining file catalogs makes it possible to generate a `volumeset` table of contents without having to mount a tape. It also improves the efficiency of accessing files on multi-volume `volumesets`.

`length`=Length of tape in feet. Specifies the length of 9-track media. Default: 2400

Note: For 8mm, 4mm, and QIC media, use the `capacity` keyword.

`offset`=Record offset. The number of bytes reserved at the beginning of each block for additional information. Offset only effects the ANSI tape format.

`passwd`=Volume Access Password. An optional password which, if specified, must be given before the volumeset can be read or written. Up to fourteen characters long. If `passwd` is specified with no value, the existing password is deleted.
Example: `password=undertow`.

`pool`= Source Pool. Volumesets acquire additional tapes from their source pool. Use this keyword to move volumesets between pools.

Note: The full specification of pool is `userid/pool`. If `userid` is omitted, the current effective user ID is assumed. Only operators may create pools for IDs other than their own.

`rformat`=

Record Format. `rformat` has the general form:
`fmt:blen:rlen`. Where `fmt` is the record format: `f` - fixed length records; `fb` - fixed length, blocked records; `v` - variable length records; `vb` - variable length, blocked records; `vs` - variable length, spanned records; `vbs` - variable length, blocked, spanned records. `u` - unformatted data. `blen` is the block length in bytes. `rlen` is the record length in bytes.
Example: `rformat=fb:800:80`.

`scratch`=

Scratch status. Sets scratch status for all volumeset members.

`slocation`=

Scheduled Volume Location. If a volume's current location (`clocation`) does not equal its scheduled location. The volume will show up on the Volume maintenance report as wanting to move from `clocation` to `slocation`. The value given must be defined as a REELlibrarian storage site (the command `rlr sites` will produce a list of currently defined storage sites). Up to twelve characters long.

`type`= Volume Media Type. Media type is an arbitrary name assigned during REELlibrarian configuration to describe volume

formats supported by devices under REELlibrarian control. For 9-track devices, this is typically the tape density (i.e. 1600, 6250, etc.). For cartridge devices, it is typically the cartridge standard (i.e. QIC11, QIC24, etc). The given value must be a defined media type. The command `rlr mtype` produces a list of currently defined media types. Up to eight characters long.

`vaccess=`

Volume Access Byte. The character assigned to `vaccess` is written as byte 11 (Volume Accessibility) in the VOL1 label. REELlibrarian attaches no particular significance to the Volume Access byte; control is provided for export of volumes to sites that do require certain values.

`valloc=`Specify whether a volume is allocated (`valloc=yes`) or not (`valloc=no`). Default is `yes`.

`vcomment=`

Volumeset comment. A comment about the volumeset. If the comment includes spaces, it must be enclosed in quotes. Up to forty characters long. Example: `vcom="my favorite volumeset"`.

`vexpire=`

Volumeset Expiration Date. One of:

I - infinite (never expired),

S - scratch (always expired),

RN - expires *N* days after creation,

AN - expires if not accessed in *N* days,

L - expires when all files on the volumeset have expired,

rotsched - follows the rotation schedule *rotsched* (see `rlrcreate(1)`),

Xccyyymmdd or *Xmm/dd/yy* - expires on given date

GN - expire when there are *N* newer generations.

Example: `vexpire=R30`

`vmode=`Volumeset Permission Mask. Three octal digits controlling

owner, group and others (respectively) permission to the volumeset (similar to the UNIX file mode). Bit 4 controls read access. Bit 2 controls write access. Bit 1 permits viewing of the volume's catalog entry. Example: `vmode=744` allows the volumeset owner all permissions, and the group and others read permission only.

The following keywords may only be used by REELlibrarian operators:

`clean`=Volume cleaning count. The number of times the volume has been mounted since it was last cleaned. This value is normally maintained by REELlibrarian.

`clocation`=

Current Volume Location. The value given must be defined as a REELlibrarian storage site (the command `rlr sites` produces a list of currently defined storage sites). Up to twelve characters long.

`ctype`=Current Media Type. For initialized volumes, this field contains the Media Type REELlibrarian has verified is the correct media type for the volume. This field is normally maintained by REELlibrarian.

`group`=Volumeset Group. The group (from `/etc/group`) to which the volumeset belongs. By default, `group` is the current effective group ID. Up to twelve characters long.

`initialize`=

Volumeset Initialization. `yes` indicates the volumeset requires initialization. The first time each volume is mounted, REELlibrarian will initialize it. `no` indicates the volumeset contains existing data. The first time each volume is mounted REELlibrarian checks to make sure the Volume Serial Number on the volume label matches the Volume Serial Number in the catalog.

`maintenance`=

Scheduled Maintenance. Maintenance May be set to one or more of `move`, `age`, `erase`, `remove` and `clean`. Multiple items are selected by separating them with "&" (i.e. `maint=`

rlvedit(1)

erase&clean). move indicates the volume is awaiting movement. age causes the volume to wait *N* days after it is scratched before becoming free (*N* is defined by the pool the tape belongs to). erase schedules the volume for erasure. clean schedules the volume for cleaning.

Note: Normally the maintenance value is maintained by REELlibrarian.

remove=Volume Usage Count. The number of times the volume has been mounted since it entered the catalog. This value is normally maintained by REELlibrarian.

status=Volume submission status. Status may be assigned the following integer values.

- 1 - submitted awaiting acceptance.
- 2 - accepted awaiting identification.
- 3 - fully submitted.
- 4 - retrieved awaiting return.

Note: Submission status is normally maintained by REELlibrarian.

user= Volumeset Owner. The user ID (from `/etc/passwd`) to which the volumeset belongs. By default user is the current effective user ID. Up to twelve characters long.

See Also rlvaccess(1), rlvretrieve(1), rlvsubmit(1).

Name	<code>rlvmount</code> - mount a volume
Synopsis	<code>rlvmount</code> [<i>keywords</i>] <i>volset</i> [[<i>on</i>] <i>psd</i>]
Description	<p><code>rlvmount</code> asks the operator to mount volume <i>volset</i> on device <i>psd</i>. If no device is reserved as <i>psd</i>, <code>rlvmount</code> reserves one.</p> <p>When a volume is mounted with <code>rlvmount</code>, interaction with the volume takes place directly through the “/dev/...” device names. The <code>rlinq(1)</code> command may be used to display the device names associated with a mounted volume.</p> <p>The preferred way of accessing volumesets is with the commands <code>rlvaccess</code>, <code>rlvread</code>, <code>rlvwrite</code>, and <code>rlvrelease</code>. The commands <code>rlvmount</code> and <code>rlvunmount</code> provide an alternative way to access volumes for applications that need to interact directly with the device special files.</p> <p>Note: When accessing volumes with <code>rlvmount</code> and <code>rlvunmount</code> there is no support for label processing, positioning, record blocking or End-of-Volume processing; REELlibrarian assures that the proper volume is mounted.</p> <p>Note: Because the timeout period between <code>rlreserve</code> and <code>rlvmount</code> requests is short, it is necessary to issue the <code>rlvmount</code> command soon after the <code>rlreserve</code> command. To give yourself more time, include <code>idle=yes</code> in your <code>rlreserve</code> request.</p>
Options	<p><i>volset</i> Volumeset Name. Either the name given to the volumeset when it was submitted or ‘.’ followed by the volume ID of the first volume in the volumeset.</p> <p><i>psd</i> Pseudo Device Name. An arbitrary name assigned during device reservation; used to distinguish between devices when multiple devices are reserved.</p> <p><i>keywords</i></p> <p>One or more keyword=value assignments. Keywords recognized by this command are:</p> <p><i>priority</i>=</p> <p>Reservation Queue Priority. An integer value from 0 to 9 used</p>

rlvmount(1)

to order reservation requests in the queue. Lower priority requests are serviced first. The default priority is 5.

`queue`=Reservation Queue Flag. If `yes`, the device reservation request is queued if it cannot be satisfied immediately. The default is to return an error.

`write`=Write Flag. By default, volumes are mounted read-only. Specifying `write=yes` allows both read and write access.

`passwd`=Volume Access Password. An optional password which, if specified, must be given before the volumeset can be read or written. Example: `passwd=undertow`. There is no specified default.

`force=force=yes` instructs the command to continue even if the volumeset is not expired. Only the owner of a volumeset can use the `force` keyword. Default: `no`.

See Also `rlvunmount(1)`, `rlreserve(1)`.

Name	<code>rlvmove</code> - schedule a volumeset for movement to a different site
Synopsis	<code>rlvmove location=<i>loc</i> <i>volset</i></code>
Description	<p><code>rlvmove</code> schedules all volumes in the named <i>volset</i> for movement to <i>loc</i>. <i>loc</i> must be a site known to REELlibrarian.</p> <p>After the <code>rlvmove</code> command completes, the volumes in the volumeset will show up on the operator's Volume Maintenance Report as scheduled to move from the current location to the new location.</p>
Options	<p><i>volset</i> Volumeset Name. Either the name given to the volumeset when it was submitted or '.' followed by the volume ID of the first volume in the volumeset.</p> <p><i>location</i>=</p> <p>Volume Location. Location for the volume. The value given must be defined as a REELlibrarian storage site (the command <code>rlr sites</code> will produce a list of currently defined storage sites). There is no specified default.</p>
See Also	<code>rlr(1)</code> .

Name	rlvread - read a file from a volumeset
Synopsis	rlvread [<i>keywords</i>] [[on] <i>psd</i>]
Description	<p>rlvread reads a file from the volumeset accessed (rlvaccess(1)) on device <i>psd</i>. The file to read may be identified by its position in the volumeset with keyword <i>fseq</i> or its name with keyword <i>fid</i>. If <i>fseq</i> or <i>fid</i> are not given, the next file on the volumeset is read.</p> <p>Output from rlvread can be directed to a file with keyword <i>of</i> or a FIFO with keyword <i>fifo</i>. By default, rlvread sends its output to stdout.</p> <p>rlvread follows the UNIX convention of exiting with a zero return if successful and a non-zero on failure.</p> <p>rlvread should not be interrupted; if the process is killed, rlvrelease the referenced volumeset.</p>
Options	<p><i>psd</i> Pseudo Device Name. An arbitrary name assigned during device reservation; used to distinguish between devices when multiple devices are reserved.</p> <p>keywords</p> <p>One or more keyword=value assignments. Keywords recognized by this command are:</p> <p><i>fifo</i>= Output FIFO. If <i>fifo</i> is given, rlvread directs its output to the named FIFO. If the FIFO does not exist, it is created. When directing output to a FIFO rlvread puts itself in the background after positioning to the proper file. Default: no.</p> <p><i>of</i>= Output File. If <i>of</i> is given, rlvread directs its output to the named file. If the special value <i>FID</i> is given, output is directed to a file in the current directory whose name is the File Identifier of the file in the volumeset. If the special value <i>SEQ</i> is given, output is directed to a file in the current directory is the integer value of the file sequence number of the file in the volumeset. Default: stdout.</p> <p><i>fid</i>= File ID. Specifies a tape file name. The assigned value has the general form: <i>fname</i>[:<i>Ggn</i>][:<i>Vvno</i>]. <i>fname</i> is the file</p>

name, an arbitrary string up to 17 characters long. *Ggno* gives a file generation number. *Vvno* gives a file version number. Generation and version numbers are used as subscripts for files with the same *vname*.

gno can be an integer or a signed integer. If *gno* is an integer, it references the given generation number. If *gno* is a signed integer, it references the highest existing generation number offset by *gno*. For example, if the highest generation number for file *forecast* is 10, *:G-1* refers to generation 9 and *:G+1* refers to generation 11. If the generation specification is omitted, the highest existing generation is referenced. If there are no existing generations, generation 0 is selected. The highest allowable generation number is 9999.

vno specifies a file version number. Version numbers behave like generation numbers. The highest allowable version number is 99. The default is the next available file.

fseq= File Sequence Number. Selects a file by its relative position on the *volumeset*. *fseq=1* selects the first file, *fseq=2* selects the second, etc. The default is the next available file

fsect=File Section Number. Files that span multiple volumes are broken into sections. Normally, access starts at the first section (i.e. the beginning of the file). The *fsect* keyword may be used with either the *fid* or *fseq* keywords to initiate access at a specific section. Default: 1.

Note: Spanned record formats (*vs* and *vbs*) should not be used with a file section number other than one.

ortype=Output Record Type. Either *fixed* (default) or *variable*. The setting *fixed* has no effect on the output records. Specifying *variable* causes *rlvread* to prefix each output record with the record's length. The length is written as a six byte ASCII integer.

span= Span Flag. By default *rlvread* and *rlvwrite* cross volume boundaries transparently. That is, an I/O operation that encounters End-of-Volume automatically requests the next volume. Setting *span=no* disables transparent volume crossing, the first End-of-Volume encountered is treated like

End-of-File. Default: yes.

rformat=

Record Format. `rformat` has the general form:

`fmt:blen:rlen`. Where `fmt` is the record format: `f` - fixed length records; `fb` - fixed length, blocked records; `v` - variable length records; `vb` - variable length, blocked records; `vs` - variable length, spanned records; `vbs` - variable length, blocked, spanned records. `u` - unformatted data. `blen` is the block length in bytes. `rlen` is the record length in bytes. Example: `rformat=fb:800:80`. There is no specified default.

offset=Record offset. The number of bytes reserved at the beginning of each block for additional information. Offset only effects the ANSI tape format when the HDR2 labels are absent. Default: 0.

The following keywords are optional and an empty value is acceptable:

app= Only expired volumes belonging to the named application are scratched.

conv= Record conversion specification. `conv` controls conversion of records to/from tape. `conv` may be set to one of `text` for text records, `etext` for EBCDIC text records, `data` for fixed length ASCII or binary data records, `edata` for fixed length EBCDIC data records.

faccess=

File Access Byte. The character assigned to `faccess` is written as byte 54 (Accessibility) in the HDR1 label. REELlibrarian attaches no particular significance to the File Access byte; control is provided for export of volumes to sites that require certain values.

fcomment=

File comment. A comment about the File. If the comment includes spaces, it must be enclosed in quotes. Up to forty characters long. Example: `fcomment="my favorite`

File".

`fexpire=`

File Expiration Date. One of:

I - infinite (never expires)

S - scratch (always expires)

R*N* - expires *N* days after creation (maximum: 999)

AN - expires if not accessed in *N* days

L - expire when all files on the volumeset have expired

X*ccyyymmdd* or X*mm/dd/yy* - expires on given date

GN - expire when there are *N* newer generations.

Note: File expiration is only considered if volumeset expiration is set to L. Example: `cdate=R30`.

`force=force=y` instructs the command to continue even if the volumeset is not expired. Only the owner of a volumeset can use the `force` keyword.

`fpasswd=`

File Access Password. An optional password which, if specified, must be given before the file can be read. Example: `fpasswd=undertow`.

See Also `rlvaccess(1)`, `rlvrelease(1)`, `rlvwrite(1)`.

rlvrelease(1)

Name	rlvrelease - terminate access to a volumeset
Synopsis	rlvrelease [<i>psd</i>]
Description	rlvrelease releases a volumeset accessed with <code>rlvaccess(1)</code> . If the corresponding <code>rlvaccess</code> command reserved a device, the device is automatically freed.
Options	<i>psd</i> Pseudo Device Name. An arbitrary name assigned during device reservation; used to distinguish between devices when multiple devices are reserved.
See Also	<code>rlvaccess(1)</code> , <code>rlvread(1)</code> , <code>rlvwrite(1)</code> .

Name	rlvretrieve - initiate volumeset retrieval
Synopsis	rlvretrieve <i>volset</i>
Description	<p>rlvretrieve initiates retrieval of all volumes in a volumeset.</p> <p>On completion, if REELlibrarian is not configured to require volume acceptance, or, the volumes in <i>volset</i> have not yet been accepted, the volume records are deleted from the catalog, and retrieval is complete.</p> <p>If REELlibrarian does require volume acceptance, and, the volumes in the <i>volset</i> have been accepted, REELlibrarian assigns each volume a receipt number. To complete retrieval, the receipt numbers are presented to an operator who completes the process and turns over the volumes.</p>
Options	<p><i>volset</i> Volumeset Name. Either the name given to the volumeset when it was submitted or ' .' followed by the volume ID of the first volume in the volumeset.</p> <p>keywords</p> <p>One or more keyword=value assignments. Keywords recognized by this command are:</p> <p>prompt=If yes, prompts user to verify execution of rlvretrieve process.</p>
See Also	rlvcreate(1), rlvsubmit(1).

rlvscratch(1)

Name rlvscratch - scratch a volumeset

Synopsis rlvscratch {[force={y|n}] *volset* | pool=*pname* | app=*application*}

Description rlvscratch disbands *volset*. All constituent volumes are examined for cleaning or removal processing and then returned to scratch status in their pool. The volumeset owner can scratch a volumeset. If the volumeset has not expired, *force=y* must be specified or rlvscratch will refuse to scratch it. The owner of the pool to which the constituent volumes belong may scratch a volumeset but only if the volumeset has expired.

Options *volset* Volumeset Name. Either the name given to the volumeset when it was submitted or '.' followed by the volume ID of the first volume in the volumeset.

keywords

One or more keyword=value assignments. Keywords recognized by this command are:

force=force=y instructs the command to continue even if the volumeset is not expired. Only the owner of a volumeset can use the *force* keyword. Default: no.

The following keywords are optional and an empty value is acceptable:

pool= All expired volumes in the named pool are scratched. If *pool=all* is used, then all pools are affected. However, default volumesets are not scratched.

Note: The full specification of pool is *userid/pool*. If *userid* is omitted, the current effective user ID is assumed. Only operators may create pools for IDs other than their own.

app= Only expired volumes belonging to the named application are scratched.

See Also rlvcreate(1), rlvsubmit(1).

Name	<code>rlvsubmit</code> - submit a volumeset
Synopsis	<code>rlvsubmit keywords [volset]</code>
Description	<p><code>rlvsubmit</code> initiates submission of a volumeset to REELlibrarian. The <code>rlvsubmit</code> keywords describe attributes of the volumeset such as length, label format, media type, etc. If a keyword is not given, the corresponding attribute is given a default value. If the volumeset is a new Generation/Version, of an existing volumeset, default values are taken from the highest existing Generation/Version. Otherwise, REELlibrarian looks for the closest matching Default Definition Record provided by the user. If the user has not provided a Default Definition Record, REELlibrarian looks for Default Definition Records provided by root (See the <i>REELlibrarian Master Guide</i> for a detailed description of Default Formats).</p> <p>The only keyword that is required is <code>lvsn</code> which gives the volume serial numbers associated with the volumes belonging to the volumeset.</p> <p>Upon successful completion, <code>rlvsubmit</code> displays the volume ID assigned to each constituent volume. The volume ID is a unique name for the volume within the catalog; it should be written on an external label affixed to the volume.</p> <p>After <code>rlvsubmit</code> completes, the catalog contains a volume record for each volume in the volumeset. If REELlibrarian is configured to require volume acceptance and/or volume identification, the volumeset is not accessible until it is handed over to an operator who uses <code>rlaccept(8)</code> and <code>rlid(8)</code> to complete submission.</p>
Options	<p><i>volset</i> Volumeset Name. A meaningful name assigned by the volumeset owner. The general form of <i>volset</i> is <code>[userid/]vname[:Ggno][:Vvno]</code>. If <i>userid</i> is omitted, the effective user ID is assumed. <i>vname</i> is an arbitrary string up to 12 characters long. <i>Ggno</i> specifies a volumeset generation number. <i>Vvno</i> specifies a volumeset version number. Generation and version numbers are used as subscripts for volumesets with the same <i>vname</i>.</p> <p>Note: Only an operator may submit to a <i>userid</i> other than the current effective user ID.</p>

gno can be an integer or a signed integer. If *gno* is an integer, it references the given generation number. If *gno* is a signed integer, it references the highest existing generation number offset by *gno*. For example, if the highest generation number for volumeset test is 10, *:G-1* refers to generation 9 and *:G+1* refers to generation 11. If the generation specification is omitted, the highest existing generation is referenced. If there are no existing generations, generation 0 is selected. The highest allowable generation number is 9999.

Vvno specifies a volumeset version number. Version numbers behave like generation numbers. The highest allowable version number is 99.

If *volset* is omitted, an unnamed volumeset is created. Unnamed volumesets are referenced by a '.' followed by the volume ID of the first volume in the volumeset. For example, if the first volume has volume ID *1ump-992* the volumeset is referenced as *.1ump-992*.

keywords

One or more keyword=value assignments. There are no specified defaults associated with these keywords except where noted. Keywords recognized by this command are:

app= Application prefix. This prefix is used with file reports to limit selections to those with the specified *app=* value. *app* may be up to 12 characters long.

capacity=

Volume capacity in Mbytes. Specifies the volume capacity for cartridge media types.

Note: For 9-track media, use the length keyword.

convert=

Record conversion specification. *convert* controls conversion of records to/from tape. *convert* may be set to one of: *text* for text records, *etext* for EBCDIC text records, *data* for fixed length ASCII or binary data records.

edata for fixed length EBCDIC data records.

dispose=

Volume disposition. `dispose` controls when volumes leave the volumeset and what happens when they do. `dispose` may be set to `erase` and/or `retain`. To select both, specify `dispose="erase&retain"`. If set to `retain`, the volumeset is never truncated, and all original volumes remain as volumeset members. If set to `erase`, whenever a file on the volumeset is overwritten, all volumes following the new file are scratched.

ecnt= Error count. Set error count as specified.

finger=File fingerprint. Set fingerprint field as specified.

flocation=

Free volume location. When a volumeset is truncated or scratched, the newly unattached volumes will return to the location assigned to `flocation`. The value given must be defined as a REELibrarian storage site (the command `rlr sites` will produce a list of currently defined storage sites). Up to twelve characters long.

format=Label Format. `format` may be set to one of `ANSI`, `IBM`, `IBMU`, `TAR`, `CPIO` or `RAW`.

ftrack=File Tracking Flag. `yes` causes the catalog to maintain a record for every file written to the volumeset. `no` disables file cataloging. Maintaining file catalogs makes it possible to generate a volumeset table of contents without having to mount a tape. It also improves the efficiency of accessing files on multi-volume volumesets.

length=Length of tape in feet. Specifies the length of 9-track media. Default: 2400.

Note: For 8mm, 4mm, and QIC media, use the `capacity` keyword.

lvsn= Volume Serial Number List. Required. A comma separated list giving the Volume Serial Number (VSN) for each volume in the volumeset. If the VSN is specified as `agen` REELibrarian will automatically assign the volume a unique

VSN (in general, VSNs do not have to be unique). In place of the comma separator “,” may be used as a separator to indicate volumes that belong to the volumeset but do not yet contain data. For example, `lvsn=000000,000001/agen`, describes a volumeset with three volumes. The first two volumes contain existing files. The third volume is a spare to allow room for future expansion. The VSN for the third volume is automatically generated. The “/” separator may occur at the beginning of the list (Example: `lvsn=/000000,000001`) to indicate a volumeset with no existing files. Up to six characters long for each. The special value `vms` means that this value is selected by the Vault Management System.

`offset`=Record offset. The number of bytes reserved at the beginning of each block for additional information. Offset only effects the ANSI tape format.

`passwd`=Volume Access Password. An optional password which, if specified, must be given before the volumeset can be read or written. Up to fourteen characters long. Example:
`password=undertow`. There is no specified default.

`pool`= Pool membership. Every volume must belong to a pool. By default, volumes belong to the user’s private pool. Volumes may only be assigned to pools owned by the current effective user ID. Up to twelve characters long.

Note: The full specification of pool is `userid/pool`. If `userid` is omitted, the current effective user ID is assumed. Only operators may create pools for IDs other than their own.

`rformat`=

Record Format. `rformat` has the general form:

fmt:blen:rlen. Where *fmt* is the record format:

f - fixed length records;

fb - fixed length, blocked records;

v - variable length records;

vb - variable length, blocked records;

vs - variable length, spanned records;

vbs - variable length, blocked, spanned records.

u - unformatted data.

blen is the block length in bytes. *rlen* is the record length in bytes. Example *rformat=fb:800:80*. Default: *u*.

slocation=

Scheduled Volume Location. If a volume's current location (*clocation*) does not equal its scheduled location. The volume will show up on the Volume maintenance report as wanting to move from *clocation* to *slocation*. The value given must be defined as a REELlibrarian storage site (the command *rlr sites* will produce a list of currently defined storage sites). Up to twelve characters long. Default: *clocation* setting.

type= Volume Media Type. Media type is an arbitrary name assigned during REELlibrarian configuration to describe volume formats supported by devices under REELlibrarian control. For 9-track devices, this is typically the tape density (i.e. 1600, 6250, etc.). For cartridge devices, it is typically the cartridge standard (i.e. QIC11, QIC24, etc). The given value must be a defined media type. The command *rlr mtype* produces a list of currently defined media types. Up to eight characters long.

vaccess=

Volume Access Byte. The character assigned to *vaccess* is written as byte 11 (Volume Accessibility) in the VOL1 label. REELlibrarian attaches no particular significance to the Volume Access byte; control is provided for export of

volumes to sites that do require certain values.

vcomment=

Volumeset comment. A comment about the volumeset. If the comment includes spaces, it must be enclosed in quotes. Up to forty characters long. Example: `vcom="my favorite volumeset"`.

vexpire=

Volumeset Expiration Date. One of:

I - infinite (never expires),

S - scratch (immediately expired),

RN - expires *N* days after creation,

AN - expires if not accessed in *N* days,

L - expires when all files on the volumeset have expired,

Orotsched - follows the rotation schedule *rotsched* (see `rlrcreate(1)`),

Xccyyymmdd or *Xmm/dd/yy* - expires on given date

GN - expire when there are *N* newer generations.

Please note that upon expiration the volumeset is not disbanded until it is scratched. The `rlvscratch` program disbands volumesets. Example: `vexpire=R30`. Default: `s`.

vmode=Volumeset Permission Mask. Three octal digits controlling owner, group and others (respectively) permission to the volumeset (similar to the UNIX file mode). Bit 4 controls read access. Bit 2 controls write access. Bit 1 permits viewing of the volume's catalog entry. Example: `vmode=744`.

The following keywords may only be used by REELlibrarian operators:

clean=Volume cleaning count. The number of times the volume has been mounted since it was last cleaned. This value is normally maintained by REELlibrarian.

clocation=

Current Volume Location. The value given must be defined as

a REELlibrarian storage site (the command `rlr sites` produces a list of currently defined storage sites). Up to twelve characters long.

`ctype`=Current Media Type. For initialized volumes, this field contains the Media Type REELlibrarian has verified is the correct media type for the volume. This field is normally maintained by REELlibrarian.

`group`=Volumaset Group. The group (from `/etc/group`) to which the volumaset belongs. By default, `group` is the current effective group ID. Up to twelve characters long.

`maintenance`=

Scheduled Maintenance. Maintenance May be set to one or more of `move`, `age`, `erase`, `remove` and `clean`. Multiple items are selected by separating them with “&” (i.e. `maint=“erase&clean”`). `move` indicates the volume is awaiting movement. `age` causes the volume to wait *N* days after it is scratched before becoming free (*N* is defined by the pool the tape belongs to). `erase` schedules the volume for erasure. `clean` schedules the volume for cleaning.

Note: Normally the maintenance value is maintained by REELlibrarian.

`remove`=Volume Usage Count. The number of times the volume has been mounted since it entered the catalog. This value is normally maintained by REELlibrarian.

`status`=Volume submission status. Status may be assigned the following integer values.

- 1 - submitted awaiting acceptance.
- 2 - accepted awaiting identification.
- 3 - fully submitted.
- 4 - retrieved awaiting return.

Note: Submission status is normally maintained by REELlibrarian.

`user`= Volumaset Owner. The user ID (from `/etc/passwd`) to

which the volumeset belongs. By default user is the current effective user ID. Up to twelve characters long.

The following keywords are optional and an empty value is acceptable:

`ftemplate=`

File Name template. The filename template is used to dynamically construct names for files written to the volumeset. The name constructed by the template is limited to seventeen (17) characters. The template consists of constant text and substitution patterns. Substitution patterns have the general form:`@sublen@` where the “@” symbol delimits the beginning and end of the substitution specification. `sub` is a character indicating what to substitute. `len` gives the length in characters of the substituted value. In principle `len` can be any number between 1 and 17, however, for some values of `sub` only a particular value of `len` is reasonable. The following values are recognized.

@Y4@ - numeric year,

@C2@ - numeric month (Jan = 1),

@E3@ - month name,

@D2@ - day of month,

@J3@ - Julian day,

@W3@ - day of week (Sun = 1),

@H2@ - hour (24 hour clock),

@M2@ - minute,

@S2@ - second,

@F9@ - value assigned `fid` keyword on the `rlvwrite` command,

@U9@ - base name of file assigned to the `if` keyword on the

rlvwrite command,
 @G4@ - file generation,
 @V2@ - file version.

Numeric values are truncated on the left. Character values are truncated on the right. If a numeric value is shorter than *len* it is padded on the left with zeros. Example on January 25 1987: `ftemplate=@D2@@E3@@Y2@G@G4@V@V2@`, results in a file name of: 25Jan89G0000V01.

Note: Even unlabeled volume formats (IBMU, RAW, TAR and CPIO) must be assigned volume serial numbers.

`lrack=`Rack Number List. A comma separated list giving the rack numbers corresponding to the Volume Serial Number List (keyword `lvsn`). The `lrack` keyword is provided here so operators can submit tapes in a single step. Each rack number can be up to twelve characters long. The special value `vms` means that this value is selected by the Vault Management System.

Note: If volume acceptance is required, the rack number is normally assigned when the operator accepts the volume.

`lvid=` Volume ID List. A comma separated list giving the volume IDs corresponding to the Volume Serial Number List (keyword `lvsn`). Volume IDs must be unique. Sites that maintain unique VSNs may make the volume ID the same as the VSN. By default, REELlibrarian, assigns each volume a unique volume ID by combining a four character word with a three digit number (Example: `root-496`). Each volume ID may be up to twelve characters long. The special value `vms` means that this value is selected by the Vault Management System.

See Also `rlr(1)`, `rlvretrieve(1)`.

Name	<code>rlvtran</code> - reinitialize a transient volumeset
Synopsis	<code>rlvtran write={yes no} volset</code>
Description	<p>A transient volumeset is a volumeset whose constituent volumes change periodically. For example, if a site receives weekly sales updates from a regional sales office on tape, it would be reasonable to create a transient volumeset <code>'sales_up'</code>. Each week, referring to volumeset <code>'sales_up'</code> selects the latest sales information.</p> <p>There is no real difference between a normal volumeset and a volumeset that is considered transient. The only trick is when a mount request is made for a transient volumeset, the operator must be able to recognize that the requested volume will not be found in the tape library but in some other location (probably the incoming mail). This is usually accomplished by giving the rack number a special value. In this example, the rack number could be <code>'sales_up'</code>. If it is a multi-volume volumeset the volumes might have rack numbers <code>'sales_up.1'</code>, <code>'sales_up.2'</code>, etc.</p> <p>The difficulty for REELlibrarian in managing a transient volumeset is detailed information about the constituent volumes changes periodically. For example, the volume fingerprint which is taken the first time the volumeset is mounted will be incorrect when a new volume arrives. The command <code>rlvtran</code> resets the necessary volume information so a new transient volume can be accessed successfully. Specifically, <code>rlvtran</code> wipes out all associated file records, sets the volume fingerprint to unknown, the volume type to unknown, and if <code>write=yes</code> is specified, turns on volume initialization, otherwise it sets the volume serial number to unknown. The <code>rlvtran</code> command should be given before accessing a transient volumeset with new constituent volumes. Default is <code>no</code>.</p>
See Also	<code>rlvcreate(1)</code> , <code>rlvsubmit(1)</code> .

Name rlvtruncate - truncate a volumeset

Synopsis rlvtruncate *fseq=fileno vsname*

Description rlvtruncate truncates the named volumeset at the specified file sequence number.

For example, if a volumeset consists of ten files on eight tapes and the tenth file alone consumes the last three tapes, then this command removes the last three tapes from the volumeset when the file sequence number '10' is specified. The other nine files are left unaltered on the remaining five tapes. The three truncated tapes are expired.

Options *fseq=* The file sequence number of the file targeted. Volumeset files are numbered starting with '1'.
*vsname*The volumeset name.

See Also rlr(1).

rlvunmount(1)

Name	rlvunmount - unmount a volume
Synopsis	rlvunmount [<i>psd</i>]
Description	rlvunmount unmounts the volume associated with device <i>psd</i> . If the corresponding <i>rlvmount</i> command reserved the device the device is automatically freed.
Options	<i>psd</i> Pseudo Device Name. An arbitrary name assigned during device reservation; used to distinguish between devices when multiple devices are reserved.
See Also	rlvmount(1).

Name	<code>rlvwrite</code> - write a file to a volumeset
Synopsis	<code>rlvwrite keywords [[on] <i>psd</i>]</code>
Description	<p><code>rlvwrite</code> writes a file to the volumeset accessed (<code>rlvaccess(1)</code>) on device <i>psd</i>. By default, <code>rlvwrite</code> writes the new file at the current volumeset position. The <code>fseq</code> keyword may be used to write the file at a location other than the current location.</p> <p>Input to <code>rlvwrite</code> can be taken from a file with keyword <code>if</code> or a FIFO with keyword <code>fifo</code>. By default, <code>rlvwrite</code> gets its input from <code>stdin</code>.</p> <p><code>rlvwrite</code> follows the UNIX convention of exiting with a zero return if successful and a non-zero on failure.</p> <p><code>rlvwrite</code> should not be interrupted; if the process is killed, <code>rlvrelease</code> the referenced volumeset.</p>
Options	<p><i>psd</i> Pseudo Device Name. An arbitrary name assigned during device reservation; used to distinguish between devices when multiple devices are reserved.</p> <p>keywords</p> <p>One or more keyword=value assignments. Keywords recognized by this command are:</p> <p><code>app=</code> Application prefix. This prefix is used with file reports to limit selections to those with the specified <code>app=</code> value. <code>app</code> may be up to 12 characters long. There is no specified default.</p> <p><code>if=</code> Input File. If “if” is given, <code>rlvwrite</code> gets its input from the named file. Default: <code>stdin</code>.</p> <p><code>fid=</code> File ID. Specifies a tape file name. The assigned value has the general form: <i>fname</i>[:<i>Ggno</i>][:<i>Vvno</i>]. <i>fname</i> is the file name, an arbitrary string up to 17 characters long. <i>Ggno</i> gives a file generation number. <i>Vvno</i> gives a file version number. Generation and version numbers are used as subscripts for files with the same <i>fname</i>.</p> <p><i>gno</i> can be an integer or a signed integer. If <i>gno</i> is an integer, it references the given generation number. If <i>gno</i> is a signed integer, it references the highest existing generation number</p>

rlvwrite(1)

offset by *gno*. For example, if the highest generation number for file forecast is 10, *:G-1* refers to generation 9 and *:G+1* refers to generation 11. If the generation specification is omitted, the highest existing generation is referenced. If there are no existing generations, generation 0 is selected. The highest allowable generation number is 9999.

vno specifies a file version number. Version numbers behave like generation numbers. The highest allowable version number is 99. The default is the next available file.

fseq= File Sequence Number. Selects a file by its relative position on the volumeset. *fseq=1* selects the first file, *fseq=2* selects the second, etc. *fseq=EOT* appends to the volumeset without specifying a specific file sequence number. The default is the next available file.

fsect=File Section Number. Files that span multiple volumes are broken into sections. Normally, access starts at the first section (i.e. the beginning of the file). The *fsect* keyword may be used with either the *fid* or *fseq* keywords to initiate access at a specific section. Default: 1.

Note: spanned record formats (*vs* and *vbs*) should not be used with a file section number other than one.

irtype=Input Record Type. Can be set to either *fixed* (the default setting) or *variable*. When set to *variable*, it looks for a six byte ASCII integer at the beginning of each input line. When set to *fixed*, *rlvwrite* does not look for the prefixed integer.

span= Span Flag. By default *rlvread* and *rlvwrite* cross volume boundaries transparently. That is, an I/O operation that encounters End-of-Volume automatically requests the next volume. Setting *span=no* disables transparent volume crossing, the first End-of-Volume encountered is treated like End-of-File. Default: *yes*.

rformat=

Record Format. *rformat* has the general form:

fmt:blen:rlen. Where *fmt* is the record format: *f* - fixed length records; *fb* - fixed length, blocked records; *v* - variable

length records; *vb* - variable length, blocked records; *vs* - variable length, spanned records; *vbs* - variable length, blocked, spanned records. *u* - unformatted data. *blen* is the block length in bytes. *r1en* is the record length in bytes.

Note: A special case exists for ANSI tapes when *r1en* is set equal to 0 and *fmt* is either *vs* or *vbs*. In this situation, REELlibrarian reads each block and places each record into record buffer of 32 kilobytes unless the user's environment variable *RL_RMAX* is set to a different buffer size (in bytes). If the user knows that the file's records can exceed 32 kilobytes, then the *RL_RMAX* environment variable should be used or data will be truncated. Example: *rformat=fb:800:80*. There is no specified default.

offset=Record offset. The number of bytes reserved at the beginning of each block for additional information. Offset only effects the ANSI tape format. There is no specified default.

conv= Record conversion specification. *conv* controls conversion of records to/from tape. *conv* may be set to one of *text* for text records. *etext* for EBCDIC text records. *data* for fixed length ASCII or binary data records. *edata* for fixed length EBCDIC data records. There is no specified default.

fexpire=

File Expiration Date. One of:

I - infinite (never expires)

S - scratch (always expires)

RN - expires *N* days after creation (maximum: 999)

AN - expires if not accessed in *N* days

L - expire when all files on the volumeset have expired

XCCYYmmdd or *Xmm/dd/YY* - expires on given date

GN - expire when there are *N* newer generations. There is no specified default.

Note: file expiration is only considered if volumeset expiration is set to *L*. Example: *cdate=R30*.

rlvwrite(1)

`force=force=yes` instructs the command to continue even if the `volumeset` is not expired. Only the owner of a `volumeset` can use the `force` keyword. Default: `no`.

`faccess=`

File Access Byte. The character assigned to `faccess` is written as byte 54 (Accessibility) in the HDR1 label. REELlibrarian attaches no particular significance to the File Access byte; control is provided for export of volumes to sites that do require certain values. There is no specified default.

`fmode=`File Permission Mask. Three octal digits controlling owner, group and others (respectively) permission to the file. Bit 4 controls read access. Bit 2 controls write access. Bit 1 permits viewing of the file's catalog entry. Example: `fmode=744`. There is no specified default.

The following keywords are optional and an empty value is acceptable:

`fcomment=`

File comment. A comment about the File. If the comment includes spaces, it must be enclosed in quotes. Up to forty characters long. Example: `fcomment="my favorite File"`.

`fifo=` Input FIFO. If `fifo` is given, `rlvwrite` takes its input from the named FIFO. If the FIFO does not exist, it is created. When taking input from a FIFO `rlvwrite` puts itself in the background after positioning to the proper file.

`fpasswd=`

File Access Password. An optional password which, if specified, must be given before the File can be read or written. Up to fourteen characters long. Example: `fpasswd=undertow`.

`warn=` Warning messages. If set to `yes`, warning messages will be provided. Default is `no`.

See Also `rlvaccess(1)`, `rlvread(1)`.

C Library Functions (UNIX Section 3)

This appendix includes the UNIX-style manual pages for each of the C Library functions.

RL_KEY(3)

Name	RL_KEY - creates unique instances of access for same UIDs to allow for multiple users of same session
Synopsis	<pre>int rl_setkey(key) char *key ;</pre> <p>Note: This call is just like setting the RL_KEY environment variable.</p>
Description	<p>When used, RL_KEY will allow multiple users with the same User ID (UID) to be treated as different users through the use of a unique key (RL_KEY). This helps prevent deadlock situations.</p> <p>Note: The key can be no longer than 12 characters (13 with NULL).</p>
Examples	<pre>RL_KEY=\$\$ export RL_KEY</pre> <p>This command sets the process ID \$\$ to RL_KEY and exports it into the environment.</p>

Name	rec_get - read a record from a volumeset file
Synopsis	<pre>#include "reel.h" int rec_get(vsd, buf) int vsd; char *buf;</pre>
Description	<p>vsd is a volumeset descriptor obtained from vs_vaccess(3) and opened for reading with vs_open(3). rec_get reads the next record from the volumeset and places the result in the buffer pointed to by buf. The record is unblocked and converted as specified by the RSPEC and CSPEC structures given to vs_open.</p> <p>buf must be large enough to hold the largest record in the file after conversion processing.</p>
Return Values	<p>Upon successful completion, the number of bytes in the record is returned. At End-of-File, the value TAPE_EOF. If there is an error, -1 is returned and terrno indicates the error condition.</p> <p>Note: Zero-length records are possible for some record formats.</p>
See Also	vs_open(3), vs_vaccess(3).

rec_put(3)

Name	rec_put - write a record to a volumeset file
Synopsis	<pre>#include "reel.h" int rec_put(vsd, buf, len) int vsd, len; char *buf;</pre>
Description	<p>vsd is a volumeset descriptor obtained from <code>vs_vaccess(3)</code> and opened for writing with <code>vs_open(3)</code>. <code>rec_put</code> converts and blocks the record in <code>buf</code> according to the conversion specification and record specification given to <code>vs_open</code>.</p> <p>Record conversion is performed directly in <code>buf</code> so the data in <code>buf</code> may be modified. <code>buf</code> must be large enough to hold the largest record in the file after conversion processing.</p>
Return Values	Upon successful completion, the number of bytes in the converted record is returned. If there is an error, -1 is returned and <code>terrno</code> indicates the error condition.
See Also	<code>vs_open(3)</code> , <code>vs_vaccess(3)</code> .

Name reel_error - provide a descriptive error message

Synopsis

```
#include "reel.h"
char * reel_error()
```

Description When the REELlibrarian procedures encounter an error, the global integer `terrno` is set to a value indicating the error condition. `reel_error` returns a pointer to a null terminated character buffer containing a brief description of the error condition.

rl_dev(3)

Name	rl_dev - modify device status
Synopsis	<pre>#include "reel.h" int rl_dev(adname, status) char *adname; int status;</pre>
Description	rl_dev modifies the status of device adname to reflect status. Status of 0 takes the device down. Status of 1 brings the device up. Status of 2 resets the device.
Return Values	Upon successful completion, 0 is returned. If there is an error, -1 is returned and terrno indicates the error condition.

Name	rl_done - indicate completion of an operator request
Synopsis	<pre>#include "reel.h" int rl_done(mid, vid, flags, rack, vsn) int mid; char *vid, *rack, *vsn; int flag</pre>
Description	<p>rl_done informs RL that the operator request associated with mount ID mid is complete. If mid is -1, the outstanding operator request with the lowest mid is assumed. vid is the volume ID for the request. vid may or may not be required depending on the corresponding request.</p> <p>The following flags are accepted:</p> <ul style="list-style-type: none">TD_DONE which has rl_done correspond as normalTD_FORCE which overrides a mount requestTD_NEW which corresponds to a VADD request
Return Values	Upon successful completion, 0 is returned. If there is an error, -1 is returned and terrno indicates the error condition.

rl_errlist(3)

Name	<code>rl_errlist</code> - return a list of descriptive error messages
Synopsis	<pre>#include "reel.h" MSG *rl_errlist()</pre>
Description	A number of REELlibrarian function calls do extensive error checking before allowing creation or modification of items in the REELlibrarian database. In order to provide error messages that are meaningful, the error checking mechanism needs a way to return error messages that describe exactly what is wrong with a particular request. When a REELlibrarian function call returns with <code>terrno</code> , the <code>rl_errlist</code> may be called to return a linked list of error messages generated by the corresponding request.
Return Values	Upon successful completion, a pointer to the head of the error message list is returned. If <code>rl_errlist</code> is called when <code>terrno</code> is not set to <code>RL_ELIST</code> a null pointer is returned.

Name	rl_fedit - edit the catalog entry of a file
Synopsis	<pre>#include <time.h> #include "reel.h" int rl_fedit(vname, fname, fseq, fpt) char *vname, *fname; int fseq; FREC *fpt;</pre>
Description	<p>rl_fedit modifies the catalog entry of the specified file. The file must be specified uniquely. Files are specified in one of three ways:</p> <ol style="list-style-type: none"> 1. By <code>fname</code>. <code>fname</code> must be unique in the library. 2. By <code>vname</code> and <code>fname</code>. This allows specification of <code>fname</code> that is unique on volumeset <code>vname</code>; <code>fname</code> may occur on other volumesets within the library. 3. By <code>vname</code> and <code>fseq</code>. This allows specification of a file that does not have an <code>fname</code>, or whose <code>fname</code> is not unique on volumeset <code>vname</code>. <p><code>fpt</code> is an edit template that is initialized with a call to <code>free_null(fpt)</code> and then modified to reflect the fields that are being changed.</p> <p>Default values of unused parameters are listed below.</p> <pre>vname = (char *) 0; fname = (char *) 0; fseq = -1;</pre> <p>Files that are on volumesets that are in use cannot be edited.</p>
Return Values	Upon successful completion, 0 is returned. If there is an error, -1 is returned and <code>terrno</code> indicates the error condition.
Security	A file may only be edited by its owner.

rl_flist(3)

Name	<code>rl_flist</code> - given a volumeset name, return a linked list of the files on the volumeset
Synopsis	<pre>#include "reel.h" int rl_flist(vname, fhead) char *vname; FREC **fhead;</pre>
Description	<code>rl_flist</code> returns a linked list of the files on the volumeset to file <code>fhead</code> .
Return Values	Upon successful completion, 0 is returned. If there is an error, -1 is returned and <code>terrno</code> indicates the error condition.

Name rl_frec - given a unique file name return the corresponding file record

Synopsis

```
#include "reel.h"
int rl_frec(fname, fpt)
char *fname;
FREC *fpt;
```

Description rl_frec return the file record that corresponds to the supplied file name.

Return Values Upon successful completion, 0 is returned. If there is an error, -1 is returned and `terrno` indicates the error condition.

rl_free(3)

Name	rl_free - free a device
Synopsis	<pre>#include "reel.h" int rl_free(psd) char *psd;</pre>
Description	rl_free frees the previously reserved device (rl_reserve(3)) with the given pseudo device name. If psd is a null string (""), all devices associated with the caller's user name and request key are released.
Return Values	Upon successful completion, 0 is returned. If there is an error, -1 is returned and terrno indicates the error condition.
See Also	rl_reserve(3).

Name rl_inq - retrieve device names associated with a mounted volume

Synopsis

```
#include "reel.h"
ADN *rl_inq(psd)
char *psd;
```

Description rl_inq returns the ADN structure of the device associated with psd.

Return Values If a device has been assigned, then a pointer to the corresponding ADN structure is returned.

If no device reservation has been made, a NULL pointer is returned and terrno is set to RL_NODEV.

If the device reservation is still queued and waiting for a device assignment, a NULL pointer is returned and terrno is set to RL_QUEUED.

rl_moved, rl_cleaned, rl_removed(3)

Name	rl_moved, rl_cleaned, rl_removed - volume management operations
Synopsis	<pre>#include "reel.h" int rl_moved(loc, vid, nrack) char *loc, *vid, *nrack; int rl_cleaned(vid) char *vid; int rl_removed(vid) char *vid;</pre>
Description	<p>rl_moved confirms movement of volume <code>vid</code> to location <code>loc</code>. If <code>nrack</code> is not a null pointer then it is taken as the rack number for the volume at the new location.</p> <p>rl_cleaned resets the cleaning count associated with volume <code>vid</code>.</p> <p>rl_removed removes volume <code>vid</code> from the catalog.</p>
Return Values	Upon successful completion, 0 is returned. If there is an error, -1 is returned and <code>terrno</code> indicates the error condition.

Name	rl_op - return a linked list of outstanding operator actions
Synopsis	<pre>#include "reel.h" int rl_op(head) MSG **head;</pre>
Description	<p>rl_op makes head a pointer to a linked list of outstanding operator action requests. If there are no outstanding actions, head is returned as a null pointer.</p> <p>The first two items in the linked list contain column headings.</p>
Return Values	Upon successful completion, 0 is returned. If there is an error, -1 is returned and <code>terrno</code> indicates the error condition.

rl_poll(3)

Name	rl_poll - query for completion of a reservation or volume mount
Synopsis	<pre>#include "reel.h" int rl_poll(type, psd) int type; char *psd;</pre>
Description	<p>rl_poll allows processes to test for completion of a reservation request (see rl_reserve(3)) or a volume mount (see rl_vaccess(3)).</p> <p>type must be either WAIT_RSV to check for a device reservation or WAIT_MNT to check for a mount request. psd is the Pseudo Device Name of the request to wait for (when doing a WAIT_RSV, psd is ignored).</p> <p>This function returns the same value as rl_wait, except no waiting is done. To poll until a request completes: Submit a request and call rl_poll occasionally until it returns (-1).</p> <p>Note: If your application has no other work to perform between rl_poll calls, rl_wait would be a better choice.</p>
Return Values	If 0 is returned, rl_wait suspends. If there is an error, -1 is returned and terrno indicates the error condition.
See Also	rl_wait(3)

Name rl_reserve - reserve devices

Synopsis

```
#include "reel.h"
int rl_reserve(q_flg, prio, rlist)
```

```
int q_flg, prio;
RLIST *rlist;
```

Description rl_reserve places a reservation request with the REELlibrarian Mount Request System. If devices are immediately available to satisfy the request, the devices are reserved and rl_reserve returns zero. If devices are not immediately available and q_flg is zero, -1 is returned and terrno is set to RL_QNOQ. If q_flg is non-zero, the request is queued, -1 is returned, and terrno is set to RL_QUEUED. A queued request's position in the queue is determined by prio. prio is an integer value from 0 to 9 indicating a request's priority; the lowest numbers have the highest priority.

rlist is a linked list of device reservation specifications. Each node defines the attributes of a single requested device. A reservation request does not succeed until devices are available to satisfy all nodes in a request.

In each node one of vname, type, or adn should be specified to indicate the type of device required. vname specifies a volumeset name which implies a type compatible with the volumeset. type specifies the Media type explicitly. adn specifies a specific device.

If mach is specified, the device reservation is made on the network node with the given machine name. If idle_ok is non-zero, after the reservation is complete, the Mount Request System will not reclaim the device if it accumulates excessive idle time. psd gives the Pseudo Device Name to associated with the device for future reference.

DEVICE RESERVATION LIST

The device reservation list structure RLIST is shown below.

```
typedef struct RLIST
{
char    vname[L_VKEY1];
char    type[L_MTYP1];
char    adn[L_ADN1];
```

rl_reserve(3)

```
char    mach[L_MACH1];
int     idle_ok;
char    psd[L_PSD1];
struct  RLIST *next;
} RLIST;
```

The RLIST structure is used to request a device reservation. A linked list of RLIST structures may be used to request the simultaneous reservation of multiple devices.

When filling out the `rlist` structure, the `idle_ok` and `psd` fields must always be provided. In addition, either `vname`, `type`, `adn` should be provided. The `mach` field should be provided with either `vname` or `type`.

`vname`

Volumeset name. This field requests a device that is compatible with the given volumeset.

`type`

Media Type. This field requests a device that supports the given media type.

`adn`

Device Name. This field requests a device by name. `adn` is assigned the configured device name for a specific device.

`mach`

Machine Name. This field allows requests made by `vname` or `type` to further request a device physically connected to a specific host. `mach` may be either a host name or the special value `any` to indicate no preference. If `mach` is not specified, its value is assumed to be the name of the host from which the request was made if there are compatible devices locally attached, otherwise it defaults to `any`.

`idle_ok`

Idle flag. If set to a non-zero value, the device will not be taken away if it is not used for long periods of time. By default, devices that are not used for a configurable length of time are reclaimed for use by other users.

psd

Pseudo Device Name. This is a user-defined name for the device that will be used in future REELlibrarian calls to distinguish this device from others reserved by the user.

Return Values Upon successful completion, 0 is returned. If there is an error, -1 is returned and `terrno` indicates the error condition.

rl_setkey(3)

Name	rl_setkey - set the resource allocation key
Synopsis	<pre>void rl_setkey(key) char *key;</pre>
Description	<p>rl_setkey establishes <code>key</code> as the new resource key for the process. All future REELlibrarian calls will be associated with the new resource key.</p> <p>The resource key allows two jobs running under the same user ID appear to REELlibrarian as separate jobs.</p> <p>If this function is not used, <code>key</code> is obtained from the environment variable <code>RL_KEY</code>. If <code>RL_KEY</code> is not set, the value <code>default</code> is used.</p>
Return Values	None

Name	rl_skip - skip a mount request
Synopsis	<pre>#include "reel.h" int rl_skip(mid, message) int mid; char *message;</pre>
Description	rl_skip informs RL to skip the mount request associated with mount ID mid. If mid is -1, the outstanding mount request with the lowest mid is assumed. message is an explanatory message describing why the mount request is being skipped. message is forwarded to the user who's mount request is skipped.
Return Values	Upon successful completion, 0 is returned. If there is an error, -1 is returned and errno indicates the error condition.

Name rl_structures - shared REELlibrarian data structures VREC, FREC, CSPEC, and RSPEC

VOLUME RECORD

The volume record structure VREC is shown below.

```
typedef struct VREC
{
    struct VREC *next
    char        rack[13];
    char        vid[13];
    char        vsn[7];
    long        len;
    int         scratch;
    int         init;
    int         maint;
    int         status;
    char        cloc[13];
    int         ucnt;
    int         ccnt;
    int         ecnt;
    char        fingerp[28];
    char        ctype[9];
    char        vname[23];
    char        vsid[13];
    int         vno;
    int         valloc;
    char        passwd[15];
    time_t      ctime;
    time_t      mtime;
    time_t      atime;
    int         vmode;
    char        type[9];
    int         format;
    int         dispose;
    char        rtype;
    int         rdata;
    char        rotation[26];
    char        unname[13];
    char        gname[13];
    char        vacc;
    char        sloc[13];
    char        floc[13];
}
```

```

char    pname[26];
int     ftrack;
char    vcom[41];
char    ftemp[51];
struct  RSPEC rspec;
int     offset;
struct  CSPEC cspec;
char    app[13];
} VREC;

```

Each volume under REELlibrarian control is tracked by a VREC structure.

All character strings are null-terminated.

rack

Rack Number. The address of the volume's storage slot within the current vault. This field is often the same as the volume serial number.

vid

Volume ID. A unique identifier for the volume. This field is often the same as the volume serial number.

vsn

Volume Serial Number. The volume serial number is a volume identification defined by both the IBM and ANSI standards. It is recorded in the VOL1 label of the tape and is typically used to electronically verify the identify of the volume. This field should be unique to ensure correct electronic identification.

len

Volume Length. For record-oriented devices (9 track, 3480, 3490 etc.), this field gives the tape length in feet. For block-type devices, this gives the tape length in megabytes.

scratch

Scratch Flag. If non-zero, the volume is a scratch volume.

init

Initialization Flag. If non-zero, the volume is scheduled for initialization.

maint

Maintenance Bit Mask. One or more of the following:

- MN_MOVE - Tape awaiting movement
- MN_AGE - Tape aging
- MN_ERASE - Tape pending erasure
- MN_REMOVE - Tape pending removal
- MN_CLEAN - Tape pending cleaning
- MN_CERTIFY - Tape pending certification
- MN_LOST - Tape lost

status

Submission Status. One of:

- RS_SUB - Submitted and awaiting acceptance
- RS_ACC - Accepted and awaiting identification
- RS_CMP - Submission complete
- RS_RET - Retrieved and awaiting return

cloc

Current Location. The name of the vault where the tape currently resides.

ucnt

Usage Count. The number of times the tape has been mounted since it entered the library.

ccnt

Cleaning Count. The number of times the tape has been mounted since its last cleaning.

ecnt

Error Count. Number of unrecoverable errors that have occurred since the volume entered the library.

fingerp

Volume Fingerprint. A identification string consisting of the following 27 characters:

First Character - Label type. One of: A - ANSI, I - IBM, U - unlabeled, or X - unknown. For a tape to be mountable, this field must correspond to the format of the volume.

Characters 2 - 7 - Volume Serial Number. For labeled tapes, this field gives the VSN found in the VOL1 label. For unlabeled volumes, this field is recorded as blanks.

Characters 8 - 27 - Fingerprint. A CRC calculated from the first 1000 bytes of the first 5 blocks on the volume. The CRC appears as a random sequence of characters.

ctype

Current media type. For tapes that have been previously mounted, this field gives the media type for the volume.

vname

Volumeset Name. For volumes that belong to a volumeset, this field gives the user-defined name of the volumeset. For unnamed volumesets, this field is empty.

vsid

Volumeset ID. For volumes that belong to a volumeset, this field contains the Volume ID of the first volume in the volumeset.

vno

Volume Number. For volumes that belong to a volumeset, this field gives the ordinal number of the volume within the volumeset.

valloc

Volume in Use Flag. For volumes that belong to a volumeset, if this field has a non-zero value, the volume contains useful data. A zero value indicates the volume does not currently contain useful data.

passwd

Volume Password. For volumes that belong to a volumeset, this field contains the encrypted password. On submission, this field contains the unencrypted password. If recorded as an empty string, the volume is not password protected.

ctime

Creation Time. For volumes that belong to a volumeset, this field gives the date the volumeset was created.

atime

Modification Time. For volumes that belong to a volumeset, this field gives the date the volumeset was last written.

mtime

Access Time. For volumes that belong to a volumeset, this field gives the date the volumeset was last read or written.

vmode

Mode. For volumes that belong to a volumeset, this field gives the permission mask associated with the volumeset.

type

Media Type. For volumes that belong to a volumeset, this field gives the volumeset media type. This field is normally the same as `ctype` but may not be if the volume is pending initialization.

format

Data Format. For volumes that belong to a volumeset, this field identifies the volume format as one of the following:
DF_ANSI - ANSI labeled
DF_IBM - IBM labeled
DF_UNL - Unlabeled
DF_TAR - tar format
DF_CPIO - cpio format
DF_RAW - raw tape

dispose

Volume Disposition Bit Mask. For volumes that belong to a volumeset, one or more of:
VS_ERASE - Volumes that leave the volumeset should be erased before they are made available as scratch volumes.
VS_RETAIN - Retain volumes when the volumeset is truncated. Normally, when a volumeset is truncated, the truncated volumes are returned to scratch status. If

VS_RETAIN is set, truncated volumes remain in the volumes with a `valloc` value of zero.

rtype

Expiration type. For volumes that belong to a volumeset, one of:

RET_EXPDATE - Expiration by date.
 RET_GDCYCLE - Expiration by generation.
 RET_LACCESS - Expiration by last access.
 RET_NEVER - Never expire.
 RET_ONLABEL - Expiration when all files have expired.
 RET_SCRATCH - Always expired.
 RET_RETN - Expiration in N days.
 RET_ROTATE - Expiration based on rotation.

rdata

Expiration data. For volumes that belong to a volumeset, the definition of this field depends on the value assigned to `rtype`.

RET_EXPDATE - Date. Days since 1/1/70.
 RET_GDCYCLE - Generations. Number of generations to retain.
 RET_LACCESS - Days. Specifies to expire if not accessed in given number of days.
 RET_NEVER - Not used.
 RET_ONLABEL - Not used.
 RET_SCRATCH - Not used.
 RET_RETN - Days. Number of days to retain volume.
 RET_ROTATE - Not used.

rotation

Rotation Name. For volumes that belong to a volumeset, and have an `rtype` value of `RET_ROTATE`, this field gives the name of the rotation associated with the volumeset.

uname

Owner Name. For volumes that belong to a volumeset, the login name of the user that owns the volumeset.

gname

Group Name. For volumes that belong to a volumeset, the group name of the user that owns the volumeset.

vacc

Volume Access Byte. When an IBM volume is initialized, the value of this field is used as the volume access byte as defined by the IBM standard.

sloc

Scheduled volume location. This field gives the name of the vault where the volume is scheduled to reside. If `cloc` is different the `sloc`, the volume will be shown as awaiting to move from `cloc` to `sloc` on the volume maintenance report.

floc

Free volume location. This field gives the name of the vault the volume should return to when it is scratched. As part of the volume scratching procedure, `floc` is copied over `sloc`.

pname

Pool Name. Name of the pool that contains the volume.

ftrack

File Tracking Flag. For volumes that belong to a volumeset, this field determines if file records are maintained for files written to the volumeset. A value of 0 disables file records and a value of 1 enables file records.

vcom

Volume Comment. For volumes that belong to a volumeset, arbitrary null-terminated text provided by the user.

ftemp

File Name Template. For volumes that belong to a volumeset, A string used to construct the File ID field for files written to the volumeset. See the description of the `ftemplate` keyword of the `rlvsubmit(1)` command for a description of the format.

rspec

Record Format. For volumes that belong to a volumeset, default record format for files written to the volumeset. See the description of the RSPEC structure for details.

offset

File Offset. For volumes that belong to a volumeset, default file offset (as defined in the ANSI standard) for files written to the volumeset.

cspec

Conversion Specification. For volumes that belong to a volumeset, default record conversion specification for files written to the volumeset.

app

Application. For volumes that belong to a volumeset, the name of the application that controls the volumeset. The application name can be used on certain reports to allow a listing of all volumes and/or volumesets that belong to the same application.

FILE RECORD

The file record structure `FREC` is shown below.

```

struct FREC
{
    struct FREC *next;

    int         fvalid;
    int         fstatus;
    char        vsid[13];
    char        vname[23];

    /* HDR1 */
    char        fid[18];
    int         gen;
    int         ver;
    int         fseq;
    int         fsect;
    char        rtype;

```

rl_structures(3)

```
int          rdata;
char         facc;

/* HDR2 */
struct       RSPEC rspec;
int          offset;

/* HDR3/HDR4 */
struct       CSPEC cspec;
char         app[13];
char         fcom[41];
char         passwd[15];
char         uname[13];
char         gname[13];
int          fmode;

int          nblocks;
int          vol;
int          fno;
long         begin;
long         tend;
time_t       ctime;
time_t       atime;
} ;
```

The FREC structure tracks information about files stored on volumesets. Files records are only created in the database for files which are stored on volumesets having file tracking enabled (see the `ftrack` flag within the VREC structure). One file record exists for each file section. Thus, if a file spans 10 volumes, the file will have 10 FREC structures in the database.

next

Link Pointer. Pointer to allow `frec` structures to be returned as a linked list.

fvalid

File Validity Bit Mask. This field indicates what label information is available for the file. It consists of one or more of the following:

FV_HDR1 - HDR1 information is valid.

FV_HDR2 - HDR2 information is valid.

FV_HDR3 - HDR3 information is valid.
 FVL_HDR1 - HDR1 information is valid on label.
 FVL_HDR2 - HDR2 information is valid on label.
 FVL_HDR3 - HDR3 information is valid on label.

fstatus

File Status Bit Mask. This field tracks the following status information about the file:

FM_EOT - The file is the last on the volumeset.
 FM_EOL - The file is the last known file on the volumeset (there may be others following that have not been encountered yet).
 FM_OPEN - The file has been successfully opened. This implies that the header labels have been successfully read or written.
 FM_CLOSE - The file has been successfully closed. This implies that the trailer labels have been successfully read or written.
 FM_EOF - This is the last file section of a file.

vsid

Volumeset ID. The Volume ID of the first volume in the volumeset that contains this file section.

vname

Volumeset Name. The name of the file that contains this file sections.

fid

File identifier. The file identifier as read or written in the HDR1 label.

gen

File Generation. The file generation number as read or written in the HDR1 label.

ver

File Version. The file version number as read or written in the HDR1 label.

fseq

File Sequence Number. The ordinal number of the logical file on the volumeset. The first file on the volumeset is number 1.

fsect

File Section Number. The ordinal number of this file section in the file. The first file section of a file is number 1.

rtype

Expiration type. This field is only used if the expiration type of the volumeset contains the file is RET_ONLABEL. One of:
RET_EXPDATE - Expiration by date.
RET_GDCYCLE - Expiration by generation.
RET_LACCESS - Expiration by last access.
RET_NEVER - Never expire.
RET_SCRATCH - Always expired.
RET_RETN - Expiration in *N* days.

rdata

Expiration data. The definition of this field depends on one of the following values assigned to *rtype*.
RET_EXPDATE - Date. Days since 1/1/70.
RET_GDCYCLE - Generations. Number of generations to retain.
RET_LACCESS - Days. Expire if not accessed in given number of days.
RET_NEVER - Not used.
RET_SCRATCH - Not used.
RET_RETN - Days. Number of days to retain volume.

facc

File Access Byte. The file access byte as read or written in the HDR1 label.

rspec

Record Format. The record format of records in the file. See the description of the RSPEC structure for details.

offset

Offset. The ANSI Offset defined for the file section.

cspec

Conversion Specification. The record conversion used to create the file. See the description of the CSPEC structure for details.

app

Application. The name of the application that created the file. The application name can be used on certain reports to allow a listing of all the files that belong to the application.

fcom

File Comment. Arbitrary null-terminated text provided by the user.

passwd

File Password. For files that are password protected, this field contains the encrypted password.

uname

Owner Name. UNIX name of the file owner.

gname

Group Name. UNIX group of the file owner.

fmode

Mode. Permission mask associated with the file.

nblocks

Block Count. For record-oriented devices, this field contains the number of blocks in the file. For block-type devices, this field contains the number of 512-byte blocks in the file.

vol

Starting Volume. The ordinal number of the volume in the volumeset that contains this file.

fno

File Number. The ordinal number of this file on the volume.

begin

Beginning Offset. The offset of the beginning of this file section from the beginning of the volume.

end

Ending offset. The offset of the end of this file section from the beginning of the volume.

ctime

Creation Date. The creation date of the file in UNIX internal format.

atime

Access Date. The date of the last time the file was read in UNIX internal format.

RECORD FORMAT

The record format structure `RSPEC` is shown below.

```
typedef struct
{
    char        rfmt;
    int         blen;
    int         rlen;
} RSPEC;
```

The `RSPEC` structure defines a record format. `RSPEC` is an attribute of the `VREC`, `FREC`, and `FSPEC` structures. For `VREC`, `RSPEC` gives the default record format associated with a volumeset. In `FREC`, `RSPEC` gives the record format used to create the given file. In `FSPEC`, `RSPEC` defines the record format to be associated with a new file or overrides the `RSPEC` currently associated with an existing file.

`rfmt`

Record format specification. One of:

`RF_U` - Unformatted records

`RF_F` - Fixed-length records

`RF_FB` - Fixed-length blocked records

`RF_V` - Variable-length records

`RF_VB` - Variable-length, blocked records

RF_VS - Variable-length, spanned records
 RF_VBS - Variable-length, blocked, spanned records

blen

Block length, in bytes.

rln

Record length, in bytes.

CONVERSION SPECIFICATION

The conversion specification structure CSPEC is shown below.

```
typedef struct
{
    char    tflg;
    char    tchar;
    char    pflg;
    char    pchar;
    char    tran;
} CSPEC;
```

The CSPEC structure defines the conversion specification associated with a file. CSPEC is an attribute of the VREC, FREC and FSPEC structures. For VREC, CSPEC gives the default record conversion associated with a volumeset. In FREC, CSPEC gives the record conversion used to create the given file. In FSPEC, CSPEC defines the record conversion associated with a new file.

pflg

Pad processing enable flag. If set to '1' (ASCII character 1), pad processing is enabled.

pchar

Pad character. On calls to `rec_put()` using a fixed-length record, pad characters are added to the given buffer until its length equals the fixed record length. On calls to `rec_get()` trailing pad characters are removed from each record.

tflg

Termination processing enable flag. If set to '1' (ASCII character 1), termination processing is enabled.

tchar

Termination character. On calls to `rec_put()`, records ending with `tchar` are truncated by one byte. On calls to `rec_get()`, `tchar` is added to the end of each returned record.

tran

Translation control. If set to '1' (ASCII character 1), `rec_put()` performs an ASCII-to-EBCDIC translation and `rec_get()` performs an EBCDIC-to-ASCII translation.

The following CPEC values correspond to the `conv` keyword values for `data`, `edata`, `text` and `etext`.

`data`:

```
pflg = tflg = tran = '0';
```

`edata`:

```
pflg = tflg = '0';  
tran = '1';
```

`text`:

```
pflg = tflg = '1';  
pchar = ' ';  
tchar = '\n';  
tran = '0';
```

`etext`:

```
pflg = tflg = '1';  
pchar = ' ';  
tchar = '\n';  
tran = '1';
```

Name	rl_unq - cancel a pending device reservation request
Synopsis	<pre>int rl_unq(uname, key) char *uname; char *key;</pre>
Description	<p>rl_unq cancels the pending resource allocation request for the given user and key.</p> <p>REELibrarian administrators and operators can cancel requests for all users. All other users may only cancel their own requests.</p>
Return Values	Upon successful completion, 0 is returned. If there is an error, -1 is returned and <code>errno</code> indicates the error condition.

rl_valloc(3)

Name	rl_valloc - create a new volumeset from a pool
Synopsis	<pre>#include "reel.h" int rl_valloc(vname, vpt, queue) int queue; char *vname; VREC *vpt;</pre>
Description	rl_valloc uses information provided in vpt to create a new volumeset named vname. vpt should be initialized with a call to vrec_null and then any non-default fields modified. If vname is a null string, an unnamed volumeset is created. If queue is zero, then rl_valloc will return immediately. If non-zero, then rl_valloc will wait until a volume becomes available.
Return Values	Upon successful completion, 0 is returned and the volume record of the selected volume is copied into vpt. If there is an error, -1 is returned and terrno indicates the error condition.

Name	rl_vfrec - given a volumeset name and a file name return the corresponding file record
Synopsis	<pre>#include "reel.h" int rl_vfrec(vname, fname, fpt) char *vname, *fname; FREC *fpt;</pre>
Description	rl_vfrec returns the file record that corresponds to the supplied volumeset name and file name.
Return Values	Upon successful completion, 0 is returned. If there is an error, -1 is returned and <code>errno</code> indicates the error condition.

rl_vid(3)

Name	rl_vid - locates vid in the catalog and returns the VREC in vpt
Synopsis	<pre>#include "reel.h" int rl_vid(vid, vpt) char *vid; VREC *vpt;</pre>
Description	<p>rl_vid locates vid in the catalog and returns the VREC in vpt</p> <p>vidvolume ID</p> <p>VRECvolume record</p> <p>vptvolume pointer</p> <p>Note: The volume pointer must be initialized so that it points to a volume record prior to the function call.</p>
Return Values	Upon successful completion, 0 is returned. If there is an error, -1 is returned and <code>terrno</code> indicates the error condition.

Name `rl_vlist` - given a volumeset name, return a linked list of the volumes in the volumeset

Synopsis

```
#include "reel.h"
int rl_vlist(vname, vhead)
char *vname;
VREC **vhead;
```

Description `rl_vlist` return a linked list of the volumes in the specified volumeset.

Return Values Upon successful completion, 0 is returned. If there is an error, -1 is returned and `terrno` indicates the error condition.

rl_vmount(3)

Name	rl_vmount - request raw device access to a volumeset
Synopsis	<pre>#include "reel.h" int rl_vmount(psd, vname, passwd, mode, flags, vpt) char *psd, *vname, *passwd; int mode, flags; VREC *vpt;</pre>
Description	<p>rl_vmount asks the operator to mount volumeset vname on the device associated with Pseudo Device psd. If the volumeset is password protected passwd is the unencrypted password associated with the volumeset.</p> <p>If mode is T_READ the volumeset is mount read only. If mode is T_RW, the volumeset is mounted for reading or writing. Setting flags bit OF_FORCE overrides volumeset expiration checking. Setting flags bit AF_RESV automatically frees psd as soon as vname is unmounted.</p>
Return Values	<p>If the requested volumeset is already mounted on a device that is compatible with the device reservation associated with psd, the request succeeds immediately. If it is necessary to wait for the operator to mount the volume, the function blocks until the mount is performed. If there is an error, -1 is returned and terrno indicates the error condition.</p>

Name rl_vmove - schedule volumeset movement

Synopsis

```
#include "reel.h"
int rl_vmove(loc, vname)
char *loc, *vname;
```

Description rl_vmove modifies the scheduled location of all volumes in vname to loc. This causes the volume to show up on the operators library maintenance tasks as awaiting movement.

Return Values Upon successful completion, 0 is returned. If there is an error, -1 is returned and `terrno` indicates the error condition.

rl_vrec(3)

Name	rl_vrec - locates vname in the catalog and returns the VREC in vpt.
Synopsis	<pre>#include "reel.h" int rl_vrec(vname, vpt) char *vname; VREC *vpt;</pre>
Description	rls_rec - locates vname in the catalog and returns the VREC in vpt. vname volumeset name VREC volume record vpt volume pointer
Return Values	Upon successful completion, 0 is returned. If there is an error, -1 is returned and <code>terrno</code> indicates the error condition.
Notes	The volume pointer must be initialized so that it points to a volume record prior to the function call.

Name	rl_vscratch - disband a volumeset
Synopsis	<pre>#include "reel.h" int rl_vscratch(vname, force) char *vname; int force;</pre>
Description	<p>rl_vscratch returns the constituent volumes of volumeset <code>vname</code> to scratch status. If the volumes have any maintenance activities scheduled, they are not eligible for reuse until after the maintenance is performed.</p> <p>If <code>force</code> is not zero the volume will be scratched even if it has not expired. Only the owner of a volumeset may scratch it.</p>
Return Values	Upon successful completion, 0 is returned. If there is an error, -1 is returned and <code>terrno</code> indicates the error condition.

rl_vunmount(3)

Name	rl_vunmount - release a mounted volume
Synopsis	<pre>#include "reel.h" int rl_vunmount(psd, etype, edata) char *psd, etype; int edata;</pre>
Description	<p>rl_vunmount releases the volume associated with device <code>psd</code>. After <code>rl_vunmount</code>, the device associated with <code>psd</code> is free for other uses.</p> <p><code>etype</code> and <code>edata</code> may be used to modification the volume's expiration date when it is unmounted.</p>
Return Values	Upon successful completion, 0 is returned. If there is an error, -1 is returned and <code>terrno</code> indicates the error condition.
See Also	<code>rl_vmount(3)</code> .

Name	rl_wait - wait for completion of a reservation or volume mount
Synopsis	<pre>#include "reel.h" int rl_wait(type, psd) int type; char *psd;</pre>
Description	<p>rl_wait allows processes to put themselves to sleep while awaiting completion of a reservation request (see <code>rl_reserve(3)</code>) or a volume mount (see <code>rl_vaccess(3)</code>).</p> <p>type must be either <code>WAIT_RSV</code> to wait for a device reservation or <code>WAIT_MNT</code> to wait for a mount request. psd is the Pseudo Device Name of the request to wait for (when doing a <code>WAIT_RSV</code>, psd is ignored).</p>
Return Values	Upon successful completion, 0 is returned. If there is an error, -1 is returned and <code>errno</code> indicates the error condition.
See Also	<code>rl_poll(3)</code>

vrec_null(3)

Name	vrec_null - unset all fields in a VREC structure.
Synopsis	<pre>void vrec_null(vpt) VREC *vpt;</pre>
Description	<p>vrec_null sets all fields in the VREC structure indicated by vpt to a special value that indicates that the field has not been given a value.</p> <p>vrec_null should be used to prepare a VREC structure for a call to r1_valloc(). Once a structure is initialized, non-default fields need to be specified. All unspecified fields will receive default values.</p>

Name vs_cfile - return a pointer to the FREC structure of the current file section

Synopsis #include "reel.h"
FREC * vs_cfile(vsd)
int vsd;

Return Values Upon successful completion, 0 is returned. If there is an error, -1 is returned and `terrno` indicates the error condition.

vs_close(3)

Name	vs_close - terminate access to a volumeset file
Synopsis	<pre>#include "reel.h" int vs_close(vsd) int vsd;</pre>
Description	vs_close closes a volumeset file previously opened by vs_open(3). When vs_close completes, the volumeset is positioned at the beginning of the next file on the volumeset.
Return Values	Upon successful completion, vs_close returns zero. If there is an error, -1 is returned and terrno indicates the error condition.
See Also	vs_open(3).

Name vs_open - initiate access to a volumeset file

Synopsis

```
#include "reel.h"
int vs_open(vsd, oflg, fspec)
int vsd, oflg;
FSPEC *fspec;
```

Description vs_open initiates access to a file on a volumeset. vsd is a volumeset descriptor returned by a prior call to vs_vaccess(3). oflg is either T_READ to open a file for reading, or, T_WRITE to open a file for writing. fspec is a pointer to a file specification structure (defined in reel.h).

FILE SPECIFICATION

The file specification structure FSPEC is shown below.

```
typedef struct
{
    int      fselect;
    int      fvalid;
    int      flags;

    /* HDR1 */
    char      uname[13];
    char      fid[18];
    int      gen;
    int      ver;
    int      fseq;
    int      fssect;
    time_t    ctime;
    char      rtype;
    int      rdata;
    char      facc;
    int      nblocks;

    /* HDR2 */
    struct    RSPEC rspec;
    int      offset;

    /* HDR3/HDR4 */
    struct    CSPEC cspec;
    char      app[13];
    char      fcom[41];
};
```

vs_open(3)

```
char        passwd[14];
int         fmode;
char        gname[13];
} FSPEC;
```

The FSPEC structure identifies the file to open, provides label information, defines a record format, and controls record conversion. The file specification structure is used to communicate file attributes by the `vs_open(3)` function. When opening for writing, the values given in FSPEC define the attributes of the new file. When opening for reading, the current attributes of the existing file are returned in the FSPEC structure. When opening for reading the CSPEC and RSPEC attributes may also be used to override the existing CSPEC and RSPEC attributes of the file.

fselect

File selection method. Defines the file selection mode in use. One of:

FS_NULL - Unspecified. Selects the next sequential file on the volumeset.

FS_FID - File ID. File is selected by matching the value specified in `fid`, `gen` and `ver` against the corresponding values in the HDR1 label. The first file to match is selected.

FS_FSEQ - File sequence number. File is selected by its ordinal location on the volumeset. The `fseq` defines the file sequence number to select.

FS_NEXT - Next. Selects the next sequential file on the volumeset.

FS_EOT - End of Tape. This selects an open after the last file on the volumeset. It allows an application to append to a volumeset without knowing explicitly where the end of the volumeset is.

fvalid

File validity mask. The bits in this integer indicate which information is specified in the remainder of the structure. On open for write, this field identifies what values are being provided by the caller. On open for read, this field identifies what fields contain the following information:

FV_HDR1 - HDR1 information is valid.

FV_HDR2 - HDR2 information is valid.

FV_HDR3 - HDR3 information is valid.

FVL_HDR1 - HDR1 information is valid from label.

FVL_HDR2 - HDR2 information is valid from label.

FVL_HDR3 - HDR3 information is valid from label.

The HDR1 information should always be provided. If the HDR2 or HDR3 information is not provided, then the volumeset default values are used in their place.

flags

Control Bit Mask. The following bits are defined:

OF_FORCE - Override expiration exceptions.

OF_VCROSS - Enable transparent volume crossing.

uname

File Owner. This field contains the name of the file owner. This field is returned on an open for read and ignored on an open for write, newly created files are always owned by the person who created them.

fid

File Identifier. The file identifier as read or written in the HDR1 label. Recorded in bytes 5 to 27 in the HDR1 label. These values must be given if `fselect` processing is enabled using the character in `tchar`. If `tran` is '1', ASCII/EBCDIC translation is enabled. When used with an open for write, the `fid` only has an affect if the volumeset file template is set to `@F17@`.

gen

File Generation. The file generation number as read or written in the HDR1 label. Recorded as bytes 36 to 39 in the HDR1 label.

ver

File Version. The file version number as read or written in the HDR1 label. Recorded as bytes 40 to 41 in the HDR1 label.

fseq

File Sequence Number. The ordinal number of the logical file on the volumeset. The first file on the volumeset is number 1. Recorded as bytes 32 to 35 in HDR1 label. This value must be given if `fselect` is set to `FS_FSEQ`.

fsect

File Section Number. The ordinal number of this file section in the file. The first file section of a file is number 1. Recorded as bytes 28 to 31 in the HDR1 label. This value must be given if `fselect` is set to `FS_FSEQ`.

ctype

File Creation Date. When writing a file, the creation date is always set to the current date. When reading a file, the creation date found on in the HDR1 label is returned. `ctype` is always set to `RET_EXPDATE` and `cdata` is set to the creation date given as the number of days since January 1, 1970.

ctime

Creation Date. The creation date of the file in UNIX internal format. This value is returned on an open for read and ignored on an open for write.

rtype

Expiration type. This field is only used if the expiration type of the volumeset that contains the file is `RET_ONLABEL`. One of:

`RET_EXPDATE` - Expiration by date.

`RET_GDCYCLE` - Expiration by generation.

`RET_LACCESS` - Expiration by last access.

`RET_NEVER` - Never expire.

`RET_SCRATCH` - Always expired.

`RET_RETN` - Expiration in *N* days.

rdata

Expiration data. The definition of this field depends on the value assigned to `rtype` as follows:

RET_EXPDATE - Date. Days since 1/1/70.

RET_GDCYCLE - Generations. Number of generations to retain.

RET_LACCESS - Days. Expire if not accessed in given number of days.

RET_NEVER - Not used.

RET_SCRATCH - Not used.

RET_RETN - Days. Number of days to retain volume.

facc

File Access Byte. The file access byte as read or written in the HDR1 label. Recorded as byte 54 of the HDR1 label.

nblocks

Block Count. For record-oriented devices, this field contains the number of blocks in the file. For block-type devices, this field contains the number of 512-byte blocks in the file. This field is returned by an open for read of a file that has been previously cataloged and ignored on an open for write.

rspec

Record Specification. The record format of records in the file. See the description of the RSPEC structure for details.

offset

Offset. The ANSI Offset defined for the file section. The number of bytes reserved at the beginning of each block for additional information. This field is normally 0.

cspec

Conversion Specification. The record conversion used to create the file. CSPEC controls record translation performed on input and output. See the description of the CSPEC structure for details.

app

Application. The name of the application that created the file. The application name can be used on certain reports to allow a listing of all the files that belong to the application.

vs_open(3)

fcom

File Comment. Arbitrary null-terminated text provided by the user. A string up to 39 characters long.

passwd

File Password. The password associated with the file. For files that are password protected, this field contains the encrypted password on an open for read and the unencrypted (new) password on an open for write. If a password is given, the password will be required to access the file in the future.

fmode

Mode. Permission mask associated with the file.

gname

Group Name. UNIX group of the file owner. This field is returned on an open for read and ignored on an open for write.

Return Values Upon successful completion, `vs_open` returns zero, If there is an error, -1 is returned and `terrno` indicates the error condition.

See Also `vs_close(3)`, `vs_vaccess(3)`.

Name	vs_vaccess - initiate access to a volumeset
Synopsis	<pre>#include "reel.h" int vs_vaccess(psd, volset, passwd, mode, flags) char *psd, *volset, *passwd; int mode, flags;</pre>
Description	<p>vs_vaccess initiates access to volumeset <code>volset</code> on device <code>psd</code>. vs_vaccess checks that the user is permitted to access the volumeset. If the volumeset is password protected, the proper password must be given. On successful completion the volumeset is locked for the users exclusive use until released with vs_vrelease(3).</p> <p><code>psd</code></p> <p>Pseudo Device Name. The name assigned to a device during a previous call to r1_reserve(3).</p> <p><code>volset</code></p> <p>Volumeset Name. Either the name given to the volumeset when it was submitted or '.' followed by the volume ID of the first volume in the volumeset.</p> <p><code>passwd</code></p> <p>Volumeset Password. If the volumeset is password protected, <code>passwd</code> must match the password in the catalog. If the volumeset is not password protected, <code>passwd</code> is ignored.</p> <p><code>mode</code></p> <p>Access Mode. T_READ to access the volumeset for reading only. T_RW to access the volumeset for both reading and writing.</p> <p><code>flags</code></p> <p>Access Flags. AF_NREW requests that volumes in the volumeset not be rewound when they are unmounted. AF_RESV requests that device <code>psd</code> be released automatically when the volumeset is released.</p>
Return Values	<p>Upon successful completion, a small non-negative integer (the volumeset descriptor) is returned. If there is an error, -1 is returned and <code>terrno</code> indicates the error condition.</p>

vs_vaccess(3)

See Also vs_close(3), vs_open(3).

Notes Accessing a volumeset does not actually cause the first volume of the volumeset to be mounted. It is not until the volumeset is opened (vs_open(3)) for a particular file that a volume is mounted.

Name	vs_vrelease - terminate access to a volumeset
Synopsis	<pre>#include "reel.h" int vs_vrelease(vsd) int vsd;</pre>
Description	vs_vrelease terminates access to a volumeset. vsd is a volumeset descriptor returned by a prior call to vs_vaccess(3).
Return Values	Upon successful completion, vs_vrelease returns zero. If there is an error, -1 is returned and <code>errno</code> indicates the error condition.
See Also	vs_vaccess(3).

vs_vrelease(3)

File Formats (UNIX Section 4)

This appendix includes the UNIX style manual pages for each of the file format utilities.

Name	Adn - REELlibrarian device definition files
Synopsis	Library Directory: Adn/*
Description	Each file in the Adn subdirectory defines the attributes of a tape device under REELlibrarian control. The name of the file is the associated device name of the device.

Adn files contain three different record types that provide device information. Record types are distinguished by their first character; fields within records are delimited by white space. Adn files may contain records starting with 'S', 'T' and 'O'.

Each Adn file should contain one 'S' record. The 'S' record has the following fields:

S stat mach autom soft rauto

stat '1' or '0', indicating that the device is up or down respectively.

mach name of the machine the device is attached to. (char[12])

autom '1' or '0' indicating if automatic tape recognition is enabled or disabled (respectively) for the device.

soft '1' or '0' indicating if device density is software selectable when writing a tape.

rauto '1' or '0' indicating if density selection is automatic when reading a tape.

The Adn file contains one 'T' record for each format the device supports. The 'T' record has the following fields:

T fmt type rew nrew kbps effbs

fmt The format name from the `fmts(4)` file. (char[8])

type The format type from the `tapecap(5)` file. (char[20])

rew The corresponding rewind device name. (char[100])

nrew The corresponding no-rewind device name. (char[100])

kbps Estimated throughput in kilobytes per second. (integer in

ASCII form)

effbs The block size preferred for writing operations. Cannot be larger than 512 kilobytes. (integer in ASCII form)

The Adn file contains one 'o' record for any additional names the device is known by that are not mentioned as *rew* or *nrew* above. The 'o' record has the following fields:

O *dev_name*

dev_name

Name of the device. (char[12])

Examples Following is an example Adn file defining a nine track tape device on a machine named *north*:

```
S 1 north 0 1 1
T 6250 BSD.6250 /dev/rmt8 /dev/rmt12 150 32768
T 1600 BSD.1600 /dev/rmt0 /dev/rmt4 150 32768
O /dev/rmt16
O /dev/rmt20
```

const(4)

Name	<code>const</code> - REELlibrarian configurable constants
Synopsis	Library Directory: <code>const</code>
Description	<p>The <code>const</code> file defines a variety of unrelated integer constants that affect the behavior of REELlibrarian. Each line of the file defines a single constant. The format of each line is:</p> <pre><i>const_no const_val</i></pre> <p>Where:</p> <pre><i>const_no</i></pre> <p>constant number.</p> <pre><i>const_val</i></pre> <p>constant value.</p> <p>If a constant is not given a value, zero is assumed. Constant numbers are:</p> <ol style="list-style-type: none">1 Number of seconds between operator prompts.2 Reservation queue discipline.<ol style="list-style-type: none">1-First only.2-Any within First priority.3-Any within queue.3 Number of seconds between network status checks.4 1 => Require operator to provide volume ID for all mounts.5 1 => Require volume acceptance before volumes are mountable.6 Number of seconds to keep volumes premounted after release.7 Number of seconds of idle time that can build up before a device is reclaimed from a reservation.8 1 => Require volume identification before volumes are mountable.9 1 => Unknown tapes in the tape library disable Automatic

Volume Recognition

- 10 1 => Allow VSN based Automatic Volume Recognition.
- 11 1 => Trust user format specification.
- 12 1 => Conserve RPC file handles (efficiency trade off).
- 13 1 => Enable accounting log.
- 14 Not used.
- 15 Not used.
- 16 Default volume cleaning interval.
- 17 Default volume removal interval.
- 18 Default volume hard error limit.
- 19 1 => No database journalling.
- 20 1=> Require hardware write protection.
- 21 1 => Do not prompt for unmounts - released, mounted volumes are not unmounted until a mount request is generated for the same drive.
- 22 1 => Allow resource requests to be queued when all devices are down.

Examples

```
1 120
2 3
6 600
7 300
10 1
```

See Also `rlconfig(1)`.

Name `dev.default` - REELlibrarian device lock and unlock information.

Synopsis Library Directory: `dev.default`

Description The `dev.default` defines the ownership and mode characteristics for locked and unlocked devices. The file contains four white space delimited fields:

owner group lck_mode ulck_mode

owner Owner of the device when it is unlocked.

Note: When a device is locked it is always owned by the person who is using it.

group Group of the device when it is unlocked.

lck_mode

Octal number giving the mode of the device when it is locked. This should probably always be 0600.

Note: If the device is locked read-only the number given is anded with 0444.

ulck_mode

Octal number giving the mode of the device when it is unlocked.

Examples `root sys 0600 0600`

See Also `rlconfig(1)`.

Name	<code>fmts</code> - REELlibrarian format definitions and aliases
Synopsis	Library Directory: <code>fmts</code>
Description	<p>The <code>fmts</code> file defines the media types supported by REELlibrarian. It also specifies what alternative media types a media type may be converted into (for example a 1600BPI tape can be converted into a 6250BPI tape).</p> <p>Each line of the <code>fmts</code> file defines one media type as follows:</p> <pre><i>fmt clean remove errmax len [fmt1 [fmt2 ...]]</i></pre> <p>Where:</p> <p><i>fmt</i> Format name (char [8]).</p> <p><i>clean</i> Number of tape mounts before the media is scheduled for cleaning (int).</p> <p><i>remove</i> Number of tape mounts before the media is removed from the tape library (int).</p> <p><i>errmax</i> Maximum number of hard errors tolerated for an individual volume before the volume is marked as "BAD." (int)</p> <p><i>len</i> The default length in feet for reel (9-track) media types or the default capacity if megabytes for cartridge media types. (int)</p> <p><i>fmt1</i> First alternative format (char[8]).</p> <p><i>fmt2</i> Second alternative format (char[8]).</p>
Examples	<pre>800 50 200 10 2400 1600 6250 1600 50 200 10 2400 800 6250 6250 50 200 10 2400 1600 6250</pre>

jdir(4)

Name	jdir - REELlibrarian transaction journal directory definition
Synopsis	Library Directory: jdir
Description	REELlibrarian maintains a journal of all transactions to the catalog. Each day's journal entries are kept in a different file. The file jdir contains the name of the directory journal files are kept in.
Examples	/usr/spool/REEL

Name	keys - full-screen key commands
Synopsis	Library Directory: <code>rl_menus/keys</code>
Description	<p><code>keys</code> is a file that stores the key commands recognized by the REELibrarian full-screen interface. This file may only be modified by the super user.</p> <p>Note: Control characters such as <code>^R</code> must be entered as a single character. In the <code>vi</code> editor, this is accomplished by typing <code>CNTL-V</code> before typing the control character.</p>
Examples	<p>The default configuration for the <code>keys</code> file is shown below:</p> <pre>redraw=^R escape=^A help=^B top=^U form=^F toggle=^T clear=^O</pre>

loc(4)

Name	loc - REELlibrarian tape location definition file
Synopsis	Library Directory: loc
Description	<p>The loc file defines the locations where volumes may reside. An attempt to move a volume to a location not defined in loc results in an error.</p> <p>Each line of the loc file defines one location. The format of each line is:</p> <pre>loc_name mountable</pre> <p>Where:</p> <pre>loc_name</pre> <p>Identifies the name of the location. (12 chars)</p> <pre>mountable</pre> <p>Is '1' or '0', indicating that tapes at the site may be mounted.</p>
Examples	<pre>onsite 1 offsite 0 vault 0</pre>

Name	ops - REELlibrarian operator definition file
Synopsis	Library Directory: ops
Description	The ops file identifies user names designated as REELlibrarian operators. The file contains one user name per line.
Examples	user1 user2 user3

Name	reelenv - REELlibrarian configuration file
Synopsis	<code>/etc/reelenv</code> (default)
Description	<p><code>reelenv</code> stores REELbackup environment variable settings. This file should only be modified by the administrator.</p> <p>Note: These variables must be spelled exactly as shown. There must be a blank space between the variable and any value.</p> <p>Lines beginning with “#” are ignored. Variables on such lines are not considered to be present.</p>
Options	<p>Required:</p> <p><code>CLNTNAME</code> <i>hostname</i></p> <p>Name of the current machine. Default: special value <code>HOST</code> sets value to the output of <code>uname -n</code>.</p> <p><code>RLBINDIR</code> <i>absolute_directory_path</i></p> <p>Location of REELbackup binary files. Default: <code>/usr/local/bin</code>.</p> <p><code>RLLIBDIR</code> <i>absolute_directory_path</i></p> <p>Location of REELbackup data files. Default: <code>/usr/local/lib</code>.</p> <p><code>RL_MACH</code> <i>hostname</i></p> <p>REELbackup server system node name as known to the current system.</p> <p><code>RLPBASE</code> <i>RPC_number</i></p> <p>RPC program number (decimal) of the REELbackup server program <code>RB(1)</code>. The RPC program number as well as the next 100 numbers must not be used by any other RPC application on your network. Each client machine must have the same <code>RLPBASE</code> as the REEL master that serves it. If there are multiple REEL masters, each must have its own distinct <code>RLPBASE</code> that matches its own clients. Default: <code>66777770</code>.</p> <p>Optional:</p> <p><code>ISODATES</code> [<code>on</code> <code>off</code>]</p>

Set on the master and the client.

Controls the representation of the calendar date. If set to “on,” all dates are displayed in ISO 8601-compliant format. Also, all user commands that require a date as input will accept *only* ISO 8601-compliant date format.

If this variable is set to “off” or omitted, commands will accept either the original non-ISO date format or the ISO 8601 format.

`RBTDOWNTIME` *seconds*

Set on master.

Number of seconds the host is considered temporarily down by the main server before a failed operation is retried. The number must be greater than 0. Default: 600.

`RLDISABLEFORCEMNT`

Set on master.

When present, disables `forcemount` function in `rlmon`.

`RLDISABLEMNTSKIPS`

Set on master.

When present, disables `mntskip` function in `rlmon`.

`RLDISABLENEWVOL`

Set on master.

When present, disables `newvol` function in `rlmon`.

`RL_FD_LIMIT` *count*

Set on master and every client where it is needed.

`RL_FD_LIMIT` sets the minimum number of file descriptors which may be used by REEL processes. If this variable is set, REEL processes check the number against the `RLIMIT_NOFILE` system value, and the larger value is used.

`RL_LCHECKI` *seconds*

Set on master and every client where it is needed.

Specifies how often (in seconds) to access the reelenv file to see if changes have been made to tracing parameters. If nothing is

specified, the default is to check `reelenv` each second. Note that this variable can affect REELbackup performance. Set to 0 to disable checking.

`RL_MANUAL_ROTATE`

Set on master.

If present, disables automatic `volumeset` rotation. `rlrotate` must then be run at intervals to produce volume movement. Effective only on the master node.

`RLMOUNTMAX` *count*

Set on master.

This is the maximum number of times that REEL will attempt to mount a tape. For this maximum to apply, the requests must come in successively, with no more than 10 minutes separating each mount request. If the `RLMOUNTMAX` setting is exceeded, the tape is marked as lost. If this is a scratch mount, another scratch tape is substituted if one exists. If there are no more scratch tapes or the request is not for a scratch mount, the mount request is cancelled.

`RL_NOVERIFY`

Set on master.

Disables the verification step (fingerprint check) after a mount completion response.

Tracing Variables:

`RB_LEAVE_LOG`

Set on master or client where trace file is created.

Retain trace files. If this variable is not set, trace files for programs that terminate successfully are deleted.

`RLCOREDIR` *absolute_directory_path*

Set on client.

Specifies the location of the core dump directory. If not specified, default location is `/usr/tmp`.

`RLLOGDIR`

Set on master and every client where it is needed.

Location of REEL trace files (default: commented out). If this variable is present, level 9 (highest) tracing is active for all trace files, unless RLOG_LVL is set.

RL_NOTSHORT

Set on master and every client where it is needed.

Do not wrap trace files. If this variable is not present, wrapping begins after 200,000 characters.

RLOG_LVL *trace_level*

Set on master and every client where it is needed.

Determines the tracing level. If selective tracing is desired, this variable must be set to 0; this turns off tracing for all processes not explicitly requested via the RLOG_cmd variable. Acceptable values: 0 (no tracing) - 9 (verbose tracing).

RLOG_[*cmd*|*pid*] *trace_level*

Set on master and every client where it is needed.

Enables selective tracing for the specified command (*cmd*) or process ID (*pid*). RLOG_LVL must be set to 0 for selective tracing to be effective. Trace files requested explicitly via RLOG_cmd or RLOG_pid are not removed automatically, regardless of the presence or absence of RB_LEAVE_LOG. Acceptable values: 0 (no tracing) - 9 (verbose tracing).

Examples

```

RLLIBDIR /sam/local/lib
RLBINDIR /sam/local/bin
RLPBASE 667777770
RL_MACH robin
CLNTNAME HOST
RLLOGDIR /norm2/tmp/samlogs

```

Security Only the super user may edit reelenv.

See Also reel_env(8)

Name	rlmon - REELlibrarian full-screen interface support files.
Synopsis	REEL/RL_OCD/screen.info REEL/RL_OCD/messages
Description	<p>rlmon(4) provides a full-screen display of device status, pending mount/unmount requests, and operator messages. rlmon gets the information to maintain its display from two files. The file <code>screen.info</code> provides device and request information. The file <code>messages</code> provides operator messages. These files are normally located in the directory <code>REEL/RL_OCD</code>. If the environment variable <code>RL_OCD</code> is set to the path name of a directory, the files are located in the indicated directory. Both files are frequently updated by <code>RL(1)</code> to keep the information current.</p> <p>During normal operation <code>rlmon</code> alternates between looking for keystrokes from the operator and checking the modification times of the screen information files. Whenever one of the files is modified, <code>rlmon</code> reads the file and updates the display information.</p> <p>When the operator indicates some kind of action, <code>rlmon</code> uses the operator's current cursor position and the current screen information to construct a call to <code>RL</code> indicating the appropriate action.</p> <p>The file <code>screen.info</code> contains six record types. Records always occur in the same order and are distinguished by their starting character. Device information records are first and start with 'D'. The end of device information is delimited by a null record starting with 'd'. Request information records follow starting with 'A'. The end of request information is delimited by a null record starting with 'a'. The end of the file is delimited by a null record starting with 'X'. When <code>RL</code> is not running, the file starts with the string "CDOWN". If "CDOWN" is found at the beginning of the file, <code>rlmon</code> refuses to run.</p> <p>Device information records are composed of fixed length fields with the following format:</p> <pre>Dadn fmt stat vid [uid key psd]</pre>

adn Device name (char[8]).
fmt Device format (char[8]).
stat off - off-line. on - on-line. dchk - checking density. init -
 initializing. prem - premounted. umnt - user mount. rew -
 rewinding. err - error. (char[4]).
vid Volume ID. Dashes indicate no volume (char[14]).
uid User name of device reserved (char[12]).
key Reservation key (char[8]).
psd Reservation Pseudo Name (char[8]).

Note: *uid*, *key*, and *psd* are only present if a device is currently reserved.

Request information records are composed of fixed length fields with the following format:

```
amid [*] req adn rack fmt loc
```

mid Mount ID - a unique number that distinguishes this request [(char[4]).
 [*] If present indicates that a Volume ID is required to satisfy the request.
req unmount - volume unmount. dcheck - check format. mount - volume mount. (char[7])
adn Device name (char[12]).
rack Media rack number (char[12]).
fmt Requested format (char[8]).
loc Media location (char[12]).

The file message contains message records starting with 'M' terminated by a NULL record starting with 'x'. When new messages arrive, they are appended to the end of the message file. The message file is overwritten from the beginning when RL starts.

Message records have variable length fields delimited by white-space. The format of message records is:

rlmon(4)

Mmno tstamp mtext

mno Message number.

tstamp Message time stamp (UNIX internal time)

mtext Message text.

See Also [rl_dev\(3\)](#), [rl_skip\(3\)](#), [rl_done\(3\)](#).

Name rlvms_config - REELlibrarian vault definition file

Synopsis Library Directory: rlvms_config

Description rlvms_config(4) defines the vaults and slots recognized by REELlibrarian.

Note: The rlvms_config(8) command must be run after each update to rlvms_config(4) to validate the file syntax and update the VMS catalog.

rlvms_config(4) takes the form:

```

        form_factor_spec
            media_type_spec
            .
            .
        .
        .
            vault_spec
            slot_spec
            .
        .
        .

```

Strings of tabs and blanks are treated as a single blank. The pound sign (#) indicates a comment; comments extend to the end of the line.

form_factor_spec

Form Factor Specification. Each *form_factor_spec* must begin in column one and must be followed by one or more *media_type_spec* entries. Each *form_factor_spec* takes the form:

```
FORM_FACTOR form_factor_name
```

FORM_FACTOR declares that this is a form factor specification.

form_factor_name is a the name of a media type, as defined in the *fmts* file in the library directory; entries may be

up to 12 characters in length.

media_type_spec

Media Type Specification. Entries must not begin in column one (they must be preceded by white space).

media_type_spec entries list the media types that the associated form factor will accommodate. Each

media_type_spec takes the form:

```
MEDIA_TYPE media_type_name
```

MEDIA_TYPE declares that this is a media type specification.

media_type_name is the name of a media type, as defined in the *fmts* file in the library directory.

vault_spec

Vault Specification. Each *vault_spec* must begin in column one. Each *vault_spec* takes the form:

```
VAULT vault_name vault_type
```

VAULT declares that this is a vault definition record.

vault_name gives the name of the vault. Vault names may be up to 12 characters in length.

vault_type defines how volumes are slotted within the vault. *vault_type* may be one of:

VID - volumes are slotted by volume ID.

VSN - volumes are slotted by volume serial number.

SLOT - volumes are slotted by slot numbers selected by the VMS.

Each *vault_spec* is followed by zero or more *slot_spec* entries.

Note: Before configuring vaults into the vault manager, be sure to first declare them to REELlibrarian and/or REELbackup via the Storage Sites selection on the *rlconfig(8)* or *rbconfig(8)* Main Menu.

slot_spec

Slot Specification. This entry declares the slot numbers that are valid in the associated vault. *slot_spec* entries must not begin in column one (they must be preceded by white space). Each *slot_spec* takes the form:

```
SLOT form_factor_name priority first_slot
      slot_count
```

SLOT identifies a slot definition record.

form_factor_name is a the name of a form factor.

priority is an integer value between 0 and 100 that controls the order of slot allocation within the vault. In situations where there are multiple empty slots, slots with lower alphanumeric names take priority. The special priority 100 makes slots unselectable for new volumes; this allows slots to be phased out.

first_slot and *slot_count* together define the range of slots.

Note: VID-type vaults using REELlibrarian to interchange ANSI labeled tapes should not use lower case characters for the *first_slot* field; this forces vsns to be lower-case, and is not allowed by the ANSI standard.

Note: Be sure to include vault and slot definitions for **all** existing volumes in the configuration file; failure to do this will corrupt the database.

Note: When selecting a slot incrementation scheme, be sure to leave sufficient room for future expansion.

Slot Incrementation Scheme

Slot numbers are incremented both alphabetically and numerically. Columns containing upper case letters remain upper case; lower case columns remain lower case; numbered columns remain numbers. All other characters remain unchanged as the slot is incremented. See the examples below.

Examples Example 1

A *slot_spec* with *first_slot*=AA, *slot_count*=5 produces slot numbers AA, AB, AC, AD, and AE.

Example 2

A *slot_spec* with *first_slot*=2-w, *slot_count*=7 produces slot numbers 2-w, 2-x, 2-y, 2-z, 3-a, 3-b, and 3-c.

Note: The number and type of incrementing characters limit the acceptable values for *slot_count*. For example, a *slot_spec* with *first_slot*=AA will not accept a *slot_count* greater than 676; any number greater than 676 will produce an overflow. Some sample *first_slot* and maximum *slot_count* values are shown below.

```

first_slot=aa;
maximum slot_count=676 (26*26)

first_slot=000;
maximum slot_count=1000 (10*10*10)

first_slot=004;
maximum slot_count= 996 (10 * 10 * 10 - 4)

first_slot=a000;
maximum slot_count= 26000 (26 * 10 * 10 * 10)

```

Note: Redundant slot values within a single vault are unacceptable. Be sure to provide *slot_specs* that do not create duplicate slots. The following sample *first_slot/slot_count* combination would be unacceptable in the same *vault_spec* because it produces duplicates of slots b0 and b1:

```

first_slot=a7, slot_count=5
(produces slots a7, a8, a9, b0, b1)

first_slot=b0, slot_count=5
(produces slots b0, b1, b2, b3, b4)

```

Example 3

The sample *rlvms_config(4)* file shown below lists one form factor with multiple media types, and one vault with one type of slot specification.

```
FORM_FACTOR QIC #this is a
form_factor_spec

MEDIA_TYPE C20 #these are media_types
    MEDIA_TYPE C40
    MEDIA_TYPE C60
    MEDIA_TYPE C80
    MEDIA_TYPE C100
    MEDIA_TYPE C120
    MEDIA_TYPE C150

VAULT onsite SLOT #This is a
vault_spec

SLOT QIC 1 S1 10000 #this is a
slot_spec
```

Security The `rlvms_config(4)` file should never be edited manually. All changes should be performed with the `rlvms_config(8)` command.

See Also `rlvms_config(8)`, `fmt(4)`, `rlconfig(8)` `rbconfig(8)`

Name	site_exits - site exit definitions for REELlibrarian
Synopsis	REEL/Librarian/site_exits - located in the library directory
Description	<p>The <code>site_exits</code> file contains the definitions, one per drive, of the shell scripts or programs run at event points in mount processing for the drive. The three event points are: mount, unmount, and off-line. These points allow support of automatic tape loaders.</p> <p>REELlibrarian provides some standard site exit scripts for supporting automatic cartridge loaders. Others can be created and used via the site exit configuration described here.</p> <p>The <code>site_exits</code> file contains multiple single-line records of the format:</p> <pre style="margin-left: 40px;"><i>adn mount_path unmount_path offline_path</i></pre> <p>Where:</p> <ul style="list-style-type: none"> <i>adn</i> is the name of the tape drive <p style="margin-left: 80px;">Note: If the ADN name is changed in the <code>site_exits</code> file, then it must be changed in <code>rlconfig</code> to match.</p> <ul style="list-style-type: none"> <i>mount_path</i> is the pathname of the mount event exit script <i>unmount_path</i> is the pathname of the unmount event exit script <i>offline_path</i> is the pathname of the off-line event exit script <p>The special ADN name <code>DEFAULT</code> can be used to specify how tape drives not declared in a record are to be handled. If a <code>DEFAULT</code> entry is not present and a drive is unspecified by a record, then site exit processing is disabled for the drive.</p> <p>The special name <code>none</code> can be used for any of <i>mount_path</i>, <i>unmount_path</i> or <i>offline_path</i> to indicate that no script is to be executed upon that particular event for that drive.</p> <p>All site exit scripts are invoked with <code>execvp(2)</code> which means the pathnames can be either absolute pathnames or relative pathnames - the <code>PATH</code> variable in the <code>RLnet</code> environment (the network server started by <code>reel start</code> on each node) provides the search path.</p>

Each site exit script is executed with the following parameters.

```
-v vid      Volume ID
-i vsn      Volume Serial Number
-r rack     Volume rack number
-l cloc     Volume current location
-t mtype    Volume media type
-d adn      Tape drive name
-N nrew     No-rewind device for the tape drive
-R rew      Rewind device name
```

The following option is applied to only the mount and unmount site exit scripts.

```
-I reqid    Request ID.
```

Each option is produced as two `argv[]` items, one for the option symbol and one for the value so that parsing is more convenient.

The site exits are invoked as specified in the following paragraphs.

mount_path

The mount script is invoked at the time the operator sees the mount request on the monitor. If an unmount is pending for the device and the device has an unmount site exit script, then the mount site exit script is not invoked until the unmount is acknowledged. It is the responsibility of the site exit script to remove the requested volume from the device stack list if this is the desired behavior.

Note: The standard mount script `stkmount(8)` is provided in `RLBINDIR`

unmount_path

The unmount script is invoked at the time the operator sees the unmount request on the monitor.

Note: The standard unmount script `stkunmount(8)` is provided in `RLBINDIR`

site_exits(4)

offline_path

The off-line script is invoked after a user requests an unmount and the mount request system has successfully rewound the tape. It is invoked before the operator sees the unmount request. Typical uses for this script are to send the device an unload IOCTL so the volume is ejected or to make a copy of the volume to a different device.

See Also stkmount(8), stkunmount(8).

Name	trusted_hosts - defines machines recognized by the REELlibrarian server
Synopsis	REEL/Librarian/trusted_hosts - located in the library directory
Description	<p>trusted_hosts is an optional file containing a list of machine names recognized by the RL server. It is used on startup to query for remote services and during operation to authenticate requests.</p> <p>Note: If trusted_hosts is absent (default), an RPC broadcast is used on startup.</p> <p>This file may only be created or modified by the super user.</p>
Examples	<p>The following trusted_hosts file recognizes REELlibrarian on the nodes sam and robin:</p> <pre>sam robin</pre>

trusted_hosts(4)

Maintenance Commands (UNIX Section 8)

This appendix includes the UNIX style manual pages for each of the operator and administrator maintenance commands.

Name	Intro - introduction to REELlibrarian operator and administrator commands
Description	<p>REELlibrarian facilitates the use of tapes and tape drives. It allows the user to conduct ad-hoc tape sessions and to store and retrieve files on tape. REELlibrarian keeps an on-line catalog which tracks tapes and their contents.</p> <p>All user tapes are submitted to a central library which is under the control of the operator. The user can create new volumesets, via the <code>rlvcreate(1)</code> command, and access them with the <code>rlvaccess(1)</code> command. Files are read and written with the <code>rlvread(1)</code> and <code>rlvwrite(1)</code> commands.</p> <p>REELlibrarian operators receive and respond to mount requests via the Mount Request System. A full-screen operator interface is available via the command <code>rlt1m(8)</code>. Commands that may only be performed by REELlibrarian operators are identified in the SECURITY sections of the manpages.</p> <p>Certain REELlibrarian commands, such as commands that control system configuration, may only be performed by the REELlibrarian administrator. The administrator must be the super-user, or root user. A full-screen configuration interface is available via the command <code>rlconfig(8)</code>. Commands that may only be performed by the REELlibrarian administrator are identified in the SECURITY sections of the manpages.</p> <p>The following REELlibrarian operator and administrator commands are described in their own manpages:</p> <ul style="list-style-type: none"> RLS - REELlibrarian non-user commands RL - REELlibrarian Mount Request System and catalog server RLbackup - REELlibrarian catalog maintenance utilities RLdump - dump RL state information RLlicense - REELlibrarian license management program RLnet - REELlibrarian network server RLrebuild - REELlibrarian recovery program

posn_stack - position stack site exit script

reel - REELlibrarian server control program

reel_env - REELlibrarian environment reporter

rlconfig - REELlibrarian full screen configuration interface

rl_offack - drive off-line acknowledgment

rlaccept - accept a volume submitted by a user

rlcertify - certify a volume after a hard error

rlcleaned - confirm volume cleaning

rldev - bring a device up or take a device down

rldone - confirm completion of an operator media request

rlctest - test a tape drive

rlerased - confirm volume erasure

inline - report lost tapes as found

rlid - fingerprint a new library volume

rllog - REELlibrarian network logging server

rlmon - REELlibrarian full screen operator monitor

rlmoved - confirm volume movement

rlmsg - establish a communication link with the REELlibrarian server

rlop - declare tape operator

rlpremount - premount a volume

rlr - REELlibrarian report generator

rlremoved - confirm volume removal

rlreturn - operator return a volume from the library to its owner

rlrotate - execute volumeset rotation schedules

rls - operator display of device reservation queue and device status

rlskip - skip a mount request

rlsmove - move scratch tapes

rlstampddb - set timestamp on tape database

rltapevol - identify a tape via catalog match of fingerprint

rlt1m - REELlibrarian full screen operator interface

rlunlock - tape unlock utility

rlunpremount - unpremount a volume

rlvms_config - validate and update VMS catalog

rlvms_confirm - inform the VMS of the physical location of a volume

rlvms_move - initiate movement of a volume to a new vault

rlvms_report - generate VMS catalog report

rlvms_retrieve - delete volumes from the VMS catalog

rlvms_submit - submit volumes to the VMS catalog

stc_prolog - stacker prologue site exit script

stkmount - stacker mount site exit script

stkunmount - stacker unmount site exit script

See Also *REELlibrarian Master Guide.*

Name	RLS, NUTIL, RCOM, RLV, RLdmon, RLdserve, RLidle, RLinit, RLoftctl, RLrewind, RLverify - REELibrarian non-user commands
Description	These programs are used by other REELibrarian programs only.

RL(8)

Name	RL - REELlibrarian Mount Request System and catalog server
Synopsis	RL RLexit RLtest
Description	<p>The RL server coordinates all REELlibrarian activity on the network. Normally, RL is started by the <code>reel(8)</code> program.</p> <p>The command RL starts the server and automatically puts it in the background.</p> <p>The command RLexit halts the RL server. When RL is halted, all outstanding reservation requests and mount requests are canceled. When the server is restarted all devices are free and the request queue is empty.</p> <p>The command RLtest tests whether the RL server is currently running. If the server is running, RLtest exits with a non-zero status, otherwise, it exits with zero.</p> <p>If configuration changes are made (see <code>rlconfig(8)</code>) while RL is running, the changes will not take affect until the RL is halted and restarted.</p>
Security	Only the super-user may run the RL commands.
See Also	<code>reel(8)</code> , <code>RLnet(8)</code> , <code>rlconfig(8)</code> .

Name	RLbackup - REELlibrarian catalog maintenance utilities
Synopsis	RLbackup
Description	The command RLbackup should be run after each full backup of the REELlibrarian catalog (kept in the library directory Tdb/). RLbackup disposes of database journal files that are no longer necessary given that the database can be recovered from a full backup.
Security	Only the super-user may run RLbackup.
See Also	RLrebuild(8), <i>REELlibrarian Master Guide</i> .

RLdump(8)

Name	RLdump - dump RL state information.
Synopsis	RLdump
Description	<p>RLdump causes the RL(8) daemon to dump the current request and device state information into the RL trace file.</p> <p>Note: Please refer to the reelenv(4) reference page for details about enabling RL tracing.</p>
Security	Only operators may run RLdump.
See Also	reel(8).

Name	RLlicense - REELlibrarian license management program
Synopsis	RLlicense {info extend purchase node}
Description	<p>The REEL products are protected by a Network License Protection System (NLPS). The NLPS controls the expiration of demo copies and regulates the number of network machines the REELlibrarian server will support. The command <code>RLlicense</code> performs a number of license maintenance activities. The <code>RLlicense</code> argument selects the activity to perform.</p> <p>Functions that modify the software license require a modification password provided by the software vendor. Without the proper password, attempts to modify the software license will fail.</p> <p>License information is kept in the library directory file <code>license</code>. The file is protected from tampering with a checksum-password. Should the file inadvertently be corrupted, a backup copy is kept in the file <code>license.bak</code>.</p>
Options	<p><code>info</code> Print a summary of the current software license.</p> <p><code>extend</code> Extend the expiration date of a demo copy. (Requires password supplied by the software vendor)</p> <p><code>purchase</code> Converts a demo license into a purchased license. (Requires password supplied by the software vendor)</p> <p><code>node</code> Increases the number of network machines the REELlibrarian server <code>RL(8)</code> will support. (Requires password supplied by the software vendor)</p>
Security	Only the super-user may run <code>RLlicense</code> .
See Also	<i>REELlibrarian Master Guide.</i>

RLnet(8)

Name	RLnet - REELlibrarian network server
Synopsis	RLnet [-k] RLnexit RLntest
Description	<p>The RLnet daemon acts as an agent of the RL(8) server on client nodes. Normally, it is started and stopped by the reel(8) program.</p> <p>The command RLnet starts the REELlibrarian client daemon. RLnet automatically puts itself in the background.</p> <p>The RLnet daemon provides network services for the main REELlibrarian demon RL(8). RLnet must be running on each machine on the network that has devices under REELlibrarian control.</p> <p>The command RLnexit halts the RLnet server. When RLnet is halted, all outstanding reservations and requests for the node are canceled. RLnet issued with the -k option performs the same function as RLnexit.</p> <p>The command RLntest tests whether the RLnet daemon is currently running. If the server is running, RLntest exits with status zero, otherwise, it exits with non-zero status.</p>
Security	Only the super-user may run the RLnet commands.
See Also	reel(8), RL(8).

Name	RLrebuild - REELlibrarian recovery program
Synopsis	RLrebuild [-v] [-x]
Description	<p>RLrebuild reconstructs the catalog from an earlier version of the catalog and the Journal. It should be used only when reel(8) announces errors when the server programs are invoked.</p> <p>RLrebuild with no option rebuilds both the Vault Management System and the REELlibrarian databases.</p> <p>If an attempt to issue an RLrebuild results in the making of a tdb.New directory, then the rebuild failed. The tdb.New directory is this corrupted database and was created to prevent overwriting of the tdb directory containing the catalogs.</p>
Options	<p>-v Rebuilds only the Vault Management System.</p> <p>-x rebuilds only the REELlibrarian databases.</p>
Security	Only the super-user may run RLrebuild.
See Also	reel(8), RLbackup(8), rlconfig(8).

posn_stack(8)

Name	posn_stack - position stack site exit script
Synopsis	RLBINDIR/posn_stack
Description	<p>posn_stack is sourced by the site exit scripts stkmount(8) and stkunmount(8). This script supplies device-specific loading instructions of the form:</p> <pre>drive1) Log "Positioning \${a_dname}." mt -f \${a_rew} rewoffl ; ;</pre> <p>If your system recognizes the standard UNIX tape control command <code>mt</code>, you do not need to modify this script. Modify this script as needed to accommodate systems that do not recognize the <code>mt</code> command.</p>
Security	posn_stack should only be modified by the super-user.
See Also	stkmount(8), stkunmount(8)

Name	<code>reel</code> - REELlibrarian server control program
Synopsis	<code>reel [install start stop unreg]</code>
Description	<p><code>reel</code> prints a summary of the environment the REELlibrarian servers are running in. It also pings each server and reports whether the server is responding. If any server is down, then stop the servers with <code>reel stop</code> and restart them with <code>reel start</code>.</p> <p><code>reel install</code> installs the REELlibrarian NetMaster or NetClient software. It works only when run in a directory containing either the REELlibrarian NetMaster or NetClient software distribution.</p> <p><code>reel start</code> starts all of the REELlibrarian servers and pings them to make sure they are responding.</p> <p><code>reel stop</code> stops all of the REELlibrarian servers.</p> <p><code>reel unreg</code> unregisters REELlibrarian from the RPC port mapper. This should only be necessary if the server is brought down abnormally and REELlibrarian is still registered with the port mapper.</p>
Security	Only the super-user may execute <code>reel</code> with an option.
See Also	<i>REELlibrarian Master Guide</i> .

Name reel_env - REELlibrarian environment reporter

Synopsis reel_env {GROUP | LOG | MACH | OWN | PNO | RLB | RLL}

Description reel_env reports various settings of the REELlibrarian environment.

The environment settings are determined by the reelenv file; see the reelenv(4) manpage for more information.

Options

- GROUP Reports the group ID which owns the installed REELlibrarian programs.
- LOG Reports the REELlibrarian log directory.
- MACH Reports the REELlibrarian server system node name.
- OWN Reports the user ID which owns the installed REELlibrarian programs.
- PNO Reports the RPC program number of the REELlibrarian server program RL(8).
- RLB Reports the directory where the REELlibrarian programs are stored.
- RLL Reports the directory where the REELlibrarian data files are stored.

See Also reelenv(4), *REELlibrarian Master Guide*.

Name	rlconfig - REELlibrarian full screen configuration interface
Synopsis	rlconfig
Description	<p>Many features of REELlibrarian are accessible from a full screen menu/form based interface. These commands start up the interface, on-line help is available to assist learning the interface conventions.</p> <p>rlconfig is the REELlibrarian configuration program. This program can only be run by the super user. Changes made to configuration do not take effect until the RL server is restarted.</p>
Security	Only the super-user may run rlconfig.
See Also	rl(1), rltlm(8), <i>REELlibrarian Master Guide</i> .

rl_offack(8)

Name	<code>rl_offack</code> - drive off-line acknowledgment
Synopsis	<code>rl_offack adn</code>
Description	<p><code>rl_offack</code> is a utility that the offline site exit script runs to inform the server that the drive is off-line.</p> <p><i>adn</i> Assigned device Name. When REELlibrarian is installed, each device (or drive) is assigned a name. This name is used to distinguish between devices on the system. If no name is specified, <code>drive1</code> is the default</p>
Security	Only the super user may run <code>rl_offack</code> .
Notes	This command is not normally needed, but is available for the administrator to run manually to get a drive out of <code>octl</code> state.

Name	<code>rlaccept</code> - accept a volume submitted by a user
Synopsis	<code>rlaccept</code> [<i>keywords</i>] <i>vid</i>
Description	<p><code>rlaccept</code> confirms transfer of possession of a volume from a user to the library. <code>rlaccept</code> locates the volume record in the catalog, confirms the volume is awaiting acceptance, and allows specification of additional volume information.</p> <p>The <code>rlaccept</code> command can be used to override information provided by the user as well as specify things like current volume location and rack number which the user may be unable to provide. If the volume ID is modified as part of the acceptance (for example if the volume ID is modified to be the same as the rack number) the user should be told the new volume ID.</p> <p>After <code>rlaccept</code> completes, if volume identification is required, the volume will await identification. Otherwise, it is available for use.</p>
Options	<p><i>vid</i> Volume ID. A unique identifier assigned to each volume when it is submitted. Up to twelve characters long.</p> <p><i>keywords</i></p> <p>One or more keyword=value assignments. Defaults for the following keywords are the user-specified values. Keywords recognized by this command:</p> <p><code>app=</code> Application prefix. This prefix is used with file reports to limit selections to those with the specified <code>app=</code> value. <code>app</code> may be up to 12 characters long.</p> <p><code>capacity=</code></p> <p>Volume capacity in Kbytes. Specifies the volume capacity for cartridge Media types.</p> <p>Note: For nine-track (or reel) media, use the <code>length</code> keyword.</p> <p><code>conv=</code> Record conversion specification. <code>conv</code> controls conversion of records to/from tape. <code>conv</code> may be set to one of <code>text</code> for text records. <code>etext</code> for EBCDIC text records. <code>data</code> for fixed length ASCII or binary data records. <code>edata</code> for fixed length</p>

EBCDIC data records.

`dispose=`

Volume disposition. `dispose` controls when volumes leave the volumeset and what happens when they do. `dispose` may be set to `erase` and/or `retain`. If `erase` is specified, volumes that leave the volumeset (when it is truncated or scratched) are erased before they can be reallocated. If `retain` is specified the volumeset is never truncated. (Truncation occurs when the first file on a volumeset is overwritten and the new file is not large enough to span all the volumes currently in the volumeset.) To select both, specify `dispose=erase&retain`.

`ecnt=` Error count. Set error count as specified.

`finger=` File fingerprint. Set fingerprint field as specified.

`flocation=`

Free volume location. When a volumeset is truncated or scratched, the newly unattached volumes will return to the location assigned to `flocation`. The value given must be defined as a REELibrarian storage site (the command `rlr sites` will produce a list of currently defined storage sites).

`format=` Label Format. `format` may be set to one of ANSI, IBM, IBMU, TAR, CPIO or RAW.

`ftemplate=`

File Name template. The filename template is used to dynamically construct names for files written to the volumeset. The name constructed by the template is limited to seventeen (17) characters. The template consists of constant text and substitution patterns. Substitution patterns have the general form: `@sublen@` where the “@” symbol delimits the beginning and end of the substitution specification. `sub` is a character indicating what to substitute. `len` gives the length in characters of the substituted value. In principle `len` can be any number between 1 and 17, however, for some values of `sub` only a particular value of `len` is reasonable. The following

values are recognized.

@Y4@ - numeric year,

@C2@ - numeric month (Jan = 1),

@E3@ - month name,

@D2@ - day of month,

@J3@ - Julian day,

@W3@ - day of week (Sun = 1),

@H2@ - hour (24 hour clock),

@M2@ - minute,

@S2@ - second,

@F9@ - value assigned `fid` keyword on the `rlvwrite` command,

@U9@ - base name of file assigned to the `if` keyword on the `rlvwrite` command,

@G4@ - file generation,

@V2@ - file version.

Numeric values are truncated on the left. Character values are truncated on the right. If a numeric value is shorter than *len* it is padded on the left with zeros. Example on January 25 1987: `ftemplate=@D2@@E3@@Y2@G@G4@V@V2@`, results in a file name of: `25Jan89G0000V01`.

Note: Even unlabeled volume formats (IBMU, RAW, TAR and CPIO) must be assigned volume serial numbers.

`ftrack`=File Tracking Flag. `yes` causes the catalog to maintain a record for every file written to the `volumeset`. `no` disables file cataloging. Maintaining file catalogs makes it possible to generate a `volumeset` table of contents without having to mount a tape. It also improves the efficiency of accessing files on multi-volume `volumesets`.

`group`=File group. The group (from `/etc/group`) to which the file

belongs, up to 12 characters long.

`initialize=`

Volumeset Initialization. `yes` indicates the volumeset requires initialization. The first time each volume is mounted, REELlibrarian will initialize it. `no` indicates the volumeset contains existing data. The first time each volume is mounted REELlibrarian checks to make sure the Volume Serial Number on the volume label matches the Volume Serial Number in the catalog. Default: `yes`.

`length=`Length of tape in feet. Specifies the length of nine-track or reel media. Default: 2400

`offset=`Record offset. The number of bytes reserved at the beginning of each block for additional information. Offset only effects the ANSI tape format.

Warning: Under ordinary circumstances, this field should not be edited.

`passwd=`Volume Access Password. An optional password which, if specified, must be given before the volumeset can be read or written. Example: `passwd=undertow`.

`rformat=`

Record format. The new record format for the file. `rformat` takes the form: `fmt:blen:rlen`. `fmt` is one of:

`f` - fixed length records;

`fb` - fixed length, blocked records;

`v` - variable length records;

`vb` - variable length, blocked records;

`vs` - variable length, spanned records;

`vbs` - variable length, blocked, spanned records.

`u` - unformatted data.

`blen` is the block length in bytes. `rlen` is the record length in bytes. Example `rformat=fb:800:80`.

`user=` Volumeset Owner. The user ID (from `/etc/passwd`) to

which the volumeset belongs. Up to twelve characters long.

`vaccess=`

Volume Access Byte. The character assigned to `vaccess` is written as byte 11 (Volume Accessibility) in the VOL1 label. REELlibrarian attaches no particular significance to the Volume Access byte; control is provided for export of volumes to sites that do require certain values.

`vcomment=`

Volumeset comment. A comment about the volumeset. If the comment includes spaces, it must be enclosed in quotes. Up to forty characters long. Example: `vcom="my favorite volumeset"` Default: none

`vexpire=`

Volumeset Expiration Date. One of:

I - infinite (never expired)

S - scratch (always expired)

R*N* - expires *N* days after creation

AN - expires if not accessed in *N* days

L - expires when all files on the volumeset have expired

Orotsched - follows the rotation schedule *rotsched* (see `rlrcreate(1)`)

Xccyyymmdd or *Xmm/dd/yy* - expires on given date

GN - expire when there are *N* newer generations

Example: `vexpire=R30`

`vmode=` Volumeset Permission Mask. Three octal digits controlling owner, group and others (respectively) permission to the volumeset (similar to the UNIX file mode). Bit 4 controls read access. Bit 2 controls write access. Bit 1 permits viewing of the volume's catalog entry. Example: `vmode=744`. Default: 700.

`vsn=` Volume Serial Number. The volume serial number (VSN) associated with the volume. The VSN is written in the VOL1

label for IBM and ANSI tape formats. Even unlabeled volumes (IBMU, TAR, CPIO and RAW) must be given a VSN. In general REELlibrarian does not require each volume to have a unique VSN. Up to six characters long. The special value `vms` means that this value is selected by the Vault Management System.

`slocation=`

Scheduled Volume Location. If a volume's current location (`clocation`) does not equal its scheduled location. The volume will show up on the Volume maintenance report as wanting to move from `clocation` to `slocation`. The value given must be defined as a REELlibrarian storage site (the command `rlr sites` will produce a list of currently defined storage sites).

`ctype=` Volume Media Type. Media Type is an arbitrary name assigned during REELlibrarian configuration to describe Volume formats supported by devices under REELlibrarian control. For 9-track devices, this is typically the tape density (i.e. 1600, 6250, etc.). For cartridge devices, it is typically the cartridge standard (i.e. QIC11, QIC24, etc.). The given value must be a defined Media Type. The command `rlr mtype` produces a list of defined Media Types.

The following keywords may only be used by REELlibrarian operators:

`clean=` Volume cleaning count. The number of times the volume has been mounted since it was last cleaned. This value is normally maintained by REELlibrarian.

`clocation=`

Current Volume Location. The value given must be defined as a REELlibrarian storage site (the command `rlr sites` produces a list of currently defined storage sites).

`ctype=` Current Media Type. For initialized volumes, this field contains the Media Type REELlibrarian has verified is the correct media type for the volume. This field is normally maintained by REELlibrarian.

`rack=` Rack Number. The rack number identifying the volume's storage slot. The rack keyword is provided here so operators

can submit volumes in a single step. Up to twelve characters long. The special value `vms` means that this value is selected by the Vault Management System.

Note: If volume acceptance is required, the rack number is normally assigned when the operator accepts the volume.

`scratch=`

Scratch status. Sets volume scratch status (and `volumeset` if first).

`vid=` Volume ID. A unique volume identifier. For sites that maintain unique Volume Serial Numbers, `vid` is normally identical to `vsn`. By default, REELlibrarian, assigns each volume a unique volume ID by combining a four character word with a three digit number (Example: `root-496`). Up to twelve characters long. The special value `vms` means that this value is selected by the Vault Management System.

`maintenance=`

Scheduled Maintenance. Maintenance may be set to one or more of `move`, `age`, `erase`, `remove` and `clean`. Multiple items are selected by separating them with “&” (i.e. `maint=erase&clean`). `move` indicates the volume is awaiting movement. `age` causes the volume to wait *N* days after it is scratched before becoming free (*N* is defined by the pool the tape belongs to). `erase` schedules the volume for erasure. `clean` schedules the volume for cleaning.

Note: Normally the maintenance value is maintained by REELlibrarian.

`pool=` Pool membership. Every volume must belong to a pool. By default, volumes belong to the user’s private pool. Volumes may only be assigned to pools owned by the current effective user ID.

`remove=` Volume Usage Count. The number of times the volume has been mounted since it entered the catalog. This value is

rlaccept(8)

normally maintained by REELlibrarian.

status=Volume submission status. Status may be one of:

1 - submitted awaiting acceptance.

2 - accepted awaiting identification.

3 - Fully submitted.

4 - retrieved awaiting return.

Note: Normally the status value is maintained by REELlibrarian.

Security Only operators may run rlaccept.

See Also rledit(1).

Name	<code>rlcertify</code> - certify a volume after a hard error
Synopsis	<code>rlcertify adn=drive length={y n} vid [vid ...]</code>
Description	<p>REELibrarian maintains a hard error count for each volume. When a hard error occurs, the volume that experienced the hard error is marked for either recertification or removal depending on whether the error count for the volume exceeds a configurable error limit (see <code>rlconfig(8)</code>). Recertification or removal does not actually occur until the volumeset containing the volume is scratched.</p> <p>The maintenance report (see <code>rlr(1)</code>) displays volumes that are awaiting certification. Any scratch volume may be certified if there is reason to believe it may be bad.</p> <p><code>rlcertify</code> reserves the specified drive and runs through the certification procedure for each <code>vid</code>. The certification process bypasses the normal mount request system prompting the person running the command for mounts and unmounts.</p> <p>Volumes are certified by writing a single large file until EOT is encountered. A running display of the current tape position is maintained. At EOT, the system prints out the number of seconds it took to certify the volume and the average throughput (measured in kilobytes/sec). If a hard error is encountered before EOT, the command asks if it should remove the volume from the catalog.</p> <p>If the <code>length=y</code> keyword is given, <code>rlcertify</code> keeps writing until the driver signals a hard error. If the volume is truly defective, the hard error will occur somewhere in the middle of the volume. If the volume is not defective the hard error will occur when the driver encounters EOT. Whatever the cause, when the hard error is detected, the system prompts, "Accept new length?[y/n]". A yes answer will reset the length of the volume in the catalog to just before the defect. Recertification is a useful way to set the length of tapes that have been "cut down" to extend their life.</p>
Security	Only operators may run <code>rlcertify</code> .
See Also	<code>rlr(1)</code> .

rlcleaned(8)

Name	rlcleaned - confirm volume cleaning
Synopsis	rlcleaned <i>vid</i> [<i>vid</i> ...]
Description	<p>rlcleaned confirms volume cleaning. REELlibrarian resets the cleaning counter associated with each given volume.</p> <p>Normally, rlremoved is used to confirm cleaning of volumes that have exceeded the cleaning threshold established during REELlibrarian configuration.</p>
Options	<i>vid</i> A space separated list of volume IDs.
Security	Only operators may run rlcleaned.
See Also	rlr(1).

Name	rldev - bring a device up or take a device down
Synopsis	rldev status={up down reset} <i>adn</i>
Description	The operator uses rldev to inform the Mount Request System that the status of a device has changed.
Options	<p><i>status</i>=Status of Device. If set to <i>up</i>, the named device is eligible for allocation. If set to <i>down</i>, the named device is no longer eligible for allocation (rldev automatically cancels any outstanding mounts or device reservation requests dependent on the named device). If set to <i>reset</i>, REELlibrarian cancels the current reservation for the named device and makes the device available for use by others.</p> <p><i>adn</i> ADN. The name of the affected device.</p>
Security	Only operators may run rldev.
See Also	rls(8).

Name	rldone - confirm completion of an operator media request
Synopsis	rldone [force={yes no}] [new={yes no}] [mid= <i>mountid</i>] [rack= <i>RackNo</i>] [vs= <i>VSN</i>] <i>vid</i>
Description	<p>rldone confirms that a requested mount, unmount, density check, or volume addition has been satisfied.</p> <p>If REELlibrarian does not have a fingerprint for a volume (this is the case the first time the volume is mounted), volume ID must be given to positively identify the volume. If the <code>force</code> or <code>new</code> keywords are given, volume ID is required.</p>
Options	<p><i>vid</i> Volume ID. A unique identifier assigned to each volume when it is submitted.</p> <p><i>keywords</i></p> <p>One or more keyword=value assignments. Keywords recognized by this command:</p> <p><code>force=</code>If set to <code>yes</code>, REELlibrarian overrides volume verification error checking. The <code>force</code> option is occasionally necessary to mount a volume after a system crash if the volume database does not have the correct volume fingerprint. The system still tries to fingerprint the tape but it does not require that the recorded fingerprint match the observed fingerprint. The <i>vid</i> must be given with the <code>force=</code> keyword. Default: <code>no</code>.</p> <p>Warning: This is a dangerous tool and should be used judiciously and sparingly. Forcing a mount inappropriately can have grave consequences for backup data and catalog integrity.</p> <p><code>new=</code> If set to <code>yes</code>, REELlibrarian does not electronically identify the volume as it assumes the tape is new to the system. This keyword should only be used if the system cannot obtain a fingerprint on the volume via the <code>force=</code> keyword. The <i>vid</i> must be used with the <code>new=</code> keyword. Default: <code>no</code>.</p> <p><code>mid=</code> Mount ID. By default, the command assumes the request at the top of the request list. <code>mid</code> may be used to indicate a</p>

request other than the top.

The following keywords are optional and an empty value is acceptable:

`rack=` Rack Number. Enter a rack number of up to twelve characters. This keyword is only used when confirming a volume addition (i.e. `VADD`).

`vsn=` Volume Serial Number. Enter a volume serial number of up to eight characters. This keyword is only used when confirming a volume addition (i.e. `VADD`).

Security Only operators may run `rldone`.

See Also `rlop(8)`, `rlskip(8)`, `rls(8)`.

Name	rldtest - test a tape drive
Synopsis	rldtest
Description	<p>The tape device interface provided by different manufactures of UNIX machines varies greatly. To accommodate different device interfaces and capabilities, REELlibrarian accesses tape devices through a layer of software that provides a uniform device interface. For example, if a device does not support the Back Space File IOCTL, the tape is rewound and forward spaced to the proper file. This layer of software is similar in purpose to the <code>curses</code> interface UNIX provides for accessing terminals. Like <code>curses</code>, the tape access software requires information describing the abilities of various tape devices. The file “<code>tapecap</code>” in the REELlibrarian library directory is a database of Tape Device Descriptions (TDDs).</p> <p>The <code>rldtest</code> command is used to test the <code>tapecap</code> entry configured for a specific drive. When run, <code>rldtest</code> prompts for the name of an ADN to test. During testing, it is not unusual for the UNIX device driver to occasionally complain as <code>rldtest</code> exceeds the device’s abilities. As long as the test continues to run, these complaints should be ignored. A successful test indicates that the drive is appropriately configured.</p> <p>Note: If the test fails for an EXABYTE or other 8mm device, please try, in succession, the <code>tapecap</code> types XBYTE, XBYTEB, and XBYTEC. Use the <code>rlconfig</code> program to modify the drive’s configuration to use these types.</p>
Options	None
Security	Only the super-user may run <code>rldtest</code> .
See Also	<code>rlconfig(8)</code> , <i>REELlibrarian Master Guide</i> .

Name rlerased - confirm volume erasure

Synopsis rlerased *vid* [*vid ...*]

Description rlerased is used by the operator to inform REELlibrarian that a volume has been erased.

Options *vid* A space separated list of volume IDs.

Security Only operators may run rlerased.

See Also rlr(1).

rlfound(8)

Name	<code>rlfound</code> - report lost tapes as found
Synopsis	<code>rlfound vid [vid ...]</code>
Description	<code>rlfound</code> updates the catalog to indicate that the named volumes are locatable and not lost. When a tape mount is skipped (<code>rlskip</code>), the associated volume is marked as LOST and is thereafter ignored by REELlibrarian. The <code>rlfound</code> command restores volumes to the LIB state so that they are usable by REELlibrarian.
Options	<code>vid</code> Volume ID. The named volume has its status changed to LIB.
Security	Only operators may run <code>rlfound</code> .
See Also	<code>rlr(1)</code> , <code>rlskip(8)</code> .

- Name** `rlid` - fingerprint a new library volume
- Synopsis** `rlid adn=drive vid [vid ...]`
- Description** `rlid` fingerprints newly accepted library volumes. A volume's fingerprint is determined by mounting the volume and constructing the fingerprint from the volume's data.
- If more than one volume is specified, then the program prompts the operator to mount one after another on the drive.
- The drive specified by the `adn=` keyword is reserved via the Mount Request System.
- Options**
- `vid` Volume ID List. A space separated list of volume IDs.
- keywords*
- Volume Keywords. One or more keyword=value assignments.
- `adn=` The name of the drive on which the volumes are mounted.
- Security** Only operators may run `rlid`.
- See Also** `rlaccept(8)`.
- Notes** The `rlid` command should only be issued once when a tape volume is entered into the Library. If issued on a tape volume with data already on it, then that data may be rendered useless.

Name	rlllog - REELlibrarian network logging server
Synopsis	<pre>rlllog [-f <i>filename</i>] rlllog [-m] rlllog [-n] rlllog [-p] rlllog [-x] rlllog_change rlllog_exit rlllog_test</pre>
Description	<p>The <code>rlllog</code> server coordinates all REELlibrarian logging activities on the network. Normally, <code>rlllog</code> is started by the <code>reel(8)</code> program.</p> <p>The command <code>rlllog</code> starts the server and automatically puts it in the background. When issued with the <code>-m</code> option, this command displays a log file that is useful for diagnosing problems.</p> <p><code>rlllog_change</code> restarts the log file. To save an existing log file, rename the existing file before restarting.</p> <p><code>rlllog_exit</code> halts the <code>rlllog</code> server and is the equivalent of the <code>-x</code> option.</p> <p><code>rlllog_test</code> tests whether the <code>rlllog</code> server is currently running. If the server is running, <code>rlllog_test</code> exits with status zero, otherwise, it exits with non-zero status. This is the equivalent of the <code>-p</code> option.</p>
Options	<pre>-f Display archived log file. This option allows you to specify the file's path. -m Display log file. -n Starts a new log file if there is no existing log file. Move or rename the existing log file before using this option. -p Test if <code>rlllog</code> is running. -x Exit <code>rlllog</code>.</pre>
Security	Only operators and administrators may use <code>rlllog(8)</code> .
See Also	<code>reel(8)</code>

Name	rlmon - REELlibrarian full screen operator monitor
Synopsis	rlmon [-i <i>interval</i>] [-m]
Description	rlmon is the REELlibrarian full screen operator monitor. From this screen, the operator can monitor device activity as well as receive mount and unmount requests.
Options	<p>-i Prompt Interval. The monitor prompts the operator with a beep every interval seconds when a displayed request is awaiting action. interval is in seconds and is by default set to 20. If interval is 0, then the prompting is disabled.</p> <p>-m No Messages. This flag disables the message facility by which the operator receives messages from the system or users.</p>
Security	Only operators may run rlmon.
See Also	<i>REELlibrarian Master Guide.</i>
Notes	rlmon can only be run on the NetMaster.

rlmoved(8)

Name rlmoved - confirm volume movement

Synopsis rlmoved location=*to_loc* vid[/*to_slot*]
[vid[/*to_slot*]...]

rlmoved is used by the operator to inform REELlibrarian that a volume has been moved.

Options *vid* A space separated list of volume IDs.
to_slot
The specific slot to move the volume to.
location=
Volume Location. Location for the volume. The value given must be defined as a REELlibrarian storage site (the command `rlr sites` will produce a list of currently defined storage sites).

Security Only operators may run rlmoved.

See Also rlr(1).

Name	<code>rlmsg</code> - establish a communication link with the REELibrarian server
Synopsis	<code>rlmsg [-x]</code>
Description	<p><code>rlmsg</code> establishes a communication link with REELibrarian. The communication link is used by REELibrarian to communicate device status changes to the user. During normal operation, communication links are created automatically. This command is only necessary when a communication link is desired before it would normally be created.</p> <p>If <code>rlmsg</code> is given with the <code>-x</code> option, the existing communication link between the server and user is broken.</p>
Security	Only operators may run <code>rlmsg</code> .
See Also	<code>rls(8)</code>

rlop(8)

Name	rlop - declare tape operator
Synopsis	rlop [-r] [-x]
Description	<p>rlop identifies the user as the current operator at the user's current terminal. All subsequent action requests are directed to the user for action.</p> <p>It is possible for more than one user to be an operator. If there are multiple operators, each will see the same requests.</p> <p>If the -r and -x options are both used, then the user is removed from the reserve list of operators.</p>
Options	<p>-m Request only Operator mount messages.</p> <p>-r Reserve List. Add the operator to the reserve list. Should the current operator fail to service a request, the system pages operators on its reserve list to step forward and become the new current operator.</p> <p>-x Cancel. Cancels the user's status as the current operator. If used with the -r option, it removes the user from the reserve list.</p>
Security	Only operators may run rlop.
See Also	r1mon(8), r1s(8).

Name	rlpremount - premount a volume
Synopsis	rlpremount [<i>keywords</i>] <i>vid</i>
Description	<p>rlpremount informs REELlibrarian a volume has been premounted on the device given by the adn keyword.</p> <p>When a user requests a volume and the volume is premounted, the request is satisfied immediately.</p>
Options	<p><i>vid</i> Volume ID. A unique identifier assigned to each volume when it is submitted.</p> <p><i>keywords</i></p> <p>One or more keyword=value assignments. Keywords recognized by this command:</p> <p>adn= Assigned Device Name. The named assigned to the device during REELlibrarian configuration.</p> <p>force=If set to <i>yes</i>, REELlibrarian overrides volume verification error checking. The <i>force</i> option is occasionally necessary to mount a volume after a system crash if the catalog does not have the correct volume fingerprint. Default: <i>no</i>.</p> <p>write=Write Flag. By default, volumes are mounted read-only. Specifying <i>write=yes</i> allows both read and write access.</p>
Security	Only operators may run rlpreamount.
See Also	rlop(8), rlskip(8), rls(8)

Name	rlr - REELlibrarian report generator
Synopsis	rlr <i>keywords</i> {dinfo finfo flist maint pinfo plist pvolumes rinfo rlist sites tapecap types vault vcontent vinfo vinventory vlist vsflist vslist vsvlist}
Description	rlr generates the requested report.
Options	<p><i>keywords</i></p> <p>Keyword Parameters. One or more keyword=value definitions. Some reports require keywords to identify the item being reported. Listed below are the allowed keywords and their descriptions.</p> <p><i>action</i>=Used with the maint report, it can be set equal to: move, remove, erase. Example: <i>action</i>=move.</p> <p><i>adn</i>= Used with the dinfo report, it should be set to a drive's ADN. Example: <i>adn</i>=drive1.</p> <p><i>app</i>= Application prefix. Used with the volume and file listing reports, <i>app</i> limits selection to items that begin with the given string. Example: <i>app</i>=89.</p> <p><i>entry</i>=Tapecap entry name. This keyword is used with the tapecap report to request configuration details on a specific tapecap entry. Example: <i>entry</i>=DAT.</p> <p><i>fid</i>= File ID. Identifies the file for the finfo report.</p> <p><i>force</i>=Used with the maint option in conjunction with <i>action</i>=move, causes each volume for which a 'move' status is reported to have its clocation set to be the same as its slocation, as if <i>rbmoved loc=<slocation> <vid></i> had been invoked. The keyword requires operator privileges.</p>
Examples	<p>rlr action=move maint — Generates list of movements.</p> <p>rlr action=move force=y maint – Generates list and changes clocation.</p> <p><i>fsect</i>=File section number. Identifies the file by the file section</p>

number for the `finfo` report.

`fseq=` File sequence number. Identifies the file by order in the volumeset for the `finfo` report.

`full=` Specifies full report format or parsable format (with fields delimited by colons). Default is `full=yes` which specifies full format; `full=no` specifies parsable format.

`location=`

Site name. Used with the `maint` report `location` limits the volume movement report to include only volumes moving from the given location. Example: `loc=onsite`

`old=` Displays reports in pre-REELlibrarian 3.3 format (`old=yes`). Default is `no`.

`pool=` Pool Name. Used with the `pinfo`, `vinventory`, and `pvolumes` reports. Example: `pool=private`

`rack=` Limits display to volumes in a particular rack. Used with the `vinventory` report.

`rotation=`

Rotation schedule. Used with the `rinfo` report.

`user=` Specifies a user ID by which to conduct the search. Example: `user=guest`

`vid=` Volume ID. Identifies the volume for the `vinfo` and `vault` reports.

`volset=` Volumeset name. Identifies the volumeset for the `vinfo`, `finfo`, `vsflist`, and `vsvlist` report. Example: `volset=volset1`

The following reports are available:

`dinfo` Displays the configuration for the drive specified with the `adn` keyword. Keywords accepted: `adn` and `full`.

`finfo` Displays the full catalog entry for the tape file identified by the `fid` keyword. Keywords accepted: `fid`, `fseq`, `fsect`, `full`, `volset`.

`flist` Lists all the files owned by the user. Keywords accepted: `app`,

- full, user.
- maint** Generates a list of library management tasks as defined by the `action` keyword. If `action` is omitted, then all tasks are displayed. Keywords accepted: `action`, `force`, `full`, `location`.
- rinfo** Lists rotation schedule information. Keywords accepted: `full` and `rotation`.
- rlist** Lists rotation schedules. Keywords accepted: `full` and `user`.
- sites** Lists all library sites. Keyword accepted is: `full`.
- pinfo** Displays the catalog entry for the pool named with the `pool` keyword. Keywords accepted: `pool`, `full`.
- plist** Lists all pools owned by the user. Keywords accepted: `full`, `user`.
- pvolumes**
- Lists a summary of volumes currently assigned to the specified pool. Keywords accepted: `full`, `pool`.
- tapecap** Tapecap. Lists all configured tapecap entries. When issued with the `entry=` keyword, lists configuration details for a specific tapecap entry. Keyword accepted is: `entry`.
- types** Lists all defined media types. Accepted keyword is: `full`.
- vault** Displays where volumesets will be moved. Keywords accepted: `full` and `vid`.
- vcontent**
- Displays contents of specified vault. Keywords accepted: `full` and `location`.
- vinfo** Displays the catalog entry for the volume named by the `volset` or `vid` keywords. Keywords accepted: `full`, `vid`, `volset`.
- vinventory**
- Lists all of the volumes in the catalog sorted by location and rack number. Keywords accepted: `full`, `location`, `pool`,

rack.

`vlist` Lists a summary of all selected volumes. Keywords accepted: `full` and `user`.

`vsflist` Lists all files on the volumeset named by the `volset` keyword. Keywords accepted: `full` and `volset`.

`vslist` Lists all volumesets belonging to the user. Keywords accepted: `app`, `full`, `user`.

`vsvlist` Reports the member volumes of the volumeset named by the `volset` keyword. Keywords accepted: `full` and `volset`.

See Also Chapter 5, *Reports*, *REELlibrarian Master Guide*.

rlremoved(8)

Name	rlremoved - confirm volume removal
Synopsis	rlremoved <i>keyword vid [vid ...]</i>
Description	<p>rlremoved is used by the operator to inform REELlibrarian that volumes have been removed from the library. Removed volumes are deleted from the catalog.</p> <p>Normally, rlremoved is used to confirm removal of volumes that have exceeded the usage threshold established during REELlibrarian configuration.</p>
Options	<p><i>vid</i> A space separated list of volume IDs.</p> <p><i>keyword</i></p> <p><i>nvols</i>= Number of tapes. Specifies the number of tapes to remove. The <i>vid</i> values (if given) are incremented for each repetition. Default: 1.</p>
Security	Only operators may run rlremoved.
See Also	rlr(1).

Name	<code>rlreturn</code> - operator return a volume from the library to its owner
Synopsis	<code>rlreturn [vid=vid] receipt</code>
Description	<p><code>rlreturn</code> confirms transfer of volume possession from the library to its owner. <code>rlreturn</code> locates the volume corresponding to receipt in the volume catalog. If <i>vid</i> is given, <code>rlreturn</code> matches it against the <i>vid</i> of the volume record; If the two match, the volume record is deleted and the volume may be returned, otherwise, an error message is printed.</p> <p>If <i>vid</i> is not given, <code>rlreturn</code> prints the current location and rack number for the volume associated with receipt. To return a volume, <code>rlreturn</code> is normally given twice. First without <i>vid</i> to find out where the volume is located. Second with <i>vid</i> to confirm return of the proper volume.</p>
Options	<p><i>receipt</i> Volume Receipt. A unique identifier assigned to the volume by <code>rlvretrieve(1)</code> or <code>rlpretrieve(1)</code>.</p> <p><i>keywords</i></p> <p>One or more keyword=value assignments. Keywords recognized by this command:</p> <p><i>vid</i>= Volume ID. A unique volume identifier. For sites that maintain unique Volume Serial Numbers, <i>vid</i> is normally identical to <i>vsn</i>. By default, REELlibrarian, assigns each volume a unique volume ID by combining a four character word with a three digit number (Example: <code>root-496</code>).</p>
Security	Only operators may run <code>rlreturn</code> .
See Also	<code>rlpretrieve(1)</code> , <code>rlvretrieve(1)</code> .

rlrotate(8)

Name	rlrotate - execute volumeset rotation schedules
Synopsis	rlrotate -V
Description	<p>rlrotate updates the volume catalog by scheduling volumes and volumesets for movements based on their rotation schedules.</p> <p>rlrotate is the sole means by which to enforce the movement of volumes. It can be run manually or automatically by <code>cron</code>. It should be run once a day to keep the rotation schedules on time.</p>
Options	<p>-V Verbose. Specify this option to generate output when you run this command.</p>
Security	Only operators may run <code>rlrotate</code> .
See Also	<code>rlrcreate(1)</code> , <code>rlrdelete(1)</code> .

Name	<code>rls</code> - operator display of device reservation queue and device status
Synopsis	<code>rls</code>
Description	<p><code>rls</code> provides a summary of queued device reservation requests and current device status. <code>rls</code> is only available to operators.</p> <p>The Request Q portion of the report displays all user requests waiting for drive assignments. The column headings are explained below.</p> <p>User The name of the user who generated the request.</p> <p>Prio The priority of the request - can be a digit from '0' to '9'. '0' is the highest priority, '9' the lowest.</p> <p>Key The user's key name - usually this is the same as the UID column.</p> <p>Machine The computer node from which the request was originated.</p> <p>Format The media type of the volume's involved.</p> <p>Adn If the request specified a particular drive, it is listed here.</p> <p>Psd The pseudonym is an arbitrary named assigned by the user to the drive. Usually, it is "default."</p> <p>The Device Status portion of the report lists each device and its current disposition. The column headings are explained below.</p> <p>ADN The drive's name. If it is followed by a minus sign ("-") then the drive is down and unavailable - see <code>rldev(8)</code> for more information.</p> <p>Type The media type of the volume currently mounted on the drive.</p> <p>Stat The status indicator.</p> <p><code>vrfy</code> - the mounted volume is being electronically identified.</p> <p><code>on</code> - a volume is mounted but not under user control.</p> <p><code>off</code> - the drive is off-line; there is not a mounted volume.</p> <p><code>dchk</code> - the drive is undergoing a density check.</p> <p><code>user</code> - the mounted volume is under user control.</p> <p><code>rew</code> - the mounted volume is being rewound prior to being</p>

rls(8)

unmounted.

prem - a volume has been premounted - it is unknown the volume's disposition for write protection.

wprem - a volume has been premounted for reading or writing.

rprem - a volume has been premounted for reading only.

oct1 - the drive is about to enter the off state.

VID The volume ID of the mounted volume. Eight dashes ("-----") are displayed if the drive is without a mounted volume.

UID The user name assigned the drive.

Key The user's key name - usually this is the same as the UID column.

Psd The pseudonym is an arbitrary named assigned by the user to the drive. Usually, it is "default."

Security Only operators may run rls.

See Also r1dev(8), r1done(8), r1mon(8), r1skip(8), r1status(1).

Name	<code>rlskip</code> - skip a mount request
Synopsis	<code>rlskip [mid=<i>mountid</i>] <i>text</i></code>
Description	<code>rlskip</code> skips a mount request. The given text is forwarded to the user who initiated the mount request as an explanation (i.e. “could not find tape...”).
Options	<i>keywords</i> One or more keyword=value assignments. Keywords recognized by this command: <code>mid=</code> Mount ID. By default, the command assumes the request at the top of the request list. <code>mid</code> may be used to indicate a request other than the top.
Security	Only operators may run <code>rlskip</code> .
See Also	<code>rldone(8)</code> , <code>rlmon(8)</code> , <code>rls(8)</code> .

rlsmove(8)

Name	rlsmove - move scratch tapes
Synopsis	<code>rlsmove type=mtype length=len pool=pname nvol=n from=loc to=loc</code>
Description	<p>rlsmove schedules for movement a specified number of scratch tapes. It displays the selected tapes' rack numbers and volume IDs. The keyword parameters specify the type of tapes, the targeted pool and the source and destination locations.</p> <p>If the designated pool does not have as many scratch tapes as requested, rlsmove selects the scratch tapes available and reports the shortfall.</p> <p>Once the tapes are actually moved, run the rlmoved command to update the catalog.</p> <p>The "rlr action=move maint" command displays all currently pending tape movements.</p>
Options	<p><code>type=</code> Media type. Specifies the media type of the scratch tapes. There is no specified default.</p> <p><code>length=</code>Length. Specifies the length or capacity of the scratch tapes. There is no specified default.</p> <p><code>pool=</code> Pool name. The scratch tapes are taken from the designated pool name. There is no specified default.</p> <p><code>nvol=</code> Number of tapes. Specifies the number of tapes to be selected. Default: 1.</p> <p><code>from=</code> From location. Specifies the location from which the scratch tapes are selected. There is no specified default.</p> <p><code>to=</code> To location. Specifies the destination for the selected scratch tapes. There is no specified default.</p>
Security	Only operators may run rlsmove.
See Also	rlmoved(8), rlr(1), rlvedit(1), rlvmove(1).

Name	<code>rlstampddb</code> - change timestamp of Tdb files to re-apply journal transactions
Synopsis	<code>rlstampddb <i>crashfilepathname</i></code>
Description	<p><code>rlstampddb</code> sets the timestamp on the Tdb so <code>RLrebuild</code> can re-apply transactions after disaster recovery. <code>REELbackup</code> sets the tape database (Tdb) timestamp to the current time when a crash recovery is executed; <code>REELlibrarian</code> may have executed journal transactions after the crashfile was created. To recover the journal transactions, the timestamp of the Tdb files must be set back to the time of crashfile creation. <code>rlstampddb</code> extracts the timestamp from the crashfile and timestamps the Tdb, reporting both the old and new timestamps.</p> <p>After crashfile recovery, execute the following commands as superuser:</p> <pre>reelb stop rlstampddb <i>crashfilepathname</i> RLrebuild reelb start</pre> <p>Note: If the Tdb is backed up and restored with a mechanism other than <code>REELbackup</code>, you must run <code>RLrebuild</code> before server startup, or the servers will update the Tdb timestamp and render the journal transactions obsolete. If you need to reset the timestamp, <code>rlstampddb</code> may be run against any file.</p> <p>If the argument to <code>rlstampddb</code> is a crashfile, the date is extracted from within. If the argument is not a crashfile, the modification time of the file will be used to timestamp the Tdb.</p>
Options	<p><code><i>crashfilepathname</i></code> -</p> <p>The pathname of the crashfile used for the recovery</p>
Security	Only the super-user may run <code>rlstampddb</code> .
See Also	<code>RLrebuild(8)</code> .

Name rltapevol - identify a tape via catalog match of fingerprint

Synopsis rltapevol [-aef] [-i mid] [-w secs] adn [vid]

rltapevol returns the volume ID of the tape if the REELlibrarian catalog matches the tape fingerprint, the message UNLABELED TAPE if is not a REELlibrarian tape, or the message LOOKUP ERROR if the catalog either does not match the tape fingerprint or there are several instances of this tape's fingerprint or volume serial number (vsn) in the catalog. In this case, turn on RL Level 5 tracing (see "Using Tracing Facilities" in Chapter 2 of the *REELlibrarian Master Guide*), rerun rltapevol, and look for the keyword matches in the RL trace file to determine the duplicate entries by volume ID (vid).

The -a option displays the volume's catalog record or the fingerprint if the catalog does not match the tape fingerprint.

Options

- adn The tape device on which the tape is mounted.
- vid Volume ID. A unique volume identifier.
- mid Mount ID.
- secs Number of seconds.
- a Display the volume record information found on the tape.
- e Display environment variables used for autoloader scripts.
- f Display tape fingerprint.
- i Display the volume record of a particular mount ID (mid) if checking for a mount cancel.
- w Specify the number of seconds to try to identify file volume of the tape.

Security Only operators may run rltapevol.

See Also rlid(8).

Notes rltapevol cannot access a remote device and should be used locally.

Name	r1t1m - REELlibrarian full screen operator interface
Synopsis	r1t1m
Description	<p>Many features of REELlibrarian are accessible from a full screen menu/form based interface. These commands start up the interface, on-line help is available to assist learning the interface conventions.</p> <p>r1t1m brings up the operator's tape library management screen. From this screen, the operator can perform a variety of library management tasks.</p>
Security	Only operators may run r1t1m.
See Also	r1(1), rlconfig(8), <i>REELlibrarian Master Guide</i> .

rlunlock(8)

Name	<code>rlunlock</code> - tape unlock utility
Synopsis	<code>rlunlock</code> <code>rlunlock vid [...]</code>
Description	The <code>rlunlock</code> utility, when run with no options, lists the volume ID, user ID, and key of all currently locked volumesets. When issued with one or more <code>vid</code> options, <code>rlunlock</code> unlocks the specified volumeset(s).
Options	<code>vid</code> Volume ID. Unlock the specified volumeset(s). <code>vid</code> should be the ID of the first volume in the volumeset.
Security	Only the super user may run <code>rlunlock</code> .
Notes	This command is not normally needed; it is available in the unlikely event that tape locks are lost. This utility is only available to the super user.

Name	<code>rlunpremount</code> - unpremount a volume
Synopsis	<code>rlunpremount adn</code>
Description	<p><code>rlunpremount</code> informs REELlibrarian the premounted volume on device <code>adn</code> has been taken off-line and unmounted.</p> <p>During normal operation, operators unmount tapes when they are prompted. The <code>rlunpremount</code> command gives the operator a way to unmount tapes before the REELlibrarian decides they should be unmounted.</p>
Security	Only operators may run <code>rlunpremount</code> .
See Also	<code>rlop(8)</code> , <code>rlpremount(8)</code> , <code>rlskip(8)</code> , <code>rls(8)</code> .

rlvms_config(8)

Name	rlvms_config - validate and update the VMS catalog
Synopsis	rlvms_config
Description	<p>rlvms_config(8) must be run to update the vault definition file rlvms_config(4); rlvms_config(8) validates the file syntax, creates a new rlvms_config(4) file, and updates the VMS catalog.</p> <p>If the syntax of rlvms_config(4) is incorrect, rlvms_config(8) reports the syntax problems and exits.</p> <p>If the syntax of rlvms_config(4) is correct, rlvms_config(8) checks that the new configuration declares vault and slot values for all existing volumes.</p> <p>If an error is encountered, rlvms_config(8) exits without updating the VMS catalog; if no error conditions are encountered, the VMS catalog is updated to reflect the new configuration.</p>
Examples	<p>The recommended procedure for making changes to the configuration is to make a copy of rlvms_config(4) with the rlvms_report(8) command, edit the new file, and then use the edited file for input to the rlvms_config(8) command. See the example commands below:</p> <ol style="list-style-type: none">1. rlvms_report config > newfile2. vi newfile (make any necessary changes to this file)3. rlvms_config newfile
Security	Only the super-user may run rlvms_config(8).
See Also	rlvms_config(4), rlvms_report(8)
Notes	The server must be up to run rlvms_config(8).

Name rlvms_confirm - inform the VMS of the physical location of a volume

Synopsis rlvms_confirm [-V] [-s *slot*] [-v *vault*] *vid* [*vid...*]
 rlvms_confirm [-V] -m *rlvms_move_file*
 rlvms_confirm [-V] -r *vid_loc_file*

Description

Warning: THIS COMMAND IS FOR DIAGNOSTIC USE ONLY. DO NOT ISSUE THIS COMMAND FROM THE COMMAND LINE. Instead, use the `rbmoved(8)` command to confirm volume movement. This command performs a variety of tasks, including calling `rlvms_confirm(8)`.

`rlvms_confirm(8)` updates the VMS catalog with the data provided to the command. If an error occurs when performing multiple confirmations, `rlvms_confirm(8)` will attempt to confirm the remaining entries. The exit status will reflect the first error encountered.

If `rlvms_confirm(8)` is performed without a previous `rlvms_move(8)`, the vault and slot values for the specified volume are immediately updated with the values provided by `rlvms_confirm(8)`; this use of the command requires the `-v vault` option.

If `rlvms_confirm(8)` specifies the vault and slot specified by the move, the move is completed. If the vault and/or slot supplied by `rlvms_confirm(8)` are different than those supplied by `rlvms_move(8)`, the vault and slot of the volume are updated to the value specified by `rlvms_confirm(8)`, but the move is still pending. If the vault and/or slot are not specified by the `rlvms_confirm(8)`, these values are assumed to match those supplied by the `rlvms_move(8)` command.

After volume movement is confirmed with `rlvms_confirm(8)`, the volume is no longer scheduled for movement, even in cases where the confirm indicates a slot other than the slot scheduled by the `rlvms_move(8)` request.

rlvms_confirm(8)

- Options**
- `-V` Verbose. Returns the list of volumes confirmed to standard out.
 - `-s slot` Updates the *slot* value of the VMS catalog record for the specified volume.
 - `-v vault`
Updates the *vault* value of the VMS catalog record for the specified volume.
 - `vid` The volume ID of the volume to update.
 - `-m rlvms_move_file`
Updates the VMS catalog according to the contents of *rlvms_move_file*. *rlvms_move_file* is a file previously generated with `rlvms_move`.
Note: To read from standard input, substitute a dash (“-”) for *rlvms_move_file*. An example of this syntax is shown in the EXAMPLES section of this manpage.
 - `-r vid_loc_file`
Updates the VMS catalog according to the contents of *vid_loc_file*. *vid_loc_file* is a file of the format:
vid vault/slot
Note: To read from standard input, substitute a dash (“-”) for *vid_loc_file*. An example of this syntax is shown in the EXAMPLES section of this manpage.

Examples Update the VMS catalog to confirm the movement of a specific volume:

```
rlvms_confirm AO536b
```

Update the VMS catalog record for the specified volume, specifying slot and vault:

```
rlvms_confirm -s 100 -v onsite 232934
```

Update the VMS catalog record with the contents of the file created by `rlvms_move`:

```
rlvms_confirm -m move_output
```

Update the VMS catalog record with the contents of file *newfile*.

```
cat newfile | rlvms_confirm -r -  
    or  
    rlvms_confirm -r - < newfile
```

Security Only operators and administrators may run `rlvms_confirm(8)`.

Warning: THIS COMMAND IS FOR DIAGNOSTIC USE ONLY. DO NOT ISSUE THIS COMMAND FROM THE COMMAND LINE.

See Also `rlvms_move(8)`, `rlmoved(8)`

Notes This command is for diagnostic use only.

rlvms_move(8)

Name rlvms_move - initiate movement of a volume to a new vault

Synopsis rlvms_move -v *to_vault* [-p][-o] *vid[/slot]* [...]
rlvms_move -v *to_vault* [-p][-o] -f *filename*

Description

Warning: THIS COMMAND IS FOR DIAGNOSTIC USE ONLY. DO NOT ISSUE THIS COMMAND FROM THE COMMAND LINE.

rlvms_move(8) initiates movement of a volume to a new vault by selecting an unoccupied slot in the destination vault specified by *to_vault*.

The command sends the following information to standard output:

```
vid current_vault/slot destination_vault/slot
```

Options

-v *to_vault*

Destination vault. This vault must be defined in the file *rlvms_config(4)*.

Note: If the volume's current vault is the same as the *to_vault* value, the requested move will occur in one step (-o is assumed).

-p Pull. Confirms that the volume has been pulled from its current slot. This frees the slot for immediate use by another volume.

Note: Not specifying this option means that the slot being vacated will not be available until the ensuing *rlvms_confirm(8)* has been performed.

-o One-step volume movement. The volume's current slot is freed and the volume is assigned a new current slot in the vault *to_vault*. It is unnecessary to confirm movement with *rlvms_confirm(8)*.

vid[/slot]

Volume ID and destination slot of the volume or volumes to

move.

`-f filename`

List file. *filename* is a file that supplies `rlvms_move` with a list of volumes for movement, one per line, specified by volume ID. Each line should be of the syntax:

`vid[slot]`

Examples Move the specified volumes to the vault offsite:

```
rlvms_move -v offsite 123456 123457 123458
```

Move the volumes listed in file `vollist` to the vault offsite; bypass the confirmation step:

```
rlvms_move -v offsite -o -f vollist
```

`vollist` is a text file containing a list of volume IDs and slots, one per line.

Security Only operators and administrators may run `rlvms_move(8)`.

Warning: THIS COMMAND IS FOR DIAGNOSTIC USE ONLY. DO NOT ISSUE THIS COMMAND FROM THE COMMAND LINE.

See Also `rlvms_confirm(8)`, `rlvms_config(4)`, `rlvmove(8)`

Notes If an `rlvms_move(8)` is specified for a volume that is already scheduled for movement, the most recent `rlvms_move(8)` request takes precedence and the old move is cancelled.

You may direct the output of this command to a file for later use by the `rlvms_confirm(8)` command. An example of this syntax is shown in the EXAMPLE section of this manpage.

Name	rlvms_report - generate VMS catalog reports
Synopsis	<pre>rlvms_report config rlvms_report [-v vault][-I][{-a -e}] inventory rlvms_report [-f from_vault][-t to_vault][-I] move</pre>
Description	<p>rlvms_report produces configuration, inventory and movement reports from the VMS catalog.</p> <p>The configuration report produces a summary of the current vault configuration. The format of this report is acceptable as a rlvms_config(4) vault definition file.</p> <p>The inventory report displays a list of volumes currently residing in the specified vault. If <i>vault</i> is not given, output is produced for all vaults.</p> <p>The move report lists volumes awaiting movement. With no arguments, a report is generated for each vault showing volumes scheduled to move from the vault.</p>
Options	<pre>config Configuration report. inventory Inventory report. The format of this report is shown in the EXAMPLES section, below. move Movement pending report. -v vault List inventory for specified vault. Only valid with the inventory option. -I VID sort. Sort the report by volume ID rather than slot. Valid with the inventory and move options. -a All slots. Print all slots, including empty slots, slots pending movement, and reserved slots. Only valid with the inventory option. -e Empty slots. Print empty slots only. Only valid with the</pre>

inventory option.

`-f from_vault`

From vault only. List volumes moving from *from_vault* only. Only valid with the `move` option.

`-t to_vault`

To vault only. List only volumes moving to *to_vault*. When given with `-f from_vault`, only volumes moving from *from_vault* to *to_vault* are included. Only valid with the `move` option.

Examples Generate a report of all the slots and their contents:

`rlvms_report -a inventory`

Sample output for this command is shown below.

```
VAULT: slot_vault
  slot      sts  media/form  vid      vsn
  -----  ---  -----  -
  aaa      emp  EXABYTE
  aab      emp  EXABYTE
  aaf      emp  EXABYTE
  bbb      emp  QIC
  bbc      emp  QIC
  bbc      occ  C100      A44_33B  A44_XYZ
  bbd      res  C100
  bbe      vac  C100      bbe      bbe_A
  bbe      emp  QIC
  bbf      emp  QIC
VAULT: vid_vault
  slot      sts  media/form  vid      vsn
  -----  ---  -----  -
  aaa      emp  QIC
  aab      emp  QIC
  aac      emp  QIC
  aad      emp  QIC
  aae      emp  QIC
  aaf      emp  QIC
  bbb      emp  QIC
  bbc      emp  QIC
  bbd      occ  C100      bbd      bbd
  bbe      acc  C100      bbe      bbe_A
  bbf      acc  C100      bbf      bbf_B
  bbg      emp  QIC
```

ccc	emp	QIC
ccd	emp	QIC
cce	emp	QIC

Column Definitions:

slot Slot name within the vault.

sts Status of the slot; may be one of:

occ - occupied, no movement pending

acc - accepting, pending volume movement into this slot

vac - vacating, pending volume movement from this slot

res - reserved, (VID vaults with slots only). There is a volume in the system with this id, but it is not in this vault. This slot is not available for use by other volumes. This is a result of the definition of VID vault: each volume must be slotted by its VID.

emp - empty, slot available

media/form

Media type or form factor. If the slot is empty, the form factor the slot accommodates is shown; if the slot is occupied the media type of the associated volume is shown.

Note: It is possible for the media type of reserved slots to be incompatible with the form factor of the slot; this inconsistency indicates a flaw in the initial vms configuration for the site.

vid Volume ID of the volume associated with the given slot. Empty slots have no volume associated with them.

vsn Volume serial number of the volume associated with the given slot. Empty slots have no volume associated with them.

Generate a report of empty slots in the vault onsite:

```
rlvms_report -v onsite -e inventory
```

Generate a report of volumes moving from the onsite vault to the offsite vault:

```
rlvms_report -f onsite -t offsite move
```

Note: Move reports take the same form as inventory reports, but only moving volumes (slots in state `vac` or `acc`) are shown.

Security Only operators and administrators may run `rlvms_report(8)`.

See Also `rlvms_config(4)`, `rlr(8)`

rlvms_retrieve(8)

Name	rlvms_retrieve - delete volumes from the VMS catalog
Synopsis	rlvms_retrieve [-V] <i>vid</i> [...] rlvms_retrieve [-V] -f <i>filename</i>
Description	<p>Warning: THIS COMMAND IS FOR DIAGNOSTIC USE ONLY. DO NOT ISSUE THIS COMMAND FROM THE COMMAND LINE. Instead, use the <code>rbremoved(8)</code> command to remove volumes from the catalog. This command performs a variety of tasks, including calling <code>rlvms_retrieve(8)</code>.</p> <p><code>rlvms_retrieve(8)</code> deletes specified volumes from the VMS catalog. Slots that contained the deleted volumes are marked as empty.</p>
Options	<p>-v Verbose. Returns a list of volumes retrieved to standard out.</p> <p><i>vid</i> Volume ID of the volume or volumes to remove from the catalog.</p> <p>-f <i>filename</i></p> <p>Remove volumes listed by volume ID in the file <i>filename</i>.</p>
Examples	<p>Remove the volumes specified in the file <i>filename</i> from the VMS catalog:</p> <pre>rlvms_retrieve -f <i>filename</i></pre> <p><i>filename</i> is a text file that contains a list of volume IDs, one per line.</p>
Security	<p>Only operators and administrators may run <code>rlvms_retrieve(8)</code>.</p> <p>Warning: THIS COMMAND IS FOR DIAGNOSTIC USE ONLY. DO NOT ISSUE THIS COMMAND FROM THE COMMAND LINE.</p>
See Also	<code>rlvms_submit(8)</code> , <code>rlvretrieve(1)</code> , <code>rlreturn(8)</code> , <code>rlremoved(8)</code>
Notes	None

Name rlvms_submit - submit volumes to the VMS catalog

Synopsis rlvms_submit [-V] -v *vault* -m *media_type*
vid[/vsn[/slot]] [...]

rlvms_submit [-V] -v *vault* -m *media_type* -N *nvol*

rlvms_submit [-V] -v *vault* -f -m *media_type filename*

Description

Warning: THIS COMMAND IS FOR DIAGNOSTIC USE ONLY. DO NOT ISSUE THIS COMMAND FROM THE COMMAND LINE. Instead, use the `rbadd(8)` command to manage volume submission. This command performs a variety of tasks, including calling `rlvms_submit(8)`.

`rlvms_submit` enters new volumes into the VMS catalog and assigns each *vid* a slot number within the specified *vault*. If *vsn* is omitted it defaults to *vid*.

If multiple volumes are submitted in a single command, each submission succeeds or fails as if it were submitted separately. The exit status of the command reflects the first error. Error messages are sent to standard error.

Options -V Verbose. Print a list of successful submissions to standard out.

-v *vault*

Vault. Submit volumes to the vault *vault*, as defined in the file `rlvms_config(4)`.

-m *media_type*

Media type. This must be a media type as defined in the `form_factor_spec` in the file `rlvms_config(4)`.

vid Volume ID. The volume ID of the volume to submit.

Note: This form of the command is typically used when submitting to vaults of type `SLOT`, as defined in the `rlvms_config(4)` file. Volumes in `SLOT`-type vaults are slotted by slot numbers selected by the VMS. If specifying a `VID`-type vault, the *vid* of the volume is required; if

specifying a VSN-type vault, the *vs*n of the volume is required.

*vs*n Volume serial number. (Required for VSN-type vaults.)

slot Slot. The VMS verifies that the slot specified is currently unoccupied.

Note: If *slot* is not specified, it is assigned by the VMS.

-N *nv*o1 Number of volumes. Submits the number of unnamed volumes specified by *nv*o1. This option outputs the *assigned_slot* for each volume, one per line.

-f *filename*

Submit volumes listed in file *filename*. File *filename* lists volumes, one per line, with the following syntax:

```
vid[/vs
```

```
slot]]
```

This option outputs the *assigned_slot* for each volume, one per line.

Note: If *slot* is not specified, it is assigned by the VMS. If *slot* is given, VMS checks that the slot is currently unoccupied before assigning it a *vid*.

Note: This option may be used for VID- or VSN-type vaults when you wish the VMS to select an available slot which will then be assigned as the new volume's *vid*.

Examples Submit an 800mm tape with the volume ID 123456 to the onsite vault:

```
rlvms_submit -v onsite -m 800 123456
```

Submit 12 unnamed volumes to the offsite vault:

```
rlvms_submit -v offsite -m 800 -N 12
```

Submit volumes listed in the file *vol_list* to the onsite vault:

```
rlvms_submit -v onsite -m 800 -f vol_list
```

- Security** Only operators and administrators may run rlvms_submit.
- Warning: THIS COMMAND IS FOR DIAGNOSTIC USE ONLY. DO NOT ISSUE THIS COMMAND FROM THE COMMAND LINE.**
- See Also** rlvms_retrieve(8), rlvms_config(4), rlpsubmit(1), rlvsubmit(1), rlaccept(8)
- Notes** None

stc_prolog(8)

Name	stc_prolog - stacker prologue script
Synopsis	RLBINDIR/stc_prolog
Description	stc_prolog is sourced by the site exit scripts stkmount(8) and stkunmount(8). This script parses options for later use by the stkmount(8) and stkunmount(8) scripts. Do not modify this script.
Security	stc_prolog should never be modified.
See Also	stkmount(8), stkunmount(8)

Name	stkmount - stacker mount script
Synopsis	RLBINDIR/stkmount
Description	<p>stkmount and stkunmount are the standard site exit scripts for supporting automatic cartridge loaders. To configure REELlibrarian for use with autoloaders, modify the <code>site_exits(4)</code> file to specify <code>stkmount</code> for <code>mount_path</code>.</p> <p>stkmount sources the <code>posn_stack(8)</code> and <code>stc_prolog(8)</code> scripts. If your system does not support the UNIX tape control command <code>mt</code>, you will need to modify <code>posn_stack(8)</code> to include a suitable command.</p>
Security	stkmount should never be modified; instead, write an alternative mount script and reference it via the <code>site_exits(4)</code> file.
See Also	<code>site_exits(4)</code> , <code>posn_stack(8)</code> , <code>stc_prolog(8)</code> , <code>stkunmount(8)</code>

stkunmount(8)

Name	stkunmount - stacker unmount script
Synopsis	RLBINDIR/stkunmount
Description	<p>stkmount and stkunmount are the standard site exit scripts for supporting automatic cartridge loaders. To configure REELlibrarian for use with autoloaders, modify the <code>site_exits(4)</code> file to specify <code>stkunmount</code> for <code>unmount_path</code>.</p> <p>stkunmount sources the <code>posn_stack(8)</code> and <code>stc_prolog(8)</code> scripts. If your system does not support the UNIX tape control command <code>mt</code>, you will need to modify <code>posn_stack(8)</code> to include a suitable command.</p>
Security	stkunmount should never be modified; instead, write an alternative unmount script and reference it via the <code>site_exits(4)</code> file.
See Also	<code>site_exits(4)</code> , <code>posn_stack(8)</code> , <code>stc_prolog(8)</code> , <code>stkumount(8)</code>

System Messages

This appendix lists and defines all of the REELlibrarian system messages.

REELlibrarian employs two types of system messages: log messages and user messages. Log messages are written to the REELlibrarian log; the log resides in the file `REEL/LOG` in the REELlibrarian library directory. User messages appear on screen while using REELlibrarian programs.

Log Messages

All REELlibrarian log messages are listed below. For each message, the following information is listed:

- The message code.
- The text of the message as it will appear in the log.
- The message class. These are defined below.
- The program or programs that can generate the message. This information is supplied so that logging can be turned on for specific programs.
- The situation that led to the generation of the message.
- Advice for solving the problem, if applicable.

Message Classes

All REELlibrarian log messages fall into one of the message classes listed below.

- PRFM - performance error
- MDIA - media error
- TOPR - tape operation message
- SYS0 - unrecoverable system error
- SYS1 - user recoverable system error
- SYS2 - auto recoverable system error

Message List

1000: **Server started (*version*)**
CLASS: SYS0 (unrecoverable system error)
PROGRAMS: RL
SITUATION: Informational message
ADVICE: None

1001: **RL Server: Normal Exit**
CLASS: SYS0 (unrecoverable system error)
PROGRAMS: RL
SITUATION: Informational message (server status)
ADVICE: None

- 1002: **Reserved** (*user:key:device*)(*hostname:\media:adn*)
CLASS: TOPR (tape operation message)
PROGRAMS: RL
SITUATION: Informational message (request status)
ADVICE: None
- 1003: **Freed** (*user:key:device*)
CLASS: TOPR (tape operation message)
PROGRAMS: RL
SITUATION: Informational message (request status)
ADVICE: None
- 1004: **Mount** (*user:key:device*) (*adn:media:vid*)
CLASS: TOPR (tape operation message)
PROGRAMS: RL
SITUATION: Informational message (request status)
ADVICE: None
- 1005: **Unmount** (*user:key:device*) (*adn:media:vid*)
CLASS: TOPR (tape operation message)
PROGRAMS: RL
SITUATION: Informational message (request status)
ADVICE: None
- 1006: **Hangup** (*user:key*)
CLASS: TOPR (tape operation message)
PROGRAMS: RL
SITUATION: Operator logged off.
ADVICE: None

- 1007:** **Device adn Updated**
 CLASS: TOPR (tape operation message)
 PROGRAMS: RL
 SITUATION: Informational message (device status)
 ADVICE: None
- 1008:** **Device adn Added**
 CLASS: SYS1 (user recoverable system error)
 PROGRAMS: RL
 SITUATION: Informational message (device status)
 ADVICE: None
- 1009:** **Device adn Deleted**
 CLASS: TOPR (tape operation message)
 PROGRAMS: RL
 SITUATION: Informational message (device status)
 ADVICE: None
- 1010:** **Node up (*hostname*)**
 CLASS: SYS1 (user recoverable system error)
 PROGRAMS: RL
 SITUATION: Informational message (Host status)
 ADVICE: None
- 1011:** **Node down (*hostname*)**
 CLASS: SYS1 (user recoverable system error)
 PROGRAMS: RL
 SITUATION: Informational message (Host status)
 ADVICE: None

- 1012:** **Reservation slammed (*adn:REEL_message*)**
 CLASS: TOPR (tape operation message)
 PROGRAMS: RL
 SITUATION: Device access was interrupted by failure or intention.
 ADVICE: Correct condition according to *REEL_message*.
- 1013:** **Online (*adn:media:fingerprint*)**
 CLASS: TOPR (tape operation message)
 PROGRAMS: RL
 SITUATION: Informational message (device status)
 ADVICE: None
- 1014:** **Device offline *adn***
 CLASS: TOPR (tape operation message)
 PROGRAMS: RL
 SITUATION: Informational message (device status)
 ADVICE: None
- 1015:** **Request queued (*user:key:device*) *prio***
 CLASS: TOPR (tape operation message)
 PROGRAMS: RL
 SITUATION: Informational message (request status)
 ADVICE: None
- 1016:** **Request cancel (*user:key*)**
 CLASS: TOPR (tape operation message)
 PROGRAMS: RL
 SITUATION: Informational message (request status)
 ADVICE: None

1017: Idle device *adn*
CLASS: TOPR (tape operation message)
PROGRAMS: RL
SITUATION: Informational message (request status)
ADVICE: None

1018: Mount cancel *adn*
CLASS: TOPR (tape operation message)
PROGRAMS: RL
SITUATION: Informational message (request status)
ADVICE: None

1019: Op mount (*adn:vid:mode*)
CLASS: TOPR (tape operation message)
PROGRAMS: RL
SITUATION: Informational message (request status)
ADVICE: None

1020: Op mount skip (*adn:vid:comment*)
CLASS: TOPR (tape operation message)
PROGRAMS: RL
SITUATION: Informational message (request status)
ADVICE: None

1021: Op vadd skip
CLASS: TOPR (tape operation message)
PROGRAMS: RL
SITUATION: Informational message (device status)
ADVICE: None

- 1022:** **Tape init (*adn:vid:status*)**
 CLASS: TOPR (tape operation message)
 PROGRAMS: RL
 SITUATION: Informational message (request status)
 ADVICE: None
- 1023:** **Rewound (*adn:vid:status*)**
 CLASS: TOPR (tape operation message)
 PROGRAMS: RL
 SITUATION: Informational message (request status)
 ADVICE: None
- 1024:** **Device down *adn***
 CLASS: TOPR (tape operation message)
 PROGRAMS: RL
 SITUATION: Informational message (request status)
 ADVICE: None
- 1025:** **Device up *adn***
 CLASS: SYS2 (auto recoverable system error)
 PROGRAMS: RL
 SITUATION: Informational message (device status)
 ADVICE: None
- 1026:** **Device reset *adn***
 CLASS: SYS2 (auto recoverable system error)
 PROGRAMS: RL
 SITUATION: Informational message (device status)
 ADVICE: None

1027: **Node up (RB.version RL.version)**
 CLASS: SYS0 (unrecoverable system error)
 PROGRAMS: RLnet
 SITUATION: Informational message
 ADVICE: None

1028 **Node connected (hostname)**
 CLASS: SYS1 (user recoverable system error)
 PROGRAMS: RLnet
 SITUATION: Informational message
 ADVICE: None

1029: **Node connection broken hostname**
 CLASS: SYS1 (user recoverable system error)
 PROGRAMS: RLnet
 SITUATION: Informational message
 ADVICE: None

1030: **RLnet (hostname) Normal Exit**
 CLASS: SYS1 (user recoverable system error)
 PROGRAMS: RLnet
 SITUATION: Informational message
 ADVICE: None

1049: **Hard Error (vid:file#:block#:op)**
 CLASS: MDIA (media error)
 PROGRAMS: RL
 SITUATION: Failure writing tape
 ADVICE: Check media and device. Clean device if occurring frequently.

- 1052: **Writer startup failed (*Unix_message*)**
CLASS: SYS1 (user recoverable system error)
PROGRAMS: RLnet
SITUATION: Process invocation error
ADVICE: Proceed according to *Unix_message* and restart backup.
- 1053: **Reader startup failed (*Unix_message*)**
CLASS: SYS1 (user recoverable system error)
PROGRAMS: RLnet
SITUATION: Process invocation error
ADVICE: Proceed according to *Unix_message* and restart backup.
- 1054: **Extractor startup failed (*Unix_message*)**
CLASS: SYS1 (user recoverable system error)
PROGRAMS: RLnet
SITUATION: Process invocation error
ADVICE: Proceed according to *Unix_message* and restart backup.
- 1055: **Recover startup failed (*Unix_message*)**
CLASS: SYS1 (user recoverable system error)
PROGRAMS: RLnet
SITUATION: Process invocation error
ADVICE: Proceed according to *Unix_message* and restart backup.

- 1125:** **Could not get device info for *adn***
 (*REEL_message*)
CLASS: TOPR (tape operation message)
PROGRAMS: RLverify (client utility)
SITUATION: Configuration error
ADVICE: Check device configuration.
- 1126:** **Could not open *adn:file* (*Unix_message*)**
CLASS: TOPR (tape operation message)
PROGRAMS: RLverify (client utility)
SITUATION: Device access error
ADVICE: Check REEL device configuration and accessibility.
- 1127:** **Could not get device info for *adn***
 (*REEL_message*)
CLASS: TOPR (tape operation message)
PROGRAMS: RLidle (client utility)
SITUATION: Configuration error
ADVICE: Check device configuration.
- 1128:** **Idle Limit Exceeded (*adn*)**
CLASS: TOPR (tape operation message)
PROGRAMS: RLidle (client utility)
SITUATION: Informational message
ADVICE: None

- 1129: **Could not get device info for *adn***
 (*REEL_message*)
CLASS: TOPR (tape operation message)
PROGRAMS: RLrewind (client utility)
SITUATION: Configuration error
ADVICE: Check device configuration.
- 1130: **Could not open *adn:file* (*Unix_message*)**
CLASS: TOPR (tape operation message)
PROGRAMS: RLrewind (client utility)
SITUATION: Device access error
ADVICE: Check REEL device configuration and accessibility.
- 1131: **Could not get device info for *adn***
 (*REEL_message*)
CLASS: TOPR (tape operation message)
PROGRAMS: RLinit (client utility)
SITUATION: Configuration error
ADVICE: Check device configuration.
- 1132: **Initialization failed *adn:media***
 (*REEL_message*) (*Unix_message*)
CLASS: TOPR (tape operation message)
PROGRAMS: RLinit (client utility)
SITUATION: Configuration error
ADVICE: Check REEL device configuration and accessibility.

- 1134:** **License Limit Exceeded**
CLASS: SYS0 (unrecoverable system error)
PROGRAMS: RLnet
SITUATION: Host configuration error
ADVICE: Check network configuration and license status with
 RLlicense info.
- 1157:** **Scratch volname**
CLASS: SYS2 (auto recoverable system error)
PROGRAMS: RL
SITUATION: Informational message
ADVICE: None
- 1233:** **Broadcast attempt failed (Unix_message)**
CLASS: SYS0 (unrecoverable system error)
PROGRAMS: RL
SITUATION: General system call failure
ADVICE: Correct according to *Unix_message*.
- 1234:** **Bcast fork failed (Unix_message)**
CLASS: SYS0 (unrecoverable system error)
PROGRAMS: RLbcast
SITUATION: UNIX fork () system call failure
ADVICE: Provide for additional processes per
 Unix_message.

1235: **BROADCASTING NOT AVAILABLE**
CLASS: SYS0 (unrecoverable system error)
PROGRAMS: RLbcast
SITUATION: Informational message
ADVICE: None

1236: **Host response from *network_id***
CLASS: SYS0 (unrecoverable system error)
PROGRAMS: RLbcast
SITUATION: Informational message
ADVICE: None

1253: **RL database corrupted, run RLrebuild**
CLASS: TOPR (tape operation message)
PROGRAMS: RL
SITUATION: Librarian database corruption
ADVICE: Rebuild database

1254: **Warning: DB key corrupted (*key*), run RLrebuild**
CLASS: SYS2 (auto recoverable system error)
PROGRAMS: RL
SITUATION: Librarian database corruption
ADVICE: Rebuild database

1300: ******* run RLrebuild *******
CLASS: SYS0 (unrecoverable system error)
PROGRAMS: RL
SITUATION: Librarian Database Corruption
ADVICE: Rebuild database

- 1301:** **RLOP:** *REEL_message*
CLASS: TOPR (tape operation message)
PROGRAMS: RL
SITUATION: Informational message (echo of operator message)
ADVICE: None
- 1317:** **Volume vid format format:** *Corrupted fingerprint
fprint*
CLASS: TOPR (tape operation message)
PROGRAMS: RL
SITUATION: Fingerprinting produced an invalid result
ADVICE: Device configuration should be reviewed. Rerun
 rldtest.
- 1330:** **RLnet (hostname) abort:** *REEL_message*
CLASS: SYS0 (unrecoverable system error)
PROGRAMS: RLnet
SITUATION: REEL system failure
ADVICE: Restart RLnet daemon
- 1397:** **RL Server Abort:** *REEL_message*
CLASS: SYS0 (unrecoverable system error)
PROGRAMS: RL
SITUATION: REEL system failure
ADVICE: Call customer support

5165: Popen failed for *command (Unix_message)*
CLASS: SYS0 (unrecoverable system error)
PROGRAMS: RL application
SITUATION: Popen () call failure
ADVICE: Correct according to *Unix_message*

5166: Rlwait id failed (*Unix_message*)
CLASS: TOPR (tape operation message)
PROGRAMS: RL application
SITUATION: rlcwait not printing info
ADVICE: To be determined

5167: Rlwait status failed (*Unix_message*)
CLASS: TOPR (tape operation message)
PROGRAMS: RL application
SITUATION: rlcwait not printing info
ADVICE: To be determined.

5175: Error reading *site_exits* file - *site_exits* disabled
CLASS: TOPR (tape operation message)
PROGRAMS: RL
SITUATION: *site_exits* file access failed
ADVICE: Check the
\$RLLIBDIR/REEL/Librarian/*site_exits* file
to verify it exist and that it is an executable shell
script.

5236: **REEL function: RPC message**
CLASS: SYS1 (user recoverable system error)
PROGRAMS: RL
SITUATION: Network communication failure contacting server
ADVICE: Increase system resources and/or network resources

5237: **RLnet connection established hostname**
CLASS: SYS0 (unrecoverable system error)
PROGRAMS: RL
SITUATION: Informational message
ADVICE: None

5319: **hostname: Class [A-E] license**
CLASS: SYS0 (unrecoverable system error)
PROGRAMS: RL
SITUATION: Informational message
ADVICE: None

5320: **hostname: Class [A-E] license freed**
CLASS: SYS0 (unrecoverable system error)
PROGRAMS: RL
SITUATION: Informational message
ADVICE: None

5321: **Missing Rewind Device(file) for adn adn**
CLASS: SYS0 (unrecoverable system error)
PROGRAMS: RLnet
SITUATION: Device configuration error
ADVICE: Check device configuration, especially location

5322: **Missing Norewind Device(*file*) for adn adn**
CLASS: SYS0 (unrecoverable system error)
PROGRAMS: RLnet
SITUATION: Device configuration error
ADVICE: Check device configuration, especially location

5349: **Volume *vid*: File ID expected:*fid* actual:*fid***
CLASS: MDIA (media error)
PROGRAMS: RL application
SITUATION: Tape label field mismatch
ADVICE: Verify label information and correct database.

5350: **Volume *vid*: Generation expected:*val* actual:*val***
CLASS: MDIA (media error)
PROGRAMS: RL application
SITUATION: Tape label field mismatch
ADVICE: Verify label information and correct database.

5351: **Volume *vid*: Version expected:*val* actual:*val***
CLASS: MDIA (media error)
PROGRAMS: RL application
SITUATION: Tape label field mismatch
ADVICE: Verify label information and correct database.

5352: **Volume *vid*: File Sequence expected:*val* actual:*val***
CLASS: MDIA (media error)
PROGRAMS: RL application
SITUATION: Tape label field mismatch
ADVICE: Verify label information and correct database.

5353: Volume vid: File Section expected:val actual:val
CLASS: MDIA (media error)
PROGRAMS: RL application
SITUATION: Tape label field mismatch
ADVICE: Verify label information and correct database.

5354: Volume vid: Offset expected:val actual:val
CLASS: MDIA (media error)
PROGRAMS: RL application
SITUATION: Tape label field mismatch
ADVICE: Verify label information and correct database.

5355: Volume vid: Conversion: expected does not match actual
CLASS: MDIA (media error)
PROGRAMS: RL application
SITUATION: Tape label field mismatch
ADVICE: Verify label information and correct database.

5356: Volume vid: Block Count expected:val actual:val
CLASS: MDIA (media error)
PROGRAMS: RL application
SITUATION: Tape label field mismatch
ADVICE: Verify label information and correct database.

5357: Volume vid: EOF not found where expected
CLASS: MDIA (media error)
PROGRAMS: RL application
SITUATION: Tape label field mismatch
ADVICE: Verify label information and correct database.

- 5358: **Device *adn*: *open(args)*: *Unix_message***
 CLASS: MDIA (media error)
 PROGRAMS: RL application or utility
 SITUATION: Failure initiating media access
 ADVICE: Ensure that correct media is inserted, check special
 file permissions.
- 5359: **Device *adn*: *read(args)*: *Unix_message***
 CLASS: MDIA (media error)
 PROGRAMS: RL application or utility
 SITUATION: Failure while reading media
 ADVICE: Check media and device for condition and cleanliness
 and then rerun `rldtest`
- 5360: **Device *adn*: *write(args)*: *Unix_message***
 CLASS: MDIA (media error)
 PROGRAMS: RL application or utility
 SITUATION: Failure while reading media
 ADVICE: Check media and device for condition and cleanliness
 and then rerun `rldtest`
- 5361: **Device *adn*: *ioctl(REW)*: *Unix_message***
 CLASS: MDIA (media error)
 PROGRAMS: RL application or utility
 SITUATION: Failure rewinding media
 ADVICE: Rerun `rldtest`

5362: Device *adn*: *ioctl(FSF val): Unix_message*
CLASS: MDIA (media error)
PROGRAMS: RL application or utility
SITUATION: Failure forward spacing media
ADVICE: Rerun *rldtest*

5363: Device *adn*: *ioctl(BSF val): Unix_message*
CLASS: MDIA (media error)
PROGRAMS: RL application or utility
SITUATION: Failure backward spacing media
ADVICE: Rerun *rldtest*

5364: Device *adn*: *Unexpected write result val (expected val)*
CLASS: MDIA (media error)
PROGRAMS: RL
SITUATION: Partial failure when writing past EOT
ADVICE: Review device configuration and then rerun *rldtest*

5385: *rb_vscreate* **RB server error**
CLASS: SYS1
PROGRAMS: *rb_vscreate*
SITUATION: Failure creating a volume set
ADVICE: None

5388: *<hostname>*: **notification limit exceeded**
CLASS: SYS0
PROGRAMS: RL
SITUATION: Specified host has not checked in with the master
ADVICE: Check the client

- 5389:** **<hostname>: Stale client handle**
 CLASS: SYS0
 PROGRAMS: RL
 SITUATION: Connection to client RLnet is not functional
 ADVICE: Reconnect is performed automatically
- 5394:** **Journal: <Journal message>**
 CLASS: SYS1
 PROGRAMS: RL
 SITUATION: Recovery from a transaction log encountered an error
 ADVICE: If the system did not come up, run RLrebuild and/or
 remove journal files. Check for missing data
- 5405:** **RL notification failed**
 CLASS: SYS2 (auto recoverable system error)
 PROGRAMS: RLnet
 SITUATION: RLnet failed call to contact RL server
 ADVICE: Make sure the RL server is up and the client node can
 contact it.
- 5408:** **Undefined ADN (%s) in site_exits.**
 CLASS: SYS1 (user recoverable system error)
 PROGRAMS: RL
 SITUATION: An ADN configured in the `site_exits` file is
 unknown to the REEL servers (Informational
 message)
 ADVICE: The `site_exits` file contains a reference to a device
 that has not been configured in REELlibrarian.
 Correct the entry in the `site_exits` file or define the
 device in `rlconfig.site_exits` is scanned at five-
 minute intervals; changes to `rlconfig` will be

detected during this scan without stopping the REELlibrarian servers.

5712: REEL detected operating system change
CLASS: SYS1 (user recoverable system error)
PROGRAMS: RL
SITUATION: Informational message (system update)
ADVICE: Contact your system administrator.

User Messages

All REELlibrarian user messages are listed below. For each message, the following information is listed:

- The text of the message as it appears on screen.
- The terrno value returned by the API call.
- The situation that led to the generation of the message.
- Advice for solving the problem, if applicable.

Message List

Aborted file at EOT prevents appending with this device

TERRNO: RL_EOTADA
SITUATION: To be determined.
ADVICE: To be determined.

Ambiguous file name

TERRNO: RL_FAMBIG
SITUATION: There are multiple files in the library with the same name.
ADVICE: Specify the volumeset of the file being requested.

Attempt to modify privileged volume field

TERRNO: RL_VPRIV
SITUATION: Certain fields of volume information are not user-modifiable.
ADVICE: Contact an operator to request changes.

Bad File Name

TERRNO: RL_BADFN
SITUATION: The length or format of the file name is improper.
ADVICE: To be determined.

Bad Volumeset descriptor

TERRNO: RL_BADDSD
SITUATION: An invalid vsd was passed to a vs(3) library routine.
ADVICE: The vsd must be the return code from vs_vaccess().

Bad arguments

TERRNO: RL_BADARG
SITUATION: Invalid or conflicting arguments.
ADVICE: Verify correctness of arguments to command.

Bad date specification

TERRNO: RL_BDATE
SITUATION: Invalid expiration date specification.
ADVICE: See rlfedit() or rlvedit() man page for syntax.

Bad file specification

TERRNO: RL_BADFID
SITUATION: The file selection mode is invalid.
ADVICE: Specify an appropriate mode.

Bad volume number

TERRNO: RL_BADVNO
SITUATION: NEVER HAPPENS
ADVICE: To be determined.

Can not remove a volume currently in a volumeset

TERRNO: RL_NOREM
SITUATION: Volume must be scratch to be removed.
ADVICE: Use `rlvscratch force=y` to scratch volumeset.

Cannot skip unmount

TERRNO: RL_USKP
SITUATION: An attempt was made to skip an unmount.
ADVICE: Respond to the appropriate MID.

Command permission denied

TERRNO: RL_NOTOP
SITUATION: User attempted to perform a privileged operation and has not been identified as an operator.
ADVICE: Register the user as an operator or tell him to cut it out.

Communication link already exists

TERRNO: RL_ELINK
SITUATION: Duplication of a notification endpoint was attempted.
ADVICE: Check the environment variable RL_KEY.

Could deadlock

TERRNO: RL_DEADLK
SITUATION: The user already has a device reserved.
ADVICE: Free devices before reserving more, reserve all needed devices at on time, or use different keys for multiple devices.

Could not read information source

TERRNO: RL_SREAD
SITUATION: rlvread failed opening named pipe.
ADVICE: Check permissions on fifo.

Data access error

TERRNO: RL_DBERR
SITUATION: To be determined.
ADVICE: To be determined.

Data format mismatch

TERRNO: RL_DFMT
SITUATION: The tape label does not match the specified format.
ADVICE: Dump and inspect the labels.

Device Reset failed

TERRNO: RL_NORESET
SITUATION: Error while rewinding volume.
ADVICE: Check for processing accessing the device. Run device diagnostics.

Device access error - [open|close|ioctl]

TERRNO: RL_DEVACC

SITUATION: Indicated system call returned an unexpected error.

ADVICE: Check that the device is operating and unused.

Device already has a mounted volume

TERRNO: RL_TNMNT

SITUATION: The specified device is in use.

ADVICE: Try another device. Or wait.

Device already in use

TERRNO: RL_INUSE

SITUATION: The device specified for volumeset access is already supporting a volumeset.

ADVICE: Check device status with `rls`.

Device can only append at EOT

TERRNO: RL_EOTAPP

SITUATION: Attempt to overwrite data on a tape. Tape-cap indicates that it is not allowed.

ADVICE: If the device should be able to overwrite, call customer support.

Device cannot append

TERRNO: RL_NOAPP

SITUATION: Attempt to append to a device whose tape-cap entry indicates that it cannot.

ADVICE: If the device should be able to append, call customer support.

Device does not have a volume mounted

TERRNO: RL_NOTAPE
SITUATION: To be determined.
ADVICE: To be determined.

Device down

TERRNO: RL_DEVDWN
SITUATION: The specified adn is not usable.
ADVICE: Check device status with `rls`. Check device info with `rlr dinfo`. Check host status.

Device type and volume type incompatible

TERRNO: RL_NOFMT
SITUATION: The selected device does not support the media type of the volume.
ADVICE: Check the volume attributes. Modify the device configuration or use a different device.

Duplicate Volume ID

TERRNO: RL_VIDDUP
SITUATION: The specified volume id conflicts with a pre-existing volume.
ADVICE: Select another id or use `rlremoved` to delete the other.

Duplicate rack number

TERRNO: RL_RCKDUP
SITUATION: The specified rack number conflicts with a pre-existing volume.
ADVICE: Select another number or use `rlremoved` to delete the other.

Duplicate tapecap entry

TERRNO: RL_TCDUP

SITUATION: The name specified for a new tapecap entry already exists.

ADVICE: Select another name.

Empty volumeset must be accessed for writing

TERRNO: RL_WACC

SITUATION: Newly created volumesets cannot be read before being written.

ADVICE: Set the write flag on accessing the volumeset.

End of Volume

TERRNO: RL_EOT

SITUATION: Read processing reached End of Data.

ADVICE: Cannot read past End of Data. Check for empty tape files.

File list not complete

TERRNO: RL_FLCMP

SITUATION: End of volume set has not yet been reached on the currently accessed volumeset.

ADVICE: None

File not expired

TERRNO: RL_FNOEXP

SITUATION: Attempt to write to an unexpired file.

ADVICE: To be determined.

File permission denied

TERRNO: RL_FPERM
SITUATION: The user's effective id has not been granted access to the file.
ADVICE: Have the file's owner grant the permissions.

Format mismatch on specific scratch request

TERRNO: RL_MISFMT (CRAY Only)
SITUATION: Attempt to allocate specific scratch tape has conflicting information.
ADVICE: Use `rlr vinfo` to inspect volume attributes.

Format not available on machine

TERRNO: RL_BADFM
SITUATION: A machine specification conflicts with a media type specification.
ADVICE: Do not specify a machine.

Host not trusted

TERRNO: RL_NOTRUST
SITUATION: A query was received from an unregistered host.
ADVICE: Check the file
\$RLLIBDIR/REEL/Librarian/trusted_hosts for a list of registered hosts.

Incorrect password

TERRNO: RL_PASSWD
SITUATION: The password is incorrect.
ADVICE: Submit the correct password.

Insufficient devices

TERRNO: RL_INDEV

SITUATION: A reservation request is asking for more devices than are up.

ADVICE: Check devices with rls.

Insufficient usable devices

TERRNO: RL_INUDEV

SITUATION: A reservation request cannot be satisfied with devices that are available.

ADVICE: Check devices with rls. Verify that they are appropriately placed and support the media type.

Internal assertion failed (*number*)

TERRNO: RL_ASSERT

SITUATION: Internal system error.

ADVICE: Call customer support.

Label correspondence error

TERRNO: RL_LBLCOR

SITUATION: A physical label field disagrees with the database.

ADVICE: See the network log for a specific message.

Label processing error

TERRNO: RL_LBLPROC

SITUATION: Unexpected data found where labels were expected.

ADVICE: Dump and inspect the labels.

Labeled Tape layer not reset

TERRNO: RL_LTERR
SITUATION: Unrecoverable error while processing volume.
ADVICE: Check `rllog -m` for related messages. Run `rbctest -c`.

Length mismatch on specific scratch request

TERRNO: RL_MISLEN (CRAY Only)
SITUATION: Attempt to allocate specific scratch tape has conflicting information.
ADVICE: Use `rlr vinfo` to inspect volume attributes.

Library sequence error

TERRNO: RL_SEQ
SITUATION: An operation has been attempted out of order.
ADVICE: Check that modes agree with requests and that all setup for data transfer operations is performed.

License connection limited exceeded

TERRNO: RL_NLICE
SITUATION: Too many client systems have requested services.
ADVICE: Check license status with `RLlicense info`.

Location mismatch on specific scratch request

TERRNO: RL_MISLOC (CRAY Only)
SITUATION: Attempt to allocate specific scratch tape has conflicting information.
ADVICE: Use `rlr vinfo` to inspect volume attributes.

Machine hosting device is down

TERRNO: RL_NODED

SITUATION: A requested device is unavailable because its machine is out of service.

ADVICE: Use another device. Check status with `rls`.

Missing volume label

TERRNO: RL_VOLHUH

SITUATION: No label found where one is expected.

ADVICE: Dump and inspect the labels.

Network Communication Error

TERRNO: RL_COMERR

SITUATION: Remote machine access failed.

ADVICE: Check network host status. Check remote server status.

No Label information

TERRNO: RL_UNLBL

SITUATION: The volume labels are missing.

ADVICE: Dump and inspect the tape.

No compatible volumes available

TERRNO: RL_PNA

SITUATION: No scratch volume exists to be allocated from the pool.

ADVICE: Add more tapes to the pool with `rlpsubmit`. Use `rlpedit padd=` to produce automatic addition requests.

No device reserved

TERRNO: RL_NODEV
SITUATION: A device reservation request was not made or failed.
ADVICE: Check the device specification. Check device status with `rls`.

No device supports the indicated media type

TERRNO: RL_TYPUM
SITUATION: No available devices can accept the requested media.
ADVICE: Find such a device. Check host and device status with `rls`.

No match

TERRNO: RL_NOMTCH
SITUATION: Specified volume or volumeset name not found in volume database.
ADVICE: Verify name accuracy. Use `.volname` for unnamed volumesets. Refer to the section titled *Submitting a Volumeset*.

No such file on Volumeset

TERRNO: RL_NOFILE
SITUATION: The specified tape file is not on the currently accessed volumeset.
ADVICE: Use `rlr vsflist` to display all files on the volumeset.

Non-existent communication link

TERRNO: RL_ULINK
SITUATION: Removal of a nonexistent endpoint was attempted.
ADVICE: Check the environment variable `RL_KEY` and the `uid`.

Only owner can override expiration date

TERRNO: RL_EXOVER

SITUATION: Attempt to force the write mount of a volume which is part of a volumeset.

ADVICE: Have the owner scratch the volumeset.

Only privileged users may modify automatic volume addition

TERRNO: RL_AAPRIV

SITUATION: Only privileged users may modify automatic volume addition

ADVICE: Have a privileged user modify the field using rlpedit.

Pool already exists

TERRNO: RL_PDUP

SITUATION: To be determined.

ADVICE: To be determined.

Pool not empty

TERRNO: RL_PNOE

SITUATION: A request was made to remove a pool that contains volumes.

ADVICE: Identify the volumes with rlr pvolumes. Remove with rlremove.

Pool permission denied

TERRNO: RL_PPERM

SITUATION: The effective user is not allowed to access the specified pool.

ADVICE: Verify the pool specification in the request. Have the owner of the pool provide access with rlpedit.

Port mapper not responding

TERRNO: RL_RPMAP
SITUATION: System network outage. RPC name server not responding.
ADVICE: Check portmapper with `rpcinfo -p`. Contact system administrator.

Pseudo Name already in use

TERRNO: RL_PSDIU
SITUATION: The name selected to assign to the device during reservation already exists.
ADVICE: Select another name.

REELlibrarian going down

TERRNO: RL_DOWN
SITUATION: The RL server is being brought down or has aborted abnormally.
ADVICE: Check `rllog -m` for messages. Correct any problems. Use `reel start` to bring servers back up.

REELlibrarian server not responding

TERRNO: RL_IRPC
SITUATION: NEVER HAPPENS
ADVICE: To be determined.

RPC Service appears to be hung

TERRNO: RL_NOANSW
SITUATION: System network outage.
ADVICE: Contact system administrator.

RPC connection error

TERRNO: RL_RPCERR

SITUATION: System network outage.

ADVICE: Contact system administrator.

Rack number required

TERRNO: RL_NORACK

SITUATION: No rack location was provided for volume acceptance.

ADVICE: A rack location must be provided at submission or acceptance.

Ran off end of volume list

TERRNO: RL_NOVOL

SITUATION: To be determined.

ADVICE: To be determined.

Ran out of Volumeset descriptors

TERRNO: RL_DSDOF

SITUATION: Too many accessed volumesets.

ADVICE: Use `vs_vrelease` to free descriptors.

Raw Tape layer not reset

TERRNO: RL_RTERR

SITUATION: Unrecoverable error while processing volume.

ADVICE: Check `rlllog -m` for related messages. Call customer support.

Receipt number and volume ID don't match

TERRNO: RL_VNOT

SITUATION: The user either presented the wrong receipt number or requested the wrong volume.

ADVICE: Verify the receipt number accuracy. Check the volume with `rlr vinfo`.

Record id not unique

TERRNO: RL_NOTUNQ

SITUATION: The specified key (fingerprint or vsn) matches multiple volume records.

ADVICE: To be determined.

Record length error

TERRNO: RL_RECL

SITUATION: Incorrectly sized record submitted for writing.

ADVICE: Check record format specification.

Record processing inconsistency

TERRNO: RL_BADREC

SITUATION: Record format specification invalid.

ADVICE: To be determined.

Reel system error

TERRNO: RL_SYSERR

SITUATION: The Librarian software is internally inconsistent.

ADVICE: Contact technical support.

Remote signal

TERRNO: RL_RSIG
SITUATION: Operation interrupted by a signal.
ADVICE: Repeat operation if desired.

Request canceled

TERRNO: RL_CANCL
SITUATION: A waiting request was cancelled by user request or status change.
ADVICE: Check status with `rls` and resubmit the request if desired.

Request cannot be satisfied immediately

TERRNO: RL_QNOQ
SITUATION: A device reservation request was made without the Reservation Queue Flag and could not be satisfied immediately.
ADVICE: Use the Reservation Queue Flag or repeat the reservation at intervals.

Request queued

TERRNO: RL_QUEUED
SITUATION: Indicates that a reservation or access request is pending, waiting for resources to become available.
ADVICE: None

Requested service unavailable

TERRNO: RL_NOQ
SITUATION: Indicates that the RL server is going down.
ADVICE: See the REEL administrator.

Rewind error

TERRNO: RL_REWERR
SITUATION: Attempt to rewind a volume failed.
ADVICE: Check that no other process has the device open.

Rotation Permission denied

TERRNO: RL_RPERM
SITUATION: An unprivileged user cannot create pools prefixed by another's name.
ADVICE: Do not use ` / ' in rotation names.

Rotation Specification already exists

TERRNO: RL_ROTTEXT
SITUATION: Attempt to create existing rotation.
ADVICE: Check accuracy of name. Use existing rotation.
Delete old rotation using `rlrdelete`.

See server error list

TERRNO: RL_ELIST
SITUATION: To be determined.
ADVICE: To be determined.

Service unavailable

TERRNO: RL_UNSERV
SITUATION: RPC call failed.
ADVICE: Check that servers are running. Verify network connectivity and addresses.

Template translation error

TERRNO: RL_TMPTRN

SITUATION: Syntax error in record specification template.

ADVICE: Inspect specification.

Un-accessed volume referenced

TERRNO: RL_NOACC

SITUATION: To be determined.

ADVICE: To be determined.

Unable to save access state

TERRNO: RL_NOSAVE

SITUATION: To be determined.

ADVICE: To be determined.

Undefined media type

TERRNO: RL_BADFMT

SITUATION: The specified media type is nonexistent or is not supported by the selected adn.

ADVICE: Check defined media types with `rlr types`. Check types supported by (reserved) device with `rlinq psd`.

Unknown ADN

TERRNO: RL_BADADN

SITUATION: The specified (or default) adn did not match any that are configured. Can occur when adn file is missing or corrupt.

ADVICE: Check adn list with `rls`.

Unknown Location

TERRNO: RL_BADLOC

SITUATION: An invalid name was submitted as a site location.

ADVICE: Use `rlr sites` to produce a list of locations.

Unknown Rotation Specification

TERRNO: RL_RHUH

SITUATION: Specified rotation name not found in database.

ADVICE: Check accuracy of name. Add rotation using `rlrcreate`.

Unknown Rotation Specification

TERRNO: RL_UNKROT

SITUATION: Specified rotation name not found in database.

ADVICE: Check accuracy of name. Add rotation using `rlrcreate`.

Unknown Volume Identifier

TERRNO: RL_BADVID

SITUATION: Volume id not found in database.

ADVICE: Check accuracy of volume id.

Unknown Volume Serial Number

TERRNO: RL_UNKVSN

SITUATION: Specified volume serial number not found in volume database.

ADVICE: Check accuracy of vsn.

Unknown device reference

TERRNO: RL_UNKNOW

SITUATION: A device request had an invalid psd.

ADVICE: Specify the correct psd.

Unknown file

TERRNO: RL_UNKFID

SITUATION: Specified file name not found in file database.

ADVICE: Check accuracy of file name.

Unknown finger print

TERRNO: RL_BFING

SITUATION: Specified fingerprint not found in volume database.

ADVICE: Fingerprint may be corrupted. Check with
rltapevol.

Unknown location/rack

TERRNO: RL_UNKRCK

SITUATION: Specified location and/or rack number not found in
database.

ADVICE: Use `rlr sites` to display locations. Use `rlr
vinfo` to dump volume info.

Unknown machine name

TERRNO: RL_BADMCH

SITUATION: A reservation request contains a bad host name.

ADVICE: Use `ping` to check host names.

Unknown pool

TERRNO: RL_PHUH

SITUATION: Pool specification or volume pool name not found.

ADVICE: Recheck pool specification. Recreate pool if missing.

Unknown request id

TERRNO: RL_UMID

SITUATION: A reference to a mount request has an invalid ID.

ADVICE: Check the pending requests with `rls` or `rlmon`.

Unknown reservation request

TERRNO: RL_URSV

SITUATION: A query or cancellation request could not find an associated reservation.

ADVICE: Check the environment variable `RL_KEY` and the `uid`.

Unknown tapecap type

TERRNO: RL_TCUNK

SITUATION: The specified tapecap name was not found in the tapecap database.

ADVICE: Use `rltapecap` to list known names.

Unknown volume or volumeset name

TERRNO: RL_THUH

SITUATION: Specified name not found in volume database.

ADVICE: Verify name accuracy. Use `.volname` for unnamed volumesets. Refer to the section titled *Submitting a Volumeset*.

Unrecognized group

TERRNO: RL_GAUTH

SITUATION: A group name is not known to the system hosting the RL server.

ADVICE: Add the name to the group database which the host references.

Unrecognized user name

TERRNO: RL_UAUTH

SITUATION: A user name is not known to the system hosting the RL server.

ADVICE: Add the name to the user database which the host references.

Vid not scratch on specific scratch request

TERRNO: RL_NOTSCR (CRAY Only)

SITUATION: Attempt to allocate specific scratch tape has conflicting information.

ADVICE: Use `rlr vinfo` to inspect volume attributes.

Volume ID not premounted

TERRNO: RL_NOTPRE

SITUATION: The device for which an unpremount was requested is not shown as having a premounted volume.

ADVICE: Check device status with `rls`.

Volume Identifier required

TERRNO: RL_REQVID

SITUATION: The system cannot identify a mounted tape by its fingerprint and the mount request response did not include a volume id.

Provide the volume id using `rldone` or thorough the request monitor.

Volume Serial Number incorrect

TERRNO: RL_BADVSN

SITUATION: The tape label does not match the requested VSN.

ADVICE: Use `rltapevol` to check the volume's identity.

Volume Serial Number mismatch

TERRNO: RL_VOLMIS

SITUATION: The tape label does not match the requested VSN.

ADVICE: Use `rltapevol` to check the volume's identity.

Volume Submission/Retrieval error

TERRNO: RL_TMODE

SITUATION: An operation was attempted on a volume(set) which its submission status disallows.

ADVICE: Check submission status with `rlr vinfo`. Modify using `rlaccept`, `rlvretrieve`, and `rlremoved`.

Volume already stacked

TERRNO: RL_ASTACK

SITUATION: Request to stack a volume on a device specifies a volume already assigned to a device.

ADVICE: Do not duplicate volumes in the request. Use `rlr vlist` to display volume status.

Volume format and record format incompatible

TERRNO: RL_RVFMT

SITUATION: Record formatting only exists for ANSI, IBM and Unlabeled volumes.

ADVICE: Change the `rformat` argument to `u`.

Volume format only allows unformatted (u) record format (rformat)

TERRNO: RL_UREC

SITUATION: Record formatting only exists for ANSI, IBM and Unlabeled volumes.

ADVICE: Change the rformat argument to u.

Volume ioctl error

TERRNO: RL_TIOCTL

SITUATION: Device not accepting system call.

ADVICE: Check device. Run `rldtest`

Volume mounted read only

TERRNO: RL_READO

SITUATION: Attempt to write on a volume with no write access requested.

ADVICE: Specify the write flag when accessing the volumeset.

Volume mounted write only

TERRNO: RL_WRITEO

SITUATION: Read of file attempted on volumeset accessed for write only.

ADVICE: Check the write flag on the access request.

Volume not expired

TERRNO: RL_VNOEXP

SITUATION: Attempt to overwrite a volume which is part of a volumeset.

ADVICE: Check volume status with `rlr vinfo`. Use `rlvscratch` to disband volumeset.

Volume not in mountable location

- TERRNO:** RL_TLMNT
- SITUATION:** The volume or a volumeset member is registered as being at a site from which it cannot be accessed.
- ADVICE:** Use `rlr vinfo` to check the volume attributes. Use `rlmoved` to change the volume's current location. Or: use `rlconfig` to temporarily alter the site's mountability attribute.

Volume not mountable

- TERRNO:** RL_UNMNT
- SITUATION:** The volume is not at a mountable site or is not in an available state.
- ADVICE:** Check volume status with `rlr vinfo`.

Volume not scratch

- TERRNO:** RL_VSCR
- SITUATION:** Attempt to retrieve a volume which is part of a volumeset.
- ADVICE:** Check volume status with `rlr vinfo`. Use `rlvscratch` to disband volumeset.

Volume or Volumeset in use elsewhere

- TERRNO:** RL_TLOCK
- SITUATION:** Only one access may be current on a volumeset at a time.
- ADVICE:** Free the volumeset with `rlvrelease` or wait for it to be freed.

Volume permission denied

TERRNO: RL_TPERM

SITUATION: The effective user is not allowed to access the specified volume.

ADVICE: Check that the user is known to the RL master host. Check that the volume name is accurate. Check the permissions.

Volume read error

TERRNO: RL_TREAD

SITUATION: Read failed. Possible reasons include: media error, device failure, incorrect length specification.

ADVICE: Check device and media.

Volume write error

TERRNO: RL_TWRITE

SITUATION: Write failed. Possible reasons include: media error, device failure, incorrect length specification.

ADVICE: Check device and media.

Volumeset already exists

TERRNO: RL_TDUP

SITUATION: When submitting or allocating a volumeset, the name specified matches an already existing one.

ADVICE: Check volumeset names with `rlr vslist`.

Volumeset already exists

TERRNO: RL_VSEXIS

SITUATION: Volumeset names (in full nomenclature) must be unique.

ADVICE: Use a different name or a later generation for the volumeset name.

Volumeset contains non-mountable volumes

TERRNO: RL_DSUMNT
SITUATION: One or more of the volumeset's constituents is not available.
ADVICE: Use `rlr vsvlist` to obtain the set of volumes. Use `rlr vinfo` to check their status.

Volumeset name can't begin with '.'

TERRNO: RL_DOTVNM
SITUATION: The initial '.' is reserved for accessing unnamed volumesets.
ADVICE: Retry without the '.'.

Volumeset name syntax error

TERRNO: RL_BADVN
SITUATION: The length or format of the volumeset name is improper.
ADVICE: See `rlvcreate(1)` for format description.

Volumeset not expired

TERRNO: RL_DNOEXP
SITUATION: Attempt to scratch a volumeset which has not expired.
ADVICE: Use `rlvscratch force=y` to disband volumeset.

Waiting for a Volume mount

TERRNO: RL_MWAIT
SITUATION: A mount request could not immediately be satisfied and is pending.
ADVICE: None

Waiting for new pool volume

TERRNO: RL_POOLQ

SITUATION: A scratch tape request from the pool generated a volume add dialog which has not been completed.

ADVICE: None

tapecap file error

TERRNO: RL_TCERR

SITUATION: A corrupted record was encountered while reading the tapecap database.

ADVICE: Examine \$RLLIBDIR/REEL/Librarian/tapecap for corruption.

Index

Symbols

.DF files, 149–151

A

ACC, 97

accessing tape drives, 153

accessing volumes, 153

accounting, 49

accounting logs, 49, 71

ACT, 86

action requests, 88

administrator

 command summary, 77

 defined, 16

Adn, 352

ADN, see assigned device name

age, 98

alias list, 45

aliases, 357

 media type, 42

ANSI, 19, 143

 creating labels for, 144

 data files, 148

 file formats, 145

 text files, 147

ANSI2, 19

Application field, 121

assigned device name, 84

associated device name, 352

auto density detect, 36

auto density select, 36

autoloaders

 configuring, 53

 messages, 93

 working with, 93

automatic volume recognition, 49

B

backups

 Vault Management System, 27–29

binary directory, 26

C

catalog, 17, 22

 backup and recovery, 74

 contents of, 95

- defined, 111
- editing, 104, 130
- journal, 72
- using, 141

changing hostnames, 75

cleaning interval, 44

Cleaning Regime field, 133

cleaning tapes, 106

client-server, 3

cln tape status, 98

CLNTNAME, 362

commands

- administrator, 77
- operator, 107
- user, 156

Comment field, 120

configuration, 29, 362

- of autoloaders, 53
- of drives, 50
- of REELlibrarian, 15
- of trusted_hosts file, 52
- of vault management, 55

constants, 354

controlling servers, 80

controlling the network, 81

conventions

- UNIX-style manpages, xv

conversion

- record, 120, 146, 149

CPIO, 19, 143

creating a rotation, 135

creating a volumeset, 124

creating pools, 131

ctf, 98

D

daemon, 3

data files

- ANSI, 148
- IBM, 148

Data Format field, 118

Days before Recycle field, 133

DCHK, 90, 93

deadlock situations, 155

default constants, 354

default length, 44

default volumeset formats, 149–151

deleting a pool, 135

deleting a rotation, 138

deleting volumesets, 125

density check, 90

dev.default, 356

DEV-DOWN, 87

device

- access, 153
- throughput, 39

device definition files, 352

device information report, 186

device lock, 356

device resource manager, 23

devices

- defining, 34

DEV-RESET, 87

DEV-UP, 86
dinfo, 186
direct device access, 153
directional keys, 32
directory
 journal, 46
disabling tracing, 70
disposal limit, 44
disposing tapes, 106
drive locking, 40
drive locking permissions, 41
drives
 accessing, 153
 configuring, 50
 freeing reserved, 155
 idle, 24
 multiple, 154
 status of, 85
DRM, see device resource manager
duration, 136
dynamic tracing, 69

E

editing a pool, 134
editing a rotation, 138
editing a volumeset, 121
editing tape catalog entries, 130
editing tape catalogs, 104
enabling tracing, 69
End-of-Life Disposal field, 133
entering tapes into library, 99–103

eras, 98
Erase field, 120
error limit, 44
EXIT, 87
Expire field, 119
expired volumesets, 126

F

file attributes report, 172
file formats
 ANSI, 145
 IBM, 145
file labels, 144
file list, 171
file list report, 171
file name templates, 151
File Template field, 120
File Tracking field, 120
files
 log, 412
 reading from volumesets, 140
 storing on volumesets, 139
finfo, 172
fingerprint, 7, 18, 128
flist, 171
fmts, 357
FORCE, 88
Form Factor Specification, 369
format
 of volumesets, 149–151
 record, 120

format definitions, 357
format list, 36
Format List Window, 36
freeing reserved drives, 155
full-screen interface
 control keys, 115
 defined, 113
 invoking, 113
 navigating, 114
full-screen navigation, 29

G

GID, 25
Group List field, 134

H

hard error limit, 50
hardware write protection, 50
HELP, 87
hosts
 changing name of, 75

I

IBM, 19, 143
 creating labels for, 144
 data files, 148
 file formats, 145
 no-label tapes, 148
 text files, 148
IBM2, 19
IBMU, 148

identification of volumes, 49
identification problems, 89, 92
idle drives, 24
idle time, 47
init, 99
Initialize field, 122
installation, 4
ISODATES, 362

J

jdir, 358, 359
journal directory, 46
journals, 46, 72

K

key commands, 32, 69
keys, 359

L

labels, 19
 ANSI, 144
 file, 144
 IBM, 144
 volume, 143
length, 101
Length/ Capacity field, 119
LIB, 99
library, 17, 109
 catalog, 95
 entering tapes into, 99–103

- removing tapes from, 103
- library directory, 26
- librltape.a C library, 193
- licensing, 74
- loc, 360
- location, 101, 136
- Location field, 119
- locking, 40
- log file, 412
- logging
 - reelenv variables, 70, 365
- logs
 - tracing, 69
- lost, 98

M

- machine name, 36
- maint, 183
- mandatory volume ID, 48
- manpages
 - UNIX chapter conventions, xv
- master list, 32
- MDIA, 452
- media type, 84, 101
- media type aliases, 42
- Media Type field, 118
- media type report, 177
- Media Type Specification, 370
- messages, 366
 - autoloader, 93
 - system, 451–500

- MID, see mount ID
- Miscellaneous Screen, 45
- MNT-DONE, 87
- MNT-SKIP, 87, 88
- Mode field, 119
- monitor
 - request, 83
- mount ID, 86
- mount request system, 3, 22
- mount requests
 - skipping, 92
- move, 98
- movement
 - of media, 22
- moving tapes, 106
- MRS, see mount request system
- multiple tape drives, 154

N

- Need Tape Operator, 81
- NetClient, 3, 75
- NetMaster, 3, 74
- network, 25, 47
 - control, 81
 - status, 82
- network cycle interval, 47
- network servers
 - controlling, 80
- NEW, 88
- no rewind, 38
- NUTIL, 383

O

- operations, 25
- operator
 - command summary, 107
 - current, 79
 - paging, 33, 81
 - prompt interval, 47
 - responsibilities, 79
 - unmount requests, 50
- operator definition, 361
- operator prompt interval, 47
- operator reports, 182
- operators
 - defined, 16
 - defining, 31
- ops, 361
- Other Device List, 39
- other device list, 36

P

- paging, 81
- Password field, 119
- permissions
 - drive locking, 41
- pinfo, 168
- plist, 167
- pool attributes report, 168
- Pool Creation fields, 133–134
- Pool field, 119
- pool list report, 167
- Pool Name field, 133

- pool volumes report, 170
- pools, 4, 20
 - creating, 131
 - defined, 2, 111
 - deleting, 135
 - editing, 134
 - group list, 134
 - public
 - creating, 112
 - user list, 134
- posn_stack, 390
- premount delay, 47
- PRFM, 452
- priority, 48
- programming support, 193
- public pool
 - creating, 112
- pvolumes, 170

Q

- queue discipline, 48

R

- rack number, 18, 101
- RAW, 19, 143
- RB_LEAVE_LOG, 364
- RB_NOHOLES, 363
- rbadd, 445
- rbmoved, 435
- rbremoved, 444
- RBTDOWNTIME, 363

- RCOM, 383
- rec_get, 293
- rec_put, 294
- record conversion, 146, 149
- Record Conversion field, 120
- Record Format field, 120
- reel, 206, 391
- reel command, 82
- reel_env, 26, 392
- reel_error, 295
- reelb, 412
- REELbackup
 - documentation, xiii
 - relationship with REELlibrarian, 76
 - Vault Management System, 27–29
- reelenv, 25, 362
- REELlibrarian
 - configuring, 15
 - operations, 25
 - relationship with REELbackup, 76
 - system limits, 67
 - tracing facilities, 69
- rem, 98
- removing tapes from library, 103
- reports, 20, 171
 - device information, 186
 - file attributes, 172
 - media type, 177
 - operator, 182
 - pool attributes, 168
 - pool list, 167
 - pool volumes, 170
 - rotation information, 175
 - rotation list, 176
 - storage location, 176
 - tapecap, 188
 - user, 159
 - vault content, 190
 - volume attributes, 167
 - volume inventory, 185
 - volume list, 165
 - volume maintenance, 183
 - volumeset attributes, 162
 - volumeset file list, 179
 - volumeset list, 160
 - volumeset volume, 180
- request monitor, 83
- RET, 97
- Retain field, 120
- retrieval, 18
- retrieving scratch volumes, 129
- retrieving volumesets, 123
- rewind name, 38
- rinfo, 175
- RL, 384
- rl, 113, 207
- rl program, 80
- rl_cleaned, 304
- rl_dev, 296
- rl_done, 297
- rl_errlist, 298
- RL_FD_LIMIT, 363
- rl_fedit, 299
- rl_flist, 300

rl_freq, 301
rl_free, 302
rl_inq, 303
RL_KEY, 292
RL_LCHECKI, 363
RL_MACH, 362
RL_MANUAL_ROTATE, 363
rl_mon, 366
rl_moved, 304
RL_NOTSHORT, 364
RL_NOVERIFY, 364
RL_OCD, 366
rl_offack, 394
rl_op, 305
rl_removed, 304
rl_reserve, 307
rl_setkey, 310
rl_skip, 311
rl_structures, 312
rl_unq, 327
rl_valloc, 328
rl_vfreq, 329
rl_vid, 330
rl_vlist, 331
rl_vmount, 332
rl_vmove, 333
rl_vrec, 334
rl_vscratch, 335
rl_vunmount, 336
rl_wait, 306, 337
RL0001 - get_vinfo
seek error, 73
RL0002 - get_vinfo
read error, 73
RL0003 - get_vinfo
time error, 73
RL0004 - Record newer error, 73
RL0005 - Record check error, 74
rlaccept, 102, 395
rlaccept command, 49
RLbackup, 385
RLBINDIR, 26, 362
rlcertify, 403
rlcleaned, 106, 404
rlconfig, 30, 393
RLCOREDIR, 364
rldev, 405
RLDISABLEFORCEMNT, 363
RLDISABLEMNTSKIPS, 363
RLDISABLENEWVOL, 363
RLdmon, 383
rldone, 406
RLdserve, 383
rldtest, 50, 408
RLdump, 386
rledit, 104, 130, 208
rlerased, 409
RLexit, 384
rlfedit, 216
rlfound, 410
rlfree, 155, 221
rlid, 7, 102, 128, 411

rlid command, 49
RLidle, 383
RLinit, 383
rlinq, 222
rlist, 176
RLLIBDIR, 26, 362
RLlicense, 74, 387
rlllog, 412
RLLOG_LVL, 70
RLLOGDIR, 364
rlmon, 413
RLMOUNTMAX, 364
rlmoved, 106, 414
rlmsg, 415
RLnet, 388
RLnexit, 388
RLntest, 388
Rloffctl, 383
RLOG_, 365
RLOG_LVL, 365
rlop, 81, 91, 416
rlopmsg, 223
RLPBASE, 362
rlpcreate, 224
rlpdelete, 135, 226
rlpedit, 134, 227
rlpremount, 417
rlretrieve, 129, 229
rlpsubmit, 126, 230
rlr, 234, 418
 dinfo, 186
 finfo, 172
 flist, 171
 maint, 183
 pinfo, 168
 plist, 167
 pvolumes, 170
 rinfo, 175
 rlist, 176
 sites, 176
 tapecap, 188
 types, 177
 vcontent, 190
 vinfo, 162, 167
 vinventory, 185
 vlist, 165
 vsflist, 179
 vslist, 160
 vsvlist, 180
rlrcreate, 135, 237
rlrdelete, 138, 239
RLrebuild, 74, 389
rlremoved, 106, 422
rlreserve, 240
rlreturn, 103, 423
RLrewind, 383
rlrotate, 424
RLS, 383
rls, 82, 425
rls command, 82
rlskip, 427
rlsmove, 428
rlstampdb, 429

- rlstatus, 242
- rltapevol, 430
- RLtest, 384
- rltln, 99, 431
- rlunlock, 432
- rlunpremount, 433
- rlunq, 244
- RLV, 383
- rlvaccess, 245
- rlvcreate, 124, 247
- rlvdisplay, 254
- rlvedit, 121, 256
- RLverify, 383
- rlvmount, 153, 264
- rlvmove, 266
- rlvms_config, 62, 369, 434
- rlvms_confirm, 435
- rlvms_move, 438
- rlvms_report, 440
- rlvms_retrieve, 444
- rlvms_submit, 445
- rlvread, 267
- rlvrelease, 271
- rlvretrieve, 103, 123, 272
- rlvscratch, 125, 273
- rlvsubmit, 115, 274
- rlvtran, 141, 283
- rlvtruncate, 284
- rlvunmount, 153, 285
- rlvwrite, 286
- RMNT, 89, 92

- rotation information report, 175
- rotation list report, 176
- rotation name, 136
- rotations
 - creating, 135
 - deleting, 138
 - editing, 138

S

- scr, 98
- scratch volume, 20
 - retrieving a, 129
 - submitting, 126
- screen.info, 366
- security, 20, 119
- selective tracing, 69
- server control, 80
- server failure and recovery, 73
- servers, 24
 - starting, 83
 - status of, 82
 - stopping, 83
- site_exits, 374
 - file, 53
 - scripts, 54
- sites, 176
 - defining, 41
- Slot Incrementation, 371
- Slot Specification, 371
- SMNT, 89
- stacker indexing, 55

- stacker messages, 93
- stackers, 53, 93
- starting servers, 83
- status
 - of drives, 85
 - of network, 82
 - of servers, 82
- stc_prolog, 448
- stkmount, 54, 449
- stkunmount, 54, 450
- stopping servers, 83
- storage location report, 176
- Storage Sites Screen, 42
- SUB, 96
- submission, 18
 - one-step, 49
- submitting a volumeset, 116
- submitting scratch volumes, 126
- SYS0, 452
- SYS1, 452
- SYS2, 452
- system limits, 67
- system messages, 451–500

T

- tape drives
 - autoloaders, 53, 93
 - configuring, 50
 - defining, 34
 - idle, 47
 - premounts, 47
 - simultaneous use of, 154
 - status, 85
- tape life cycle, 96
- tape location, 360
- tape movement, 22
- tape sessions, 111, 139
- tape status
 - age, 98
 - ctf, 98
 - eras, 98
 - init, 99
 - LIB, 99
 - lost, 98
 - move, 98
 - rem, 98
 - scr, 98
- tapcap, 38, 50
 - EOT detection, 51
 - report, 188
 - testing an entry, 51
- tapes
 - cleaning, 106
 - CPIO, 143
 - disposing, 106
 - editing catalog, 104
 - entering into library, 99–103
 - identification problems, 89, 92
 - moving, 106
 - removing from library, 103
 - states, 98
 - TAR, 143
- TAR, 19, 141

- creating volumesets in, 142
- reading volumesets in, 142
- writing tapes in, 143

templates

- file name, 151

termcap, 67

terminal display, 67

terminal displays, 67

text files

- ANSI, 147
- IBM, 148

throughput, 38

- device, 39

TOPR, 452

tracing, 69

- disabling, 70
- enabling, 69
- specifying dynamic or selective, 69

transaction journal, 358

transient volumesets, 141

trusted hosts

- configuring, 52

trusted_hosts, 377

types, 177

U

UID, 25

UMNT, 90, 92

UNIX chapter conventions, xv

unknown volumes, 49

unmount requests

- suppressing, 50

user

- responsibilities, 109

user command summary, 156

user ID, 85

User List field, 134

user reports, 159

users

- defined, 16

using multiple tape drives, 154

V

VADD, 90, 93

vault content report, 190

vault management

- configuring, 55
- deactivating, 66
- defined, 55
- examples, 64
- implementing, 56
- setting up, 57

Vault Management System, 27–29

Vault Specification, 370

vaults

- defining, 41

vcontent, 190

vinfo, 162, 167

vinventory, 185

vlist, 165

VMS, see Vault Management System

volume attributes report, 167

- volume ID, 17, 48, 85, 101
 - mandatory, 48
- volume inventory report, 185
- volume list report, 165
- volume maintenance report, 183
- volume serial number, 101
- volumes
 - accessing, 153
 - editing catalog entry, 130
 - entry into library, 99
 - exit from library, 103
 - fingerprinting a, 128
 - hard errors, 50
 - identification, 49
 - labels for, 143
 - maintenance, 105
 - retrieving from the library, 129
 - retrieving scratch, 129
 - submitting to the library, 126
 - write protection, 50
- volumeset attributes report, 162
- Volumeset Edit Screen, 121
- volumeset file list report, 179
- volumeset list report, 160
- volumeset name, 110
- Volumeset Name field, 118
- Volumeset Submission fields, 118–121
- volumeset volume report, 180
- volumesets, 19
 - creating from pool volumes, 124
 - creating in TAR, 142
 - default formats, 149–151
 - defined, 2, 110, 115
 - deleting, 125
 - editing, 121
 - entry to library, 116
 - exit from library, 123
 - expired, 126
 - naming specification, 110
 - reading files from, 140
 - reading from TAR, 142
 - reports, 141
 - retrieving, 123
 - scratching, 125
 - storing files on, 139
 - transient, 141
- vrec_null, 338
- vs_cfile, 339
- vs_close, 340
- vs_open, 341
- vs_vaccess, 347
- vs_vrelease, 349
- vsflist, 179
- vslist, 160
- VSN(s) field, 118
- vsvlist, 180

W

- WMNT, 89, 91
- writing tapes in TAR, 143