OpenVault™ Operator's and Administrator's Guide

007–3211–008 Version 1.5

New Features in This Manual

OpenVault now runs on the SGI ProPack for Linux family of servers and superclusters. Functionality for SGI ProPack for Linux is described throughout this document.

On IRIX systems, the location of OpenVault files has been changed in accordance with the Linux Filesystem Hierarchy, where possible.

Record of Revision

Version	Description
001	December 1997 Original publication.
002	May 1998 Incorporates information in support of the OpenVault release 1.0.
003	September 1998 Incorporates information in support of the OpenVault release 1.2.
004	March 1999 Incorporates internal update information.
005	November 2000 Incorporates information in support of the OpenVault release 1.4 for systems running on the IRIX release 6.2 with License Tools 2.1.1 or higher, IRIX release 6.4 with License Tools 3.0 or higher, or IRIX release 6.5, which includes the appropriate License Tools.
006	January 2002 Incorporates information in support of the OpenVault release 1.4.1 which runs on IRIX version 6.5.14 with patch (see the Release Notes for the specific patch number), 6.5.15, and later versions.
007	March 2002 Incorporates information in support of the OpenVault release 1.4.2. See the Release Notes with the product for details about supported IRIX platforms, hardware types, and library types.
008	June 2003 Incorporates information in support of the OpenVault release 1.5 and SGI ProPack for Linux, version 2.2 and later.

Contents

Figures

Tables

Examples

Procedures

About This Guide

This guide documents OpenVault releases running on IRIX systems and SGI ProPack for Linux.

OpenVault is a package of mediation software that helps other applications manage removable media:

- This facility can support a wide range of removable media libraries, as well as a variety of drives interfaced to these libraries.
- The modular design of OpenVault eases the task of adding support for new robotic libraries and drives.
- User interfaces are provided by OpenVault client applications, which perform I/O to drives using standard system facilities after OpenVault has mounted and loaded media for the application.

This guide describes how to administer and operate OpenVault. It also provides an introduction to tertiary storage management.

Intended Audience

This guide is intended for administrators who set up the OpenVault system and monitor its operation, and for operators who perform prescribed storage management tasks. To use the information in this guide, you should have the following experience:

- Understanding UNIX system infrastructure including devices and networking
- Writing UNIX shell and Perl scripts
- Using common text editors (for example, `emacs`, `jot`, `nedit`, or `vi`)
- Using backup utilities such as `cpio`, `tar`, `xfsdump`, or IRIX NetWorker

What This Guide Contains

The following is an overview of the material in this guide:

- Chapter 1, page 1, describes OpenVault architecture and operation.

- Chapter 2, page 13, details server and client setup.
- Chapter 3, page 69, discusses the treatment of media cartridges.
- Chapter 4, page 85, presents procedures for system administrators.
- Chapter 5, page 103, tells how to perform day-to-day operator tasks.
- Chapter 6, page 109, talks about changing configurations.
- Chapter 7, page 113, is a conceptual introduction to this topic.
- Appendix A, page 125, discusses OpenVault error conditions.
- Appendix B, page 131, lists OpenVault administration commands.

Related Publications

The following documents contain additional information that may be helpful:

- The *OpenVault Application Programmer's Guide* describes the client side of OpenVault, showing how applications can make OpenVault requests in a prescribed format.
- The *OpenVault Infrastructure Programmer's Guide* describes the server side of OpenVault, showing how to write control programs for removable media libraries and drives.
- Release notes: On IRIX systems, you can view release notes by typing either `grelnotes` or `relnotes` at the command line. On SGI ProPack for Linux systems, see the documentation in `/usr/share/doc/openvault-`*version*.

Obtaining Publications

You can obtain SGI documentation in the following ways:

- See the SGI Technical Publications Library at `http://docs.sgi.com`. Various formats are available. This library contains the most recent and most comprehensive set of online books, release notes, man pages, and other information.

- If it is installed on your SGI system, you can use InfoSearch, an online tool that provides a more limited set of online books, release notes, and man pages. With an IRIX system, select **Help** from the Toolchest, and then select **InfoSearch**. Or you can type `infosearch` on a command line.
- You can also view man pages by typing `man` *title* on a command line.

Conventions

The following conventions are used throughout this document:

Convention	Meaning
`command`	This fixed-space font denotes literal items such as commands, files, routines, path names, signals, messages, and programming language structures.
`manpage`(*x*)	Man page section identifiers appear in parentheses after man page names.
variable	Italic typeface denotes variable entries and words or concepts being defined.
`user input`	This bold, fixed-space font denotes literal items that the user enters in interactive sessions. (Output is shown in nonbold, fixed-space font.)
[ ]	Brackets enclose optional portions of a command or directive line.
...	Ellipses indicate that a preceding element can be repeated.

Reader Comments

If you have comments about the technical accuracy, content, or organization of this publication, contact SGI. Be sure to include the title and document number of the publication with your comments. (Online, the document number is located in the front matter of the publication. In printed publications, the document number is located at the bottom of each page.)

You can contact SGI in any of the following ways:

- Send e-mail to the following address:

  techpubs@sgi.com

- Use the Feedback option on the Technical Publications Library Web page:

  `http://docs.sgi.com`

- Contact your customer service representative and ask that an incident be filed in the SGI incident tracking system.

- Send mail to the following address:

  Technical Publications
  SGI
  1600 Amphitheatre Parkway, M/S 535
  Mountain View, California 94043–1351

- Send a fax to the attention of “Technical Publications” at +1 650 932 0801.

SGI values your comments and will respond to them promptly.

Understanding OpenVault

OpenVault is a storage library management facility that improves how applications can manage, store, and retrieve removable media (called cartridges). As an overseer of storage applications, and the libraries and drives that manage storage, OpenVault is aware of resources and allocates them accordingly, reducing bottlenecks and device mismanagement.

This chapter introduces OpenVault, and is divided into the following major sections:

- Section 1.1, page 1, describes the OpenVault software package.
- Section 1.2, page 2, describes how OpenVault works with other software.
- Section 1.3, page 4, defines terms used in this document.
- Section 1.5, page 7, describes the OpenVault architecture.
- Section 1.6, page 9, describes OpenVault operations.

1.1 What OpenVault Does

OpenVault is a package of mediation software that helps other applications access and manage removable media. This facility can support a wide range of removable media libraries, as well as a variety of drives associated with these libraries. OpenVault includes a database to track cartridges and storage devices. The modular design of OpenVault eases the task of adding support for new robotic libraries and drives.

The advantage of using OpenVault is that it can manage multiple applications and track the devices and removable media that each application uses. Additionally, because it adds a standards-based layer of software between the storage application and a library device, new libraries can be introduced without having to obtain an updated version of an application, and conversely, new applications can be added without having to update library or drive interface software.

OpenVault also has a command-line interface for performing administrator and operator tasks. These tasks include the following:

- Setting up and configuring applications and storage devices to run with OpenVault
- Monitoring your storage management operations

- Organizing your storage libraries for optimal operation

Chapter 4, page 85, and Chapter 5, page 103, describe how to perform administrator and operator tasks. Appendix A, page 125, provides an overview of OpenVault administrative commands.

1.2 How OpenVault Fits In with Other Software

OpenVault, as middleware, operates between applications and devices, as shown in Figure 1-1, page 3. OpenVault uses a client/server model and is designed to work equally well in either a single-host computer system or in a networked environment supporting multiple computers.

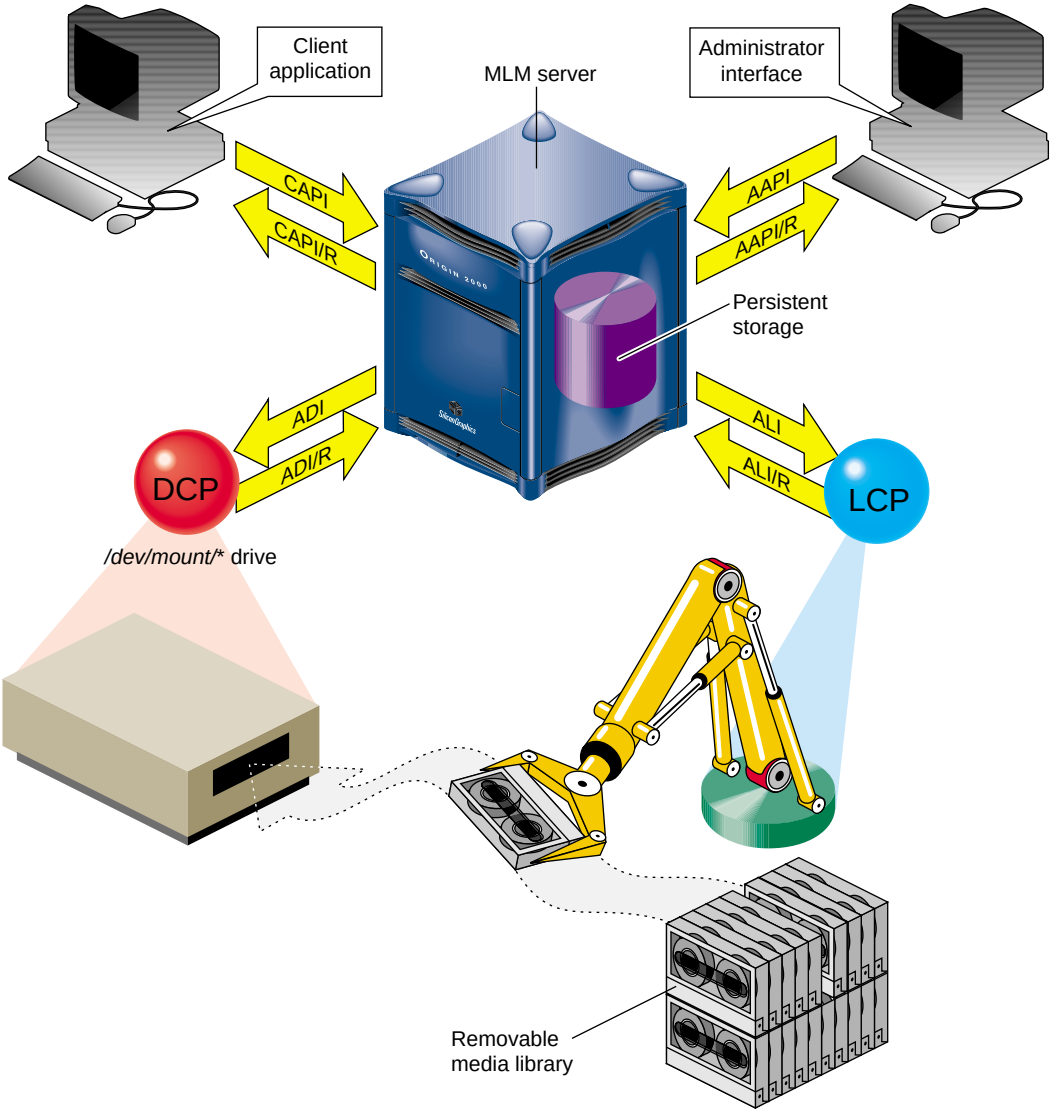

Figure 1-1 OpenVault Architecture

1.2.1 OpenVault as Middleware

Software that mediates between operating systems and application programs is called *middleware*. Middleware creates a common language so that users can access data in a variety of formats, or devices from different vendors.

As middleware, OpenVault receives high-level requests from client applications, and translates these requests into a set of low-level robotic or device-controller commands to accomplish storage-related tasks. As overseer of storage management, OpenVault schedules competing storage requests from different applications for the available devices, establishes and allocates cartridge groups by application, and provides mapping from logical cartridge names (as seen by the application) to physical cartridge numbers (as used by robots).

OpenVault software, as middleware, provides two standard interfaces:

- The first for storage-based applications that use cartridges
- The second for devices (libraries and drives) that store and retrieve information on cartridges

1.2.2 Client-Server Model

OpenVault works as a client-server model. At the most basic level, both client and server can operate on the same host. More typical, though, is a distributed system where multiple clients located on a network send requests to a centrally-managed server, which mediates access to locally attached libraries and drives.

1.3 OpenVault Terms

This section describes some storage management terms. Terms not specific to OpenVault are followed by an asterisk (*).

AAPI	Administrative application programming interface. Used to make administrative requests of the system.
AAPI/R	AAPI response. The system response to administrative requests.
ADI	Abstract drive interface. The OpenVault server issues directives to the DCP in a language called ADI.
ADI/R	ADI response. The DCP replies to the OpenVault server in a language called ADI/R.

ALI	Abstract library interface. The OpenVault server issues directives to the LCP in a language called ALI.
ALI/R	ALI response. The LCP replies to the OpenVault server in a language called ALI/R.
Barcode *	A horizontal strip of vertical bars of varying widths, which represent symbols referred to as characters. Barcodes are used as a cartridge label and are read by devices with read capability.
Bay *	A storage area in a library where an array of cartridges reside.
Cartridge*	A unit of removable media. May be a tape cartridge, a tape reel, an optical disc, a digital linear tape, a removable magnetic disk, or a videotape.
Cartridge group	A set of cartridges organized by common characteristics. A cartridge group may be created to form a cartridge set for use by an application, or to organize cartridges by subject matter, users, and so forth.
Catalog *	The listing of OpenVault components. The catalog tracks the status of cartridges, as well as authorized applications, drives, and libraries.
CAPI	Client application programming interface. Similar to AAPI, but used by applications to request service.
CAPI/R	CAPI response. The system response to application requests.
DCP	Drive control program. Required for each drive managed by OpenVault.
DLT *	Digital linear tape.
drive *	A device used to access the contents of cartridges.
drive group	A set of drives organized for a specific reason. Organization may be by type of drive (or drive capabilities) or for specific use by applications, user groups, and so forth.
LCP	Library control program. Required for each removable media library controlled by OpenVault.
library *	The collection of cartridges accessible by a storage device. A library can be automated (robot-assisted) or manual (worked by human operators).
OpenVault server	The set of processes that constitute the central mediation component that accepts client connections and fulfills access requests by forwarding them to appropriate library and drive control programs.
PCL	Physical cartridge label; usually a barcode, but may be human-readable.

shared secret *	A password that is common to two parties that share a secured transaction. For example, a client and server both share the password (secret) that is established at the setup of a client-server relationship.
slot *	A position in the library that can hold a cartridge. The slot may be free (unoccupied) or occupied.
slotmap *	A mapping of the free and occupied slots within a library.

1.4 OpenVault Definitions

This section describes some OpenVault definitions:

client host	A host computer running one or more OpenVault controlled drives or libraries, or application(s) that request services from an OpenVault server. A client host can also be the OpenVault server host.
DCP	The OpenVault client process(es) that directly control a drive under OpenVault management. The DCP interprets and services ADI requests on behalf of the OpenVault core.
LCP	The OpenVault client process(es) that directly control a library under OpenVault management. The LCP interprets and services ALI requests on behalf of the OpenVault core.
OpenVault application	The set of processes that communicate with the OpenVault server using either CAPI or AAPI. An application might be doing backup or archive, or it might be simply monitoring OpenVault activities.
OpenVault catalog	The central storage repository for an OpenVault system. This repository contains current knowledge about all libraries, drives, cartridges, and applications under OpenVault management.
OpenVault client	The set of processes that communicate with the OpenVault core, either to request services from the core (as would OpenVault applications), or to perform tasks on behalf of the core (an LCP or DCP).
OpenVault core	The set of entities that manage drives, libraries, and cartridges as well as service requests from OpenVault applications. The core does not include those processes that actually manipulate the devices on behalf of the OpenVault core (the LCPs and DCPs). The term “core” may also be used in various situations in place of the term OpenVault core.

OpenVault system	The set of hosts (server and clients) comprise the OpenVault system.
server host	The host computer running the set of entities that manage OpenVault controlled drives, libraries, and media. These entities are responsible for managing and allocating resources.

1.5 OpenVault Architecture

OpenVault is organized as a set of cooperating processes. The OpenVault server is a multithreaded process that accepts client connections and fulfills access requests by forwarding them to appropriate library and drive control programs. The OpenVault server maintains a catalog containing information about cartridges in the system, and descriptions of authorized applications, libraries, and drives. Figure 1-1, page 3, shows the arrangement of OpenVault components.

1.5.1 AAPI Programming Interface

AAPI (administrative API) is the language that administrative applications use to communicate with the OpenVault server. Commands and responses are text strings.

The command-response format is semi-asynchronous. After submitting each command, the application waits for the server to acknowledge receiving the command, but need not wait for results before sending the next command. AAPI communications libraries can also work synchronously if this makes implementation more convenient.

1.5.2 CAPI Programming Interface

CAPI (client application programming interface) is the language that client applications use to communicate with the OpenVault server. CAPI commands and responses are text strings. As with AAPI, the command-response format is semi-asynchronous, and access to the server is session-oriented. CAPI is a subset of AAPI.

The *OpenVault Application Programmer's Guide* tells how to program AAPI and CAPI.

1.5.3 OpenVault Server

The OpenVault server accepts requests from applications, and forwards commands to an LCP and DCP, which translate them into low-level robotic and drive control operations to serve that request. OpenVault also schedules competing requests from different applications, creates and enforces cartridge groups for specific application, and maps logical volume names (used by applications) to physical cartridge labels (used by libraries).

The OpenVault server manages cartridges, directing LCP and DCP to mount and unmount a cartridge. Often, cartridges store data. After requesting that a cartridge be mounted, the client application may read and write the media using POSIX standard I/O interfaces. Cartridges can also store audio-video streams for broadcast. In either case, OpenVault is not directly involved in I/O operations.

Client applications, libraries, and drives may be added to a live OpenVault server. The system administrator installs new programs on the appropriate hosts, and issues administrative commands on a live system to inform the OpenVault server that these new programs exist.

1.5.4 ALI/LCP Interface

A library control program (LCP) is a part of OpenVault that deals with low-level details of a removable media library and its configuration and control procedures. There is at least one LCP associated with each OpenVault-managed library.

The OpenVault server issues directives to the LCP in a language called ALI. The LCP replies to the OpenVault server in a language called ALI response (ALI/R).

1.5.5 ADI/DCP Interface

A drive control program (DCP) manages the configuration of drives, and performs the drive control tasks associated with CAPI mount and unmount requests. There is at least one DCP associated with each OpenVault-managed drive.

The OpenVault server issues directives to the DCP in a language called ADI. The DCP replies to the OpenVault server in a language called ADI response (ADI/R).

The *OpenVault Infrastructure Programmer's Guide* describes how to program ALI/LCP and ADI/DCP.

1.6 How OpenVault Operates

This section describes how LCP and DCP modules move from boot to operational state.

1.6.1 LCP Booting

When the LCP boots, it reads its configuration file and opens a connection with the OpenVault server. The server decides if the LCP should currently be in control of the library. If so, the OpenVault server tells the LCP that it controls the library. Once control is established, the LCP checks with its library to obtain additional information, such as:

- Whether the library is actually of a type supported by this LCP
- Whether barcodes or PCLs are supported
- List of cartridge form factors (for example, DLT)
- Total number of slots in the library
- Total number of occupied slots
- Import/export port configuration
- Slotmap (barcode-to-slot-location mapping)
- Other information that is relevant to the LCP

The LCP retrieves any stored state or configuration information from the OpenVault catalog (such as the error message log level). The LCP sends the OpenVault server its current slotmap and drive inventory so that the catalog can be updated, if necessary. At this point, the library can accept mount and unmount commands from OpenVault.

1.6.2 DCP Booting

DCP booting is similar to LCP booting: the difference is that the LCP has an inventory list and the DCP has a capability list.

When the DCP boots, the DCP also reads its configuration file and opens a connection with the OpenVault server. The server adds the drive into its managed drive list and establishes whether the DCP has sole ownership of the drive or shares it with another DCP. If a drive is simultaneously connected to more than one host, the OpenVault server must decide which DCP (on which host) has control of the drive (the server

can transfer control to another connected host on demand). Once the control is established, the DCP checks with its drive to obtain additional information, such as:

- Whether the drive is actually of a type supported by this DCP
- Supported media formats (for example, EXABYTE-8mm-5GB)
- Whether the listed access modes are supported
- Whether a cartridge is loaded in the drive
- To verify or acquire any other information that is relevant to the DCP

The DCP retrieves any stored state or configuration information from the OpenVault catalog (such as the error message log level). The DCP passes its capability list to the OpenVault server. The DCP sends the server its capability list so that the OpenVault catalog can be updated, if necessary.

Note: OpenVault applications must run on the same host as the OpenVault DCP client attached to the drive(s) employed by that application. This may or may not be the same host as the OpenVault server and LCP client.

1.7 OpenVault Installation

On IRIX systems, use the `inst` command (or comparable command) to load OpenVault software. If you choose the default installation, all server software, administrator and user commands, DCPs and LCPs, and man pages will be installed. To customize your installation, see Chapter 2, page 13.

On SGI ProPack for Linux, use the `rpm` command to install OpenVault software. The `openvault-sw` package contains all server software, administrator and user commands, DCPs and LCPS, and man pages.

For more detailed information and recommended procedures for OpenVault installation, see Chapter 2, page 13.

1.8 OpenVault Removal

On IRIX systems, to remove OpenVault software or individual components, use the `versions` command; see the `versions`(1M) man page. The following command removes all OpenVault software, databases, configuration files, and log files::

```
# versions remove OpenVault
# rm -rf /var/opt/openvault
# rm -rf /usr/openvault
```

To remove OpenVault software from SGI ProPack for Linux systems, use the `rpm` command. On SGI ProPack for Linux systems, OpenVault databases, config files, and logs are created in `/var/opt/openvault` and OpenVault executables are installed in `/opt/openvault`.

1.9 Cartridge Life Cycle

OpenVault stores and manipulates information on cartridges throughout their life cycles, and includes tools for the administrator to manage and monitor this information. For more information on this topic, see Chapter 3, page 69.

The *life cycle* of a cartridge is the chain of states and events which affect a cartridge from the time that it first becomes part of a system until in ceases to be a part of that system. The major events in the life of a cartridge include the following:

- Physical and logical introduction of the cartridge into the system
- Assignment of ownership (who or what application gets to use the cartridge)
- Use of the cartridge by applications
- Recycling of a cartridge when one owner no longer needs it
- Disposal of the cartridge either by sending it to another system or removing it for disposal when the media reaches the end of its service life

Installing OpenVault

This chapter describes how to install and set up your OpenVault system. There are two types of OpenVault hosts: the server, with the media library manager (MLM), and remote hosts, with an LCP or DCP, but without the MLM.

Depending on which hosts have drives and libraries physically or logically connected, configurations fall into one of two categories:

- Local configuration: all OpenVault-managed drives, libraries, and applications reside on the OpenVault server host.
- Local-and-remote configuration: one or more libraries or drives (or both) reside on OpenVault hosts other than the OpenVault server host.

If you have already configured OpenVault and would like to change the configuration, see Chapter 4, page 85.

2.1 OpenVault Installation Requirements

See the Release Notes that come with your OpenVault IRIX product for a detailed list of supported drive types, libraries, and the different OS versions that are supported.

On SGI ProPack for Linux systems, see the documentation in `/usr/share/doc/openvault-`*version*.

This release of OpenVault supports barcoded tape media only. You must have at least one tape for each drive type; the norm is to load a tape in each available library slot.

Though not required, if you configure OpenVault using a graphics console, resizing and backward scrolling (in `xwsh` for example) can make setup easier.

2.1.1 Licensing

You must have a license installed to run the OpenVault server software. Licensing tools must be installed beforehand on the OpenVault server system. IRIX 6.5 has the required License Tools.

To obtain a license, visit the Web site `http://www.sgi.com/support/licensing/index.html`, or send e-mail to

`license@sgi.com`, with a blank message, to obtain the license template. You may also use the FAX number, +1-650-920-0537, or this postal address:

SGI
1600 Amphitheatre Parkway
Mountain View, CA 94043
Attn: Software Licensing, MS 134

The OpenVault license you received includes instructions on how to install it.

On SGI ProPack for Linux systems, licensing tools are included with the base software.

2.1.2 Upgrading an IRIX OpenVault Installation

On IRIX systems, when upgrading from OpenVault 1.4.x or earlier versions to OpenVault 1.5 or later versions, the administrator will notice that the location of OpenVault executables, configuration files, databases, and log files has changed. In OpenVault 1.4.x and earlier releases, all OpenVault files were stored in the `/usr/OpenVault` directory. In OpenVault 1.5 and later releases, OpenVault files are stored in the `/usr/openvault` directory and in the `/var/opt/openvault` directory. OpenVault logs, configuration files, and databases are stored in the `/var/opt/openvault` directory.

When installing the software to upgrade OpenVault 1.4.x or earlier to OpenVault 1.5 or later, existing OpenVault log, configuration, and database files are copied from `/usr/OpenVault` to their new location. The upgrade script may also modify some of the new files, to conform to what OpenVault 1.5 expects. The old files are left in `/usr/OpenVault`. After doing such an upgrade, you should verify that the installation completed successfully, that all necessary files were copied, and that OpenVault functions correctly. At that point, you may want to make a backup copy of `/usr/OpenVault`, and then remove `/usr/OpenVault`.

2.1.3 OpenVault Software Components

If you have not already installed OpenVault, you can install it using the `inst` command or the graphical Software Manager on IRIX systems or `rpm` on SGI ProPack for Linux systems.

For more information about IRIX install programs, see the appropriate man pages.

The default OpenVault product images include the following subsystems on IRIX systems:

```
OpenVault.sw.core              OpenVault core servers
OpenVault.sw.admin             OpenVault administrative tools
OpenVault.sw.config            OpenVault configuration support scripts
OpenVault.sw.startstop         OpenVault scripts for starting and stopping daemons
OpenVault.sw.user              OpenVault end-user tools
OpenVault.upgrade              OpenVault scripts for upgrades to existing
                                    OpenVault installations
OpenVault.man.manpage          OpenVault manual pages
OpenVault.man.relnotes         OpenVault release notes
OpenVault.dcp.AIT1             OpenVault DCP for AIT-1 drive
OpenVault.dcp.AIT2             OpenVault DCP for AIT-2 drive
OpenVault.dcp.AIT3             OpenVault DCP for AIT-3 drive
OpenVault.dcp.DLT2000          OpenVault DCP for DLT2000 drive
OpenVault.dcp.DLT4000          OpenVault DCP for DLT4000 drive
OpenVault.dcp.DLT7000          OpenVault DCP for DLT7000 drive
OpenVault.dcp.DLT8000          OpenVault DCP for DLT8000 drive
OpenVault.dcp.IBM3590          OpenVault DCP for IBM 3590 Magstar drive
OpenVault.dcp.IBM3590E          OpenVault DCP for IBM 3590E Magstar drive
OpenVault.dcp.SDLT320          OpenVault DCP for SDLT320 drives
OpenVault.dcp.STK9840          OpenVault DCP for STK 9840 and T9840B drives
OpenVault.dcp.STK9940          OpenVault DCP for STK T9940A drive
OpenVault.dcp.STK9940B         OpenVault DCP for STK T9940B drive
OpenVault.dcp.SuperDLT1        OpenVault DCP for SuperDLT1 (SDLT220) drive
OpenVault.dcp.Ultrium1         OpenVault DCP for Seagate/IBM/HP LTO
                                      Ultrium Generation 1 drive
OpenVault.dcp.STKredwood       OpenVault DCP for STK SD-3 Redwood drive
OpenVault.dcp.STKtimberline    OpenVault DCP for STK 9490 Timberline drive
OpenVault.dcp.pseudo           OpenVault DCP for the "pseudo" drive
OpenVault.lcp.ADICDAS          OpenVault LCP for ADIC DAS interface libraries
OpenVault.lcp.ADICSCSI         OpenVault LCP for ADIC SCSI interface libraries
OpenVault.lcp.IBM3494          OpenVault LCP for IBM 3494 libraries
OpenVault.lcp.STK9700          OpenVault LCP for STK SCSI-attached 9700
                                      series libraries
OpenVault.lcp.STKLseries       OpenVault LCP for STK SCSI-attached L-series libraries
OpenVault.lcp.STKACSLS         OpenVault LCP for STK ACSLS libraries
OpenVault.lcp.pseudo           OpenVault LCP for the "pseudo" library
OpenVault.docs.adminguide      OpenVault Administrative manual
OpenVault.docs.designdoc       OpenVault design documentation
OpenVault.dev.include          OpenVault include headers for writing applications
OpenVault.dev.examples         OpenVault example application code
OpenVault.dev.libs             OpenVault LCP/DCP development libraries
```

On SGI ProPack for Linux systems, all of the server software, administrative and user tools, and supported LCPs and DCPs and man pages are included in the `openvault-sw` package.

2.2 Sample Configurations

Figure 2-1 shows an OpenVault local-only configuration.

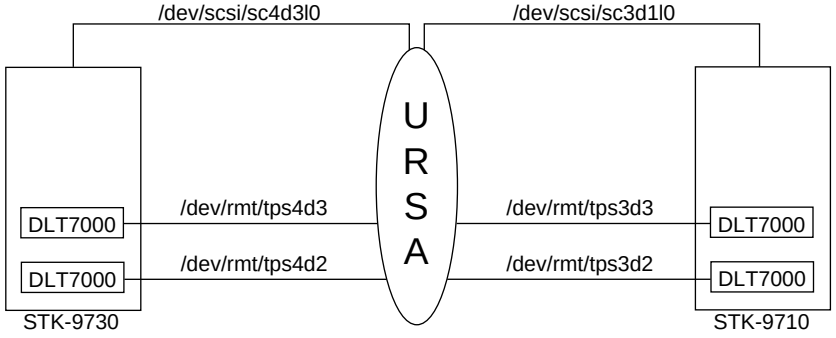

Figure 2-1 Local-Only OpenVault Configuration

This shows the OpenVault IRIX server named `ursa`, controlling two libraries: an STK-9730, connected at device address `/dev/scsi/sc2d3l0`, and an STK-9710, connected at device address `/dev/scsi/sc3d1l0`.

The STK-9730 at `/dev/scsi/sc2d3l0` contains two drives:

- DLT-7000 at `/dev/rmt/tps2d1`, physically located as the bottom drive in the library.
- DLT-7000 at `/dev/rmt/tps2d2`, physically located as the top drive in the library.

The STK-9710 at `/dev/scsi/sc3d1l0` contains two drives:

- DLT-7000 at `/dev/rmt/tps3d2`, physically located as the bottom drive in the library.
- DLT-7000 at `/dev/rmt/tps3d3`, physically located as the top drive in the library.

Figure 2-2 shows an OpenVault local-and-remote configuration.

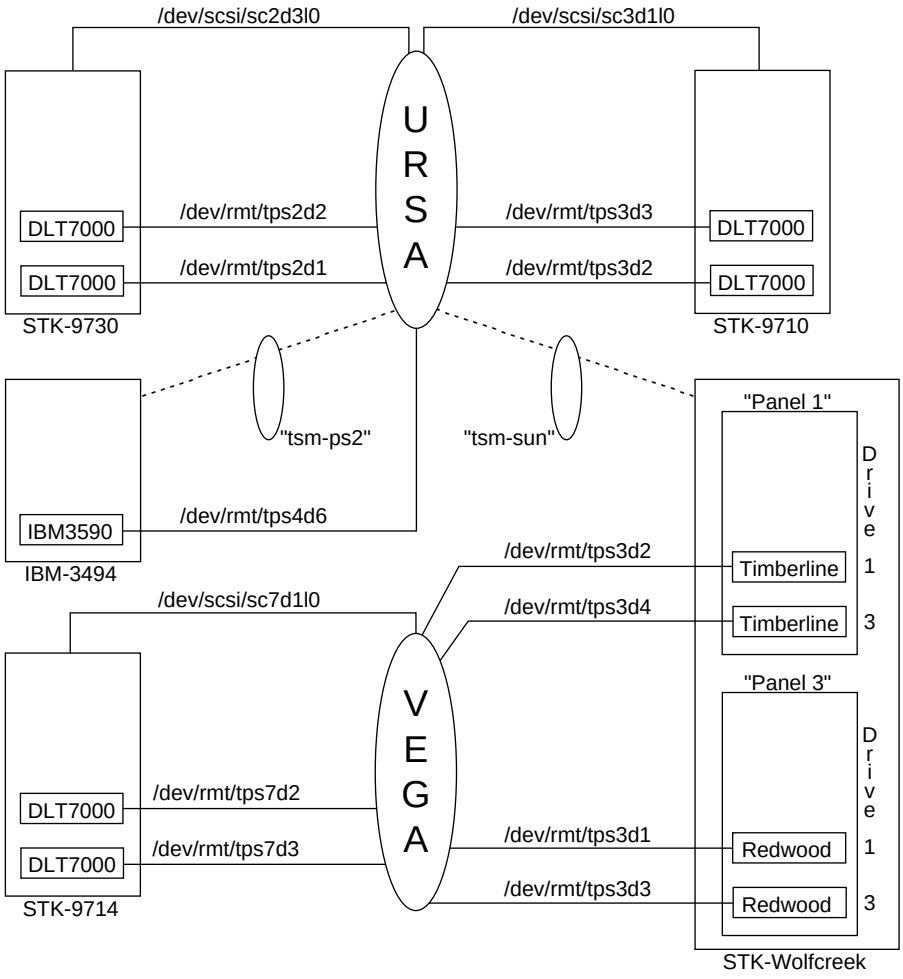

Figure 2-2 Local-and-Remote OpenVault Configuration

In Figure 2-2, `ursa` is the designated IRIX OpenVault server host, with four libraries:

1. STK-9730 connected at device address `/dev/scsi/sc2d3l0`
2. STK-9710 connected at device address `/dev/scsi/sc3d1l0`

3. IBM 3494 connected via an IBM library server running on the host, `tsm-ps2`
4. StorageTek WolfCreek silo connected via the ACSLS server host, tsm-sun

The STK-9730 on `ursa` has the following drives, both of which are connected to the host:

- DLT-7000 at `/dev/rmt/tps2d1`, physically located as the bottom drive in the library.
- DLT-7000 connected at address `/dev/rmt/tps2d2`, physically located as the top drive.

The STK-9710 on `ursa` has the following drives, both of which are connected to the host:

- DLT-7000 at `/dev/rmt/tps3d2`, physically located as the bottom drive in the library.
- DLT-7000 at `/dev/rmt/tps3d3`, physically located as the top drive.

The IBM 3494 on host `ursa` has the following drive, connected to the host `ursa`:

- IBM-3590 at `/dev/rmt/tps4d6`, identified by the IBM Library Server as drive 0

The StorageTek ACSLS-controlled WolfCreek silo has the following drives, all of which are connected to `vega`, not `ursa`:

- STK-RedWood at `/dev/rmt/tps3d1`, located physically in LSM 2, Panel 3, as Drive 1
- STK-RedWood at `/dev/rmt/tps3d3`, located physically in LSM 2, Panel 3, as Drive 3
- STK-TimberLine at `/dev/rmt/tps3d2`, located physically in LSM 2, Panel 1, as Drive 1
- STK-TimberLine at `/dev/rmt/tps3d4`, located physically in LSM 2, Panel 1, as Drive 3

Host `vega` is an OpenVault IRIX client host with one library: STK-9714 connected at device address `/dev/scsi/sc7d1l0`. The STK-9714 has the following drives, both of which are connected to `vega`:

- DLT-7000 at `/dev/rmt/tps7d3`, located physically as the bottom drive in the library.
- DLT-7000 at `/dev/rmt/tps7d2`, located physically as the top drive.

Notice that `vega` has two STK-RedWood drives and two STK-TimberLine drives connected to it, but the library in which these drives reside is not connected to `vega`.

On SGI ProPack for Linux systems, tape device pathnames have the same format as on IRIX systems (for example, `/dev/rmt/tps11d3`), but the corresponding SCSI device pathnames use an entirely different format. On SGI ProPack for Linux, SCSI device pathnames reside in subdirectories of `/dev/xscsi`, and have basenames of `ds` (for example, `/dev/xscsi/pci00.01.0-1/target0/lun0/ds` and `/dev/xscsi/pci01.00.0/node500104f0004256bb/port1/lun0/ds`.)

2.3 Configuration Roadmap

The general strategy for configuring OpenVault is shown in Table 2-1, page 19. The OpenVault configuration tool (`ov_admin`) allows you to perform many of these steps. The details of each step are explained in the rest of this chapter. Some steps may not be relevant to your specific installation:

Table 2-1 OpenVault Configuration Roadmap

1.	Prepare OpenVault hosts and devices. Install OpenVault license. Collect information and complete worksheets.	This step is required for all OpenVault hosts.
2.	Configure OpenVault core on the OpenVault server host.	This required step must be performed on the OpenVault server host.
3.	Configure drives attached to the OpenVault server host.	This step is required for drives connected to the OpenVault server host, and must be performed on the OpenVault server host.
4.	Configure libraries attached to the OpenVault server host.	This step is required for libraries connected to the OpenVault server host, and must be performed on the OpenVault server host.

5.	Enable applications on OpenVault server.	This step is required for any applications.
6.	Enable drives and libraries attached to remote OpenVault client hosts.	If you have remote libraries and drives, perform this step on the server host.
7.	Enable applications to run on remote OpenVault client hosts.	If you have applications running on remote hosts that request services from OpenVault, perform this step on the server host.
8.	Configure drives attached to OpenVault client hosts.	If you have drives attached to remote OpenVault client host(s), perform this step on the remote host(s).
9.	Configure libraries attached to OpenVault client hosts.	If you have libraries attached to remote OpenVault client host(s), perform this step on the remote host(s).
10.	Import media into OpenVault.	After configuring all libraries, perform this step on the OpenVault server host.
11.	Custom installation.	If desired.

2.4 Preparing OpenVault Devices and Hosts

This section explains how to prepare OpenVault managed devices for configuration. All drives you configure must be housed in a library, because OpenVault cannot manage standalone drives (that is, drives not housed in a robotic library). Procedure 2-1 describes the steps as preparation before configuring your OpenVault devices and hosts.

Procedure 2-1 Pre-Configuration Preparations

In this procedure, step 4 is required only if you are configuring a IBM 3494 library, and step 5 only if you are configuring a DAS library.

1. Cable all SCSI drives to appropriate hosts. Always ensure proper SCSI termination.
2. Cable all SCSI libraries to appropriate hosts.
3. For each host with an OpenVault controlled drive attached to it, run the following commands as `root`:

```
# killall mediad
# chkconfig mediad off
```

Note: The `mediad` daemon interferes with OpenVault operation; so it is essential that you prevent `mediad` from accessing OpenVault drives. Refer to the `mediad`(1M) man page for information on how to prevent `mediad` from accessing drives.

---

4. Prepare the IBM 3494 libraries that you are planning to configure:

Note: The IBM-3494 LCP is not currently supported on SGI ProPack for Linux systems.

- Enable communication between the LCP host and the IBM 3494 server. The IBM 3494 library server runs on the PS/2 system located at the rear of the library. From the graphical administration interface for the library server available on the PS/2, inform the library server of the OpenVault host that will run the LCP for the IBM 3494. Refer to the IBM 3494 Library server documentation for details on how to do this.
- On the OpenVault host that will run the LCP for the IBM 3494, prepare the `/etc/ibmatl.conf` file. If this file does not exist, you must create it. Briefly, this file contains three fields:

  libraryname *PS/2 hostname* *LCPhostname*

  Here *libraryname* can be any name not already in this file. The *PS/2 hostname* is the hostname or IP address of the PS/2 system that is running the IBM 3494 library server. The *LCPhostname* is the hostname of the machine that will run the LCP for the IBM 3494. Refer to the IBM 3494 documentation for details about the `/etc/ibmatl.conf` file.

Caution: It is critical that you remember *libraryname*, which is also the OpenVault name for this library.

- As `root`, start `lmcpd` on the IBM 3494 LCP host.

  On IRIX systems:

```
# cd /usr/openvault/clients/lcp/IBM-3494
# ./lmcpd
```

5. Prepare the DAS PS/2 system for any ADIC DAS or EMASS Grau library you plan to configure.

2.4.1 Configuration Worksheets

Use the following worksheets as an aid to collecting information that is required for configuring an OpenVault system.

Note the vendor, product, and serial number for each drive as you fill out the following worksheets.

Fill out Figure 2-3 with OpenVault server client host information.

OpenVault Server Host	OpenVault Client Host(s), If Any

Figure 2-3 Host Worksheet

Fill out Table 2-2 through Table 2-4 for any ADIC DAS or EMASS GAU libraries that you are configuring.

Table 2-2 Library Worksheets (DAS): Worksheet 1

Item	**Value**
Library Device Path	
DAS Server	
DAS Client Name	
Number of Export Areas	
Number of Import Areas	
OpenVault Name for Library	

Table 2-3 Library Worksheets (DAS): Worksheet 2

Import Area Name	**Open Vault Name**
Export Area Name	**OpenVault Name**

Fill out the following worksheet for drives in the DAS library:

Table 2-4 DAS Worksheets

DAS Name of Drive	OpenVault Name

Fill out Figure 2-4 for drives in a SCSI library.

Drive Device Path	Hostname to Which Drive Is Connected	OpenVault Name for Drive	Library in Which Drive Is Housed	Drive's Physical Location (SCSI Address)

Figure 2-4 Drive Worksheet (SCSI)

Fill out Figure 2-5 for drives in an IBM 3494 library.

Drive Device Path	Hostname to Which Drive Is Connected	OpenVault Name for Drive	Library in Which Drive Is Housed	Drive Number

Figure 2-5 Drive Worksheet (IBM 3494)

Fill out Figure 2-6 for drives in an ACSLS library.

Drive Device Path	Hostname to Which Drive Is Connected	OpenVault Name for Drive	Library in Which Drive Is Housed	LSM ID	Panel ID	Drive Number

Figure 2-6 Drive Worksheet (ACSLS)

Fill out Figure 2-7 and Figure 2-8 for each SCSI library.

Library Device Path	Hostname to Which Library Is Attached	OpenVault Name for Library

Figure 2-7 Library Worksheet (SCSI)

Description of Drive's Physical Location within Library	OpenVault Name for This Drive

Figure 2-8 Drive Worksheet (SCSI)

Fill out Figure 2-9 and Figure 2-10 for each IBM 3494 library.

Hostname or IP Address of PS/2 Controlling the 3494	OpenVault Name for Library

Figure 2-9 Library Worksheet (IBM 3494)

Drive Number as Described by `mtlib` Command	OpenVault Name for This Drive

Figure 2-10 Library Drive Worksheet (IBM)

Fill out Figure 2-11 and Figure 2-12 for each ACSLS library.

ACSLS Server Hostname	ACS ID	ACSAPI Packet Version	Hostname on Which LCP Will Run	OpenVault Name for Library

Figure 2-11 Library Worksheet (ACSLS)

Drive's Physical Location within Library			
LSM ID	Panel ID	Drive Number	OpenVault Name for This Drive

Figure 2-12 Library Worksheet (ACSLS)

2.4.2 Completing the Worksheets

Follow the order outlined to record information on the worksheets:

1. Acquire the names of the OpenVault server and client hosts. See Figure 2-3, page 22.
2. Generate unique names of the libraries. See Figure 2-7, page 26 through Figure 2-11, page 28.
3. Collect information about the attached drives. See Figure 2-4, page 24 through Figure 2-6, page 25.
4. Collect information for SCSI–attached libraries. See Figure 2-7, page 26 and Figure 2-8, page 26.

5. Collect information for IBM 3494 libraries. See Figure 2-9, page 27 and Figure 2-10, page 27.
6. Collect information for ACSLS libraries. See Figure 2-11, page 28 and Figure 2-12, page 28.
7. Collect information for ADIC DAS or EMASS Grau libraries. See Table 2-2, page 23, Table 2-3, page 23, and Table 2-4, page 24.

2.4.2.1 Acquiring Hostnames for OpenVault Hosts

You need to identify all hosts that form your OpenVault system. Identify only one host as the OpenVault server host.

In addition, you need to identify OpenVault client hosts, other than the OpenVault server host. These include any host on which you plan to configure a DCP, LCP, application, or OpenVault administrative commands. Wherever hostnames are required, use the exact output of the `hostname` command.

2.4.2.2 Generating Unique Names for Libraries

Generate unique names for all OpenVault managed libraries. You may choose any naming scheme that suits your site, provided the names remain unique. If you prepared an IBM 3494 as described earlier, you must generate a library name, which must become the OpenVault name.

2.4.2.3 Collecting Information for Attached Drives

Determine which drives are attached to each host. On SGI IRIX and Linux systems, you can use the `ov_scandev` command, which is part of OpenVault. Alternatively, on IRIX systems, you can use the `hinv` command. Example 2-1 shows the drives in the configuration in Figure 2-2, page 17.

Example 2-1 Identifying Drives

The following drives are on hosts `ursa` and `vega`:

```
ursa# hinv | grep Tape
Tape drive: unit 1 on SCSI controller 2: DLT
Tape drive: unit 2 on SCSI controller 2: DLT
Tape drive: unit 2 on SCSI controller 3: DLT
Tape drive: unit 3 on SCSI controller 3: DLT
Tape drive: unit 6 on SCSI controller 4: IBM Magstar 3590
```

```
vega# hinv | grep Tape
Tape drive: unit 1 on SCSI controller 3: STK SD3
Tape drive: unit 2 on SCSI controller 3: STK 9490
Tape drive: unit 3 on SCSI controller 3: STK SD3
Tape drive: unit 4 on SCSI controller 3: STK 9490
Tape drive: unit 2 on SCSI controller 7: DLT
Tape drive: unit 3 on SCSI controller 7: DLT
```

For each OpenVault-managed drive in an OpenVault library, follow these steps in Procedure 2-2:

Procedure 2-2 Collecting Drive Information

1. Determine to which OpenVault host the drive is physically cabled. Record this information in the drive worksheet. It may be necessary to follow SCSI cabling in order to determine this.

2. Determine the device path. You can use the `ov_scandev` command, which is part of OpenVault. See the man page for more information. On IRIX systems, you can also use the `hinv` command.

   The `ov_scandev` command provides you with drive serial numbers, which can be helpful in resolving path names. Alternatively, you could insert a tape into the drive, unload all other drives that are on the same SCSI bus as this drive, and issue an `mt status` command.

   For example, to determine this information for one of the two STK-SD3 (STK-RedWood) drives connected to `vega`, insert a cartridge into drive in question, unload the other drives on SCSI controller 3, and issue `mt -f /dev/rmt/tps3d1 status` and `mt -f /dev/rmt/tps3d3 status` commands. By process of elimination, you arrive at the required information.

3. Record the name of the library in which the device is physically housed.

4. Generate a unique name for this drive and record this name.

5. Record the physical location of drives in the library. For example, in SCSI libraries it is typical to find drives ordered in a linear fashion, either vertically or horizontally as viewed from the front of the library.

   For SCSI libraries, a scheme that describes the location of the drive in relation to the topology of the other drives in the library might suffice. For example, “second drive from bottom” or “third drive from left” are descriptive terms. Typically, StorageTek SCSI libraries order drives from the bottom, EXABYTE SCSI libraries

order drives from the top, while some SpectraLogic libraries order drives from left to right.

6. For drives in an ACSLS controlled robot, a physical description is not appropriate. You need to query ACSLS for more information. A summary of how you can do this is shown below. Refer to the ACSLS administration guide for details.

   - Log in to the ACSLS server with login **acsss**.
   - Run the command `cmd_proc`.
   - Issue the command `query drive all`.
   - Record the LSM ID, the Panel ID, and the drive # for the relevant drive(s).

     ```
     ACSSA  query drive all
     08-28-98 04:49:16                          Drive Status
     Identifier   State          Status      Volume     Type
     0, 2, 1, 1 online         available              9490
     0, 2, 1, 3 online         available              9490
     0, 2, 3, 1 online         available              SD3
     0, 2, 3, 3 online         available              SD3
     ```

     The comma-separated tokens are the ACS ID, the LSM ID, the panel ID, and the drive number, respectively.

7. For drives in an IBM 3494 library, a physical description is also not appropriate. You need to query the IBM 3494 server for more information. A summary of how to do this is shown below. Refer to the IBM 3494 documentation for details.

Note: The IBM-3494 LCP is not currently supported on SGI ProPack for Linux systems.

   - Prepare the OpenVault host to run the LCP for the IBM 3494 as described in Section 2.4.2.5, page 33. As `root` on IRIX systems, enter these commands:

     ```
     # cd /usr/openvault/clients/lcp/IBM-3494
     # ./mtlib -l libraryname -D
     ```

     Here *libraryname* is the name that you generated earlier (in Procedure 2-1, page 20) when you prepared the `/etc/ibmatl.conf` file.

With the configuration in Figure 2-2, page 17, the first output field of this `mtlib` command is a drive number; refer to the IBM 3494 documentation concerning how to map drive numbers to physical locations:

```
ursa 9# ./mtlib -l ibm3494 -D
0, 00141700 003590B1A00
```

- Record the drive number and physical location in the appropriate worksheets.

8. For drives in an ADIC DAS or EMASS Grau library, determine locality information similarly.

2.4.2.4 Collecting Information for SCSI-Attached Libraries

Follow the steps in Procedure 2-3 to obtain information for SCSI-Attached libraries.

Procedure 2-3 Collecting SCSI-Attached Libary Information

1. Record the device control path. The ov_scandev command, which is part of OpenVault, can be used to find this information. See the man page for details. Alternatively, on IRIX systems, you can use the `hinv` command to determine connected SCSI libraries, and use the output of `hinv` with the `scsicontrol` command to determine the type of each attached SCSI library, as follows:

```
# hinv | grep Juke | \
awk '/Juke/ {printf "/dev/scsi/sc%dd%dl0\n", $7, $3}' | \
xargs /usr/sbin/scsicontrol -i
/dev/scsi/sc2d3l0:  Jukebox       STK     9730            1300
/dev/scsi/sc3d1l0:  Jukebox       STK     9710            1805
```

   For each jukebox that `hinv` displays, issue a `scsicontrol` inquire command in the following form, where X is the controller number from the `hinv` output and Y is the unit number from the `hinv` output:

```
# scsicontrol -i /dev/scsi/scXdYl0
```

2. Repeat the previous step on every host with an OpenVault-controlled SCSI library.

3. Gather information about drives contained in libraries. For each SCSI library, enter these commands, where *LCPtype* is the LCP name, for example `STK9700`:

   On IRIX systems:

```
# cd /usr/openvault/clients/lcp/LCPtype
# ./LCP* getlibinfo/lcpcontrolpath| grep DRIVE
```

On SGI ProPack for Linux systems:

```
# cd /opt/openvault/clients/lcp/LCPtype
# ./LCP* getlibinfo/lcpcontrolpath | grep DRIVE
```

Here *lcpcontrolpath* is the `/dev/scsi` (or `/dev/xscsi` on SGI ProPack for Linux systems) control path as described in the first step. In the sample local-only configuration on host `ursa` (shown in Figure 2-2, page 17), enter the following commands for the STK-9730 library:

```
ursa# cd /usr/openvault/clients/lcp/STK9700
ursa# ./LCP* getlibinfo /dev/scsi/sc2d3l0 | grep DRIVE
DRIVE  1030            -    - first_drive_from_BOTTOM
DRIVE  1031            -    - second_drive_from_BOTTOM
```

Use the output information to complete worksheets for SCSI-attached libraries.

2.4.2.5 Collecting Information for IBM 3494 Libraries

Use Procedure 2-4 to collect information for IBM 3494 libraries.

Procedure 2-4 Collecting IBM 3494 Library Information

1. Record the hostname of the PS/2 system that is running the IBM 3494 server.
2. Record the drive information. The method for determining drive information is described in Section 2.4.2.3, page 29.

2.4.2.6 Collecting Information for ACSLS Libraries

Use Procedure 2-5 to collect information for ACSLS libraries.

Procedure 2-5 Collecting ACSLS Library Information

1. Record the hostname of the system running the ACSLS server.
2. Record the ACS ID of the ACSLS server that you plan to use. Refer to the documentation for the ACSLS server on how to obtain this ID.
3. Determine the packet version for ACSLS. The packet version is 1 less than the major version number of ACSLS running on the ACSLS server. For example, if you are running ACSLS version 5.1, use 4 as the packet version.
4. Record the drive information. The method for determining drive information is described in Section 2.4.2.3, page 29.

2.4.2.7 Collecting Information for ADIC DAS and EMASS Grau Libraries

For drives in an ADIC DAS or EMASS Grau library, collect drive information similarly.

2.4.3 Planning Cartridge and Drive Groups

Chapter 3, page 69, introduces the notion of cartridge and drive groups. The OpenVault `ov_admin` script initially creates one cartridge group named `carts`, and one drive group named `drives`.

All media that you import by means of the initial configuration procedure are by default introduced into the `carts` cartridge group. If you would like to create more cartridge groups, to allow importing different media into different cartridge groups, you may do so by following the instructions provided in Chapter 6, page 109. At initial configuration time, be sure to answer **no** when asked if you want to "Import Media." The Import Media option is available as part of reconfiguration procedures—after creating the desired cartridge groups, you may import media.

As with cartridge groups, at initial configuration time all drives are introduced into the `drives` drive group. You may choose to add new drive groups and move drives from one drive group to another at a later time by following instructions provided in Chapter 6, page 109.

Note: When adding new cartridge and drive groups, remember to enable appropriate applications to use these cartridge groups and drive groups.

2.4.4 Selecting a Password

If your site requires security, it is recommended that you create an OpenVault password. The `ov_admin` script asks you to enter an OpenVault password, for authentication of initially configured applications, drives, and libraries. If you want individual libraries, drives, or applications to use passwords other than the default password that you enter during initial setup, see the instructions in the Chapter 6, page 109.

2.4.5 Naming Libraries and Drives

The OpenVault setup procedure requires you to identify and provide OpenVault names for libraries and drives, each of which must have a unique OpenVault name.

The `ov_admin` script asks you to enter these names in one or more places. It is important that you enter all OpenVault drive and library names consistently and correctly. The configuration worksheets can help you do this.

If a library or drive is connected to the OpenVault *server* host, the OpenVault `ov_admin` script requests the OpenVault name for that physical library or drive. The script offers a generated name as default, which you may override by entering a name you select.

If a library or drive is connected to a remote OpenVault *client* host, the OpenVault `ov_admin` script requests an OpenVault name for that physical library / drive in these places:

- When you configure the OpenVault server host, you must enable each library / drive by providing its OpenVault name. At such time you have the choice of accepting a name generated by the `ov_admin` script, or entering a name you select.
- When you actually configure the library / drive on the OpenVault client host, the script asks you for the OpenVault library / drive name that you entered when you enabled the library / drive on the OpenVault server host.

Be sure to enter the same OpenVault library or drive name in both places!

Note: It is possible that drives are connected to remote OpenVault clients, but housed in a library connected to the OpenVault server host. In this case, configuration of the library requires entry of all drive names; the `ov_admin` script cannot offer defaults. If you wrote drive names for this library on your worksheet, then enter those existing names. If you did not write drive names on your worksheet, you must create them. Be sure to record drive names and enter them exactly later in the configuration process.

Regardless of where a drive is connected, at the time of configuring the library in which a drive is housed, you are asked to enter the OpenVault name of the drive. Be certain to enter the same OpenVault drive name in all three places!

2.5 Configuring the OpenVault Server

Procedure 2-6 describes the steps required to configure the OpenVault server, following the Section 2.3, page 19. Depending on the specific configuration at your site, some questions might not apply.

Procedure 2-6 Configuring the Server

You must configure the OpenVault server before any of its components.

1. Log in to the designated OpenVault server as `root`.
2. Ensure that the OpenVault license is installed in the proper file. On IRIX systems, it should be installed in the `/var/flexlm/license.dat` file. On SGI ProPack for Linux systems, it should be installed in the `/etc/flexlm/license.dat` file.
3. Execute the `ov_admin` script:

   # **`/usr/sbin/ov_admin`**

The `ov_admin` script is the main OpenVault configuration tool. It sets up OpenVault based on installed hardware and software, and your input to various questions. If you do not see a choice you want, double check your installation to make sure the items are installed. Where possible, the `ov_admin` script presents a default, indicated by "[*value*]:" at the end of line. You may accept this default by pressing the `Enter` key. Help is available at many prompts by entering a question mark (**?**).

2.5.1 Setting OpenVault Server Configuration Parameters

The `ov_admin` script goes through the following steps in setting the OpenVault server port:

1. The script starts by asking for the name of the OpenVault server:

   ```
   What is the name of the OpenVault Server? [ursa]
   ```

   Your answer informs `ov_admin` whether it is executing on the OpenVault server or not. Press `Enter` if you are configuring the OpenVault server host.

2. If the OpenVault server is not yet configured, the script asks:

   ```
   The OpenVault server is not yet configured;
   would you like to do so now?[Yes]
   ```

3. The script continues:

   ```
   What port number is OpenVault using? [44444]
   ```

   The default port number for OpenVault communications is 44444. Press `Enter` if this is acceptable.

If another application uses port number 44444, or if you prefer to select a different port, enter that port number. Make sure that no other application uses this port on the OpenVault server host.

4. The script continues:

   ```
   What default security key would you like to use? [none]
   ```

   The OpenVault server, LCPs, DCPs, and applications are by default configured without a security key, implying no security. If you enter a security key, the OpenVault server uses it to authenticate new connections from a client (an LCP, DCP, or application). If you have already selected a security key (password), enter it now.

   You may choose to configure OpenVault initially without security, which simplifies setup. Later, you can establish OpenVault security passwords by following steps in Chapter 6, page 109.

   Note: If you have remote OpenVault client hosts, you are prompted for the security key value at the time of configuring remote OpenVault client hosts. It is imperative that you enter the same value that you chose while configuring the OpenVault server. Failure to do so prevents successful configuration of the OpenVault client host.

   If the configuration was successful, the `ov_admin` script displays:

   ```
   The OpenVault server was successfully started.
   ```

5. At this point, the following items have been configured:

   - The OpenVault core including its catalog and an `ovroot` process is ready to service connections from OpenVault clients.
   - A default cartridge group has been created named `carts`.
   - A default drive group has been created named `drives`.
   - The user mounting application `ov_umsh` has been enabled on the OpenVault server (this host); see the `ov_umsh`(1M) man page for information.
   - The `ov_umsh` application is eligible to use cartridges in the `carts` cartridge group, and drives in the `drives` drive group.

6. The `ov_admin` script presents the **OpenVault Configuration Menu**:

```
OpenVault Configuration Menu

      Configuration on Machines Running LCPs and DCPs
         1 - Manage LCPs for locally attached Libraries
         2 - Manage DCPs for locally attached Drives

      Configuration on Admin-Enabled Machines
        11 - Manage Cartridge Groups
        12 - Manage Drive Groups
        13 - Import Media

      Configuration on the OpenVault Server Machine
        21 - Manage remote Libraries and LCPs
        22 - Manage remote Drives and DCPs
        23 - Manage Applications and Admin CLI Tools

         q - Exit.

    Which operation would you like to do:
```

It is recommended that you choose option 2 to proceed with configuration of locally attached drives at this time.

2.5.2 Configuring Locally Attached Drives on the OpenVault Server

The OpenVault `ov_admin` script scans hardware to determine which SCSI tape drives are attached to this host. The following menu is displayed when you run the `ov_admin` script:

```
OpenVault DCP Configuration Menu

      1 - Create a new DCP
      r - Return to Main Menu.
      q - Exit.

    Which operation would you like to do:
```

Choose option 1 to proceed.

1. The OpenVault `ov_admin` script scans the hardware to determine which SCSI tape drives are attached to this host.

If any drives are present, they are presented for configuration, as in the following example:

```
OpenVault -- DCP Creation Menu

          1 - /dev/rmt/tps4d2
          2 - /dev/rmt/tps4d3
          3 - /dev/rmt/tps3d2
          4 - /dev/rmt/tps3d3

        r - Return to Previous Menu.
        q - Exit.

    For which Drive would you like to add a DCP:
```

2. Check the drive worksheets that you created in Section 2.4, page 20. Choose a drive for which you would like to add a DCP. You can use the following criteria to determine which drive to choose:

   - It matches one of the "Device Drive Pathnames" on the worksheet and "Hostname Drive is Connected to" is equal to the current hostname.
   - You need to configure this drive at this time.

3. The `ov_admin` script then issues a series of prompts:

   ```
   Enter a name for the device, or <enter> to use the default name [tape1]
   ```

   The value entered here is the OpenVault name for the drive. See Section 2.4.5, page 34 for details. The script then prompts:

   ```
   What instance name would you like to give to this DCP? [tape1@ursa]
   ```

   The instance name can be any string. It is used to differentiate between two DCPs that are controlling the same drive. The script then prompts:

   ```
   What security key would you like to give to this DCP? [none]
   ```

   The script then prompts:

   ```
   What polling interval would you like this DCP to use? [30]
   ```

   The DCP needs to check the status of the drive periodically when the drive is not in use. The "polling interval" is the number of seconds between each of those status checks. The script then prompts:

```
What directory should hold handles for this DCP?
[/var/opt/openvault/clients/handles]
```

The DCP will be creating and deleting handles which are character special files that applications use to access the drive. The DCP needs a directory where it can store those handles. That directory can be shared with other DCPs but not with anything else.

4. The configuration parameters you have entered are displayed, and you are asked:

   ```
   Create the DCP now? [Yes]
   ```

   If you answer `no`, you can re-enter the configuration options. If you answer yes, and the DCP could be created, `ov_admin` displays the following message:

   ```
   DCP successfully created
   The following Drive Groups currently exist:
           drives
   What Drive Group do you want to use (or create)? [drives]
   ```

5. Press `Enter` to choose the default, or enter the name of an existing drive group, or enter the name of a drive group to be created. If you enter the name of a drive group to be created, you will be prompted for more information about it. If the DCP was successfully created, the script displays the results, as in the following example:

   ```
   Drive name                 :  tape1
   DCP name                   :  tape1@ursa
   DCP type                   :  DLT7000
   OpenVault Server host name  :  ursa
   OpenVault Server port number:  44444
   Security key               :  none
   DCP polling interval       :  30
   Handle creation directory   :  /var/opt/openvault/clients/handles
   Drive access path          :  /dev/rmt/tps2d1
   Direct SCSI access path    :  /dev/scsi/sc2d1l0
   ```

6. At this point, the following items have been configured:

   - The named drive entry has been added to the OpenVault catalog, and the drive has been added to the specified `drives` drive group.
   - An authentication entry has been added to the `/var/opt/openvault/server/config/core_keys` file.

- The drive's configuration file has been created (/var/opt/openvault/clients/dcp/*dname*/*instname*/config, where *dname* is the name chosen for this drive and *instname* is the instance name chosen).

2.5.3 Configuring Locally Attached Libraries on the OpenVault Server

To configure locally attached libraries, choose option 1 from the **OpenVault Configuration Menu**. This includes SCSI-attached libraries and network-attached libraries that you wish to configure. The ov_admin script prompts you:

```
OpenVault LCP Configuration Menu

      1 - Create a new SCSI LCP
      2 - Create a new network LCP

      r - Return to Main Menu.
      q - Exit.
```

2.5.3.1 Configuring SCSI-Attached Libraries

This section describes how to configure SCSI libraries.

1. Choose option 1 from the menu above. The ov_admin script then examines the hardware for attached libraries, and presents a list of SCSI-Attached libraries, as in this example:

   ```
   OpenVault -- SCSI LCP Creation Menu
         1 - /dev/scsi/sc4d3l0
         2 - /dev/scsi/sc3d1l0
         r - Return to Previous Menu.
         q - Exit.
     For which SCSI library would you like to add a LCP:
   ```

2. This prompt offers you the choice of configuring the library(s) at the address(es) shown. Choose the library you want to configure, based on the SCSI library worksheets that you created in Section 2.4, page 20. The ov_admin script then prompts:

   ```
   Enter a name for the device, or <enter> to use the default name [lib1]
   ```

The name entered here is the OpenVault name for the drive. See Section 2.4.5, page 34 for details.

3. The following prompt appears:

   ```
   What instance name would you like to give to this LCP? [lib1@ursa]
   ```

   The name can be any string. It is used to differentiate between two LCPs that are controlling the same library.

4. The following prompts appear:

   ```
   What security key would you like to give to this LCP? [none]

   What polling interval would you like this LCP to use? [30]
   ```

   The LCP needs to check the status of the library occasionally. The "polling interval" is the number of seconds between each of those status checks.

5. The `ov_admin` script then displays a message similar to this one:

   ```
   Configuring STK-9700 at /dev/scsi/sc4d3l0 to be "lib1"
   ```

6. This is followed by a prompt, similar to this one:

   ```
   What is the default shape of the slots in the library? [DLT]
   ```

   The `ov_slottype` command can be used to list the allowable slot types. The `ov_admin` script now prompts you to configure all of the drives contained in the library. The script queries the device to determine information about drives housed in this library. This device query might take some time, especially if the host has just been rebooted, if the device has just been power-cycled, or if the main door to the library has just been closed.

7. At this point, you see output similar to the following, repeated for each drive:

   For the drive at location *dlocation*,
   enter a drive name for the element address *daddr*

   dlocation is a text description of the drive's location, and *daddr* is the drive's "slot" address (the slot number that the SCSI robot uses to address the drive).

   You must enter the OpenVault name for the drive at the given physical location within the SCSI library.

- Refer to the SCSI library worksheet for this library to determine the OpenVault name of this drive.
- You need to match the description string *dlocation* with the column labeled "Description of Drive's Physical Location within Library." Although your description may not exactly match the description provided by the `ov_admin` script, choose the one that has the same meaning.

Tip: It is critical to the proper functioning of OpenVault that all drive names match up. If you have not already recorded a drive name on your worksheet, enter the correct name now. If you are running `ov_admin` in a scrolling window, scroll up to where you configured this drive, and use the name that was entered there.

- The *daddr* (the drive's slot address) is offered as a guide to the expert user. If you are familiar with the SCSI library's addressing scheme, you may choose to use this as an unambiguous reference to the drive's physical location within the library.
- If you do not want OpenVault to manage this drive, press `Enter`.

  The `ov_admin` script bypasses configuring the drive into the library; so OpenVault never mounts into this drive.

Note: If this drive is connected to a remote OpenVault client system, enter the name from your worksheet to configure this drive. If you have not yet recorded a name for this drive, select a unique drive name to enter at the prompt, and record this name on the worksheet. You will be required to enter this name when you later configure drives on the remote OpenVault client system to which the drive is attached.

---

8. After you enter the names or bypass, output similar to the following is displayed:

```
LCP Configuration Parameters:

 Library name                :  lib1
 LCP name                    :  lib1@ursa
 LCP type                    :  STK-9700
 OpenVault Server host name  :  ursa
 OpenVault Server port number:  44444
 Security key                :  none
```

```
LCP polling interval        :  30
Number of drives            :  2
Direct SCSI access path     :  /dev/scsi/sc2d3l0
Default cartridge shape     :  DLT

Drives in the Library

Drive Name      Drive Address
----------      -------------
drive1          500
drive2          501

Create the LCP now? [Yes]
```

The final line prompts you for confirmation:

- If you are satisfied with the SCSI library configuration summary shown above, enter **Y**.
- If you want to change one or more parameters, enter **N** and you can change the values.

9. Upon confirmation, the `ov_admin` script proceeds to configure the SCSI library:

   ```
   LCP successfully created
   ```

10. At this point, the following items have been configured:
    - The named library entry has been added to the OpenVault catalog.
    - An authentication entry has been added to the `/var/opt/openvault/server/config/core_keys` file.
    - The library's configuration file is created (`/var/opt/openvault/clients/lcp/`*lname*`/`*linst*`/config`, where *lname* is the name chosen for this library and *linst* is the instance name for the library.).

2.5.3.2 Configuring an IBM 3494 Library

Choose the **Create a New Network LCP** option from the **OpenVault LCP Configuration Menu** if you want to configure an IBM 3494 library. If you have software for the IBM 3494 LCP installed, then the `ov_admin` script displays the following menu:

```
OpenVault -- Network LCP Creation Menu

        1 - IBM-3494
        2 - STK-ACSLS
        3 - ADIC-DAS
        r - Return to Previous Menu.
        q - Exit.
    Which type of network LCP would you like to add:
```

Choose option 1 to configure an IBM 3494 LCP.

1. The following prompt is displayed:

   ```
   Enter a name for the device, or <enter> to use the default name [lib1]
   ```

   The ov_admin script proposes a default name for the library, lib*N* where *N* is 1 in this example.

2. Consult the IBM 3494 library worksheets you created in Section 2.4, page 20. If you have multiple IBM 3494 libraries to be configured, select one of them.

   If you have not yet chosen a name, or can accept the default name, press Enter. (When the library is being configured on an OpenVault client, the ov_admin script does not propose a default library name, but instead asks you to enter the name.)

   If you have chosen another name for this library, enter it at the prompt.

3. The ov_admin script issues the following prompts:

   ```
   What instance name would you like to give to this LCP? [ibm3494@ursa]
   What security key would you like to give to this LCP? [none]
   What polling interval would you like this LCP to use? [30]
   ```

4. The ov_admin script configures the LCP, and you see output similar to the following:

   ```
   Configuring IBM-3494 to be "ibm3494"
   ```

5. You are then prompted for the hostname or TCP/IP address for the PS/2 that is running the IBM Library Server for this library:

   ```
   What is the hostname or TCP/IP address of the PS/2 inside the library?
   ```

6. The `ov_admin` script queries the device to get information about drives housed in this library. This device query might take a while, especially if the host has just been rebooted, if the device has just been power-cycled, or if the main door to the library has just been closed. Soon you see output like this:

   ```
   Acquiring drive information from library (may take a while) ...
   ```

7. After querying information about contained drives, the `ov_admin` script asks you to enter the OpenVault name for each drive in this library, one after the other.

```
For the drive at location ''-'' Enter a drive name for the element address daddr:  3590
```

   - Here *daddr* is the drive's address as reported by the IBM 3494 Library server, usually a number starting from 0. Enter the OpenVault name for the drive at the given address. Refer to the IBM 3494 library worksheet to determine the OpenVault name for this drive. You must match drive address *daddr* with the column "Drive Number as Described by `mtlib` Command." See Section 2.4.2.3, page 29, for details on how to run the `mtlib` command.

     Tip: It is critical to the proper functioning of OpenVault that all drive names match up. If you have not already recorded a drive name on your worksheet, enter the correct name now. If you are running `setup` in a scrolling window, scroll up to where you configured this drive, and use the name that was entered there.

   - If you do not want OpenVault to manage this drive, press `Enter`. The script bypasses configuring the drive in the library; so OpenVault does not mount cartridges in this drive.

     Note: If this drive is connected to a remote OpenVault client system, enter the name from your worksheet to configure this drive. If you have not yet recorded a name for this drive, select a unique drive name to enter at the prompt, and record this name on the worksheet. You will be required to enter this name when you later configure drives on the remote OpenVault client system to which the drive is attached.

     ---

8. The script prompts you for OpenVault names for each drive housed in the library. After you enter names of all drives (or bypass) you see output similar to the following:

```
LCP Configuration Parameters:

Library name                 :  ibm3494
LCP name                     :  ibm3494@ursa
LCP type                     :  IBM-3494
OpenVault Server host name   :  ursa
OpenVault Server port number :  44444
Security key                 :  none
LCP polling interval         :  30
Number of drives             :  2
TCP/IP Address of the Library:  tsm-ps2

Drives in the Library

Drive Name      Drive Address
----------      -------------
3590            0

Create the LCP now? [Yes]
```

2.5.3.3 Configuring a StorageTek ACSLS library

To configure a StorageTek ACSLS library, choose the **Create a new network LCP** option from the **OpenVault LCP Configuration Menu**.

If you have LCP software for the StorageTek ACSLS installed on an OpenVault host, the `ov_admin` script offers the option of configuring this library. The following is an example of the **Network LCP Creation Menu**:

```
OpenVault -- Network LCP Creation Menu

        1 - IBM-3494
        2 - STK-ACSLS
        3 - ADIC-DAS

        r - Return to Previous Menu.
        q - Exit.

    Which type of network LCP would you like to add:
```

1. Choose option 2 to configure an STK-ACSLS library. You are then prompted:

```
Enter a name for the device, or <enter> to use the default name [lib1]
```

The name entered here is the OpenVault name for the library. See Section 2.4.5, page 34 for details.

2. The next prompt displays:

   ```
   What instance name would you like to give to this LCP? [lib1@ursa]
   ```

   The name can be any string. It is used to differentiate between two LCPs that are controlling the same library.

3. The following prompts appear:

   ```
   What security key would you like to give to this LCP? [none]

   What polling interval would you like this LCP to use? [30]
   ```

   The LCP needs to check the status of the library occasionally. The "polling interval" is the number of seconds between each of those status checks.

4. The ov_admin script then displays:

   ```
   Configuring STK-ACSLS to be "lib1"
   ```

5. You are then prompted for the host name of the ACSLS server for this library:

   ```
   What is the host name of the ACSLS server? []
   ```

6. After entering this host name, enter the version number of the ACSLS interface. Identify the version of ACSLS installed on the ACSLS server, subtract 1 from this number, and enter that value. See Section 2.4.2.6, page 33, for details.

   ```
   What is the version number of the ACSLS interface? [4]
   ```

7. Next enter the ID of the ACS that you are using for this LCP. Refer to the vendor's library documentation for information about obtaining the ACS ID.

   ```
   What is the ACS ID of ACS? [0]
   ```

8. The ov_admin script queries the device to determine information about drives housed in this library. This device query might take a while, especially if the host has just been rebooted, if the device has just been power-cycled, or if the main door to the library has just been closed.

9. Upon querying the information about contained drives, the `ov_admin` script asks you to enter the OpenVault name for each drive in this library, one after the other.

   ```
   For the drive at location ''ACSid 0, LSMid 2, Panelid 1,Driveid1''
   Enter a drive name for the element address "0,2,1,1": timberline1
   ```

   - The location string displayed is the ACS ID, LSM ID, Panel ID, and Drive number for the drive. See Section 2.4.2.3, page 29, for steps to obtain information for drives in an ACSLS library. Consult the ACSLS library worksheet for this library and select the drive entry with the corresponding ACS ID, LSM ID, Panel ID, and Drive number.
   - Enter the OpenVault name you selected.

     Tip: It is critical to the proper functioning of OpenVault that all drive names match up. If you have not already recorded a drive name on your worksheet, enter the correct name now. If you are running `ov_admin` in a scrolling window, scroll up to where you configured this drive, and use the name that was entered there.

   - If you do not want OpenVault to manage this drive, press `Enter`. The script bypasses configuring the drive in the library; so OpenVault does not mount cartridges in this drive.

     Note: If this drive is connected to a remote OpenVault client system, enter the name from your worksheet to configure this drive. If you have not yet recorded a name for this drive, select a unique drive name to enter at the prompt, and modify your worksheet. You will be required to enter this name when you later configure drives on the remote OpenVault client system to which the drive is attached.

   - The script prompts you for OpenVault names for each drive housed in the library.

10. After you enter names of all drives (or bypass) you see output similar to the following:

    ```
    LCP Configuration Parameters:

     Library name                 :  wolfcreek
     LCP name                     :  wolfcreek@ursa
    ```

```
LCP type                    :  STK-ACSLS
OpenVault Server host name  :  ursa
OpenVault Server port number:  44444
Security key                :  none
LCP polling interval        :  30
Number of drives            :  2
ACSLS Server Host name      :  tsm-sun
ACSLS Server Version        :  4
ACSLS Server ACS ID         :  0

Drives in the Library

Drive Name      Drive Address
----------      -------------
redwood1        0,2,3,1
redwood3        0,2,3,3
timberline1     0,2,1,1
timberline3     0,2,1,3

Create the LCP now? [Yes]
```

11. The final line prompts you for confirmation. If you are satisfied with the library and drive configuration shown, enter **Y**. If you want to change one or more parameters, enter **N** and you can change the configuration values.

Note: The `ov_admin` script does not allow you to change the name of the library at this time. However, you may do so at a later time by following steps in Chapter 7, page 113.

---

12. Upon confirmation, the `ov_admin` script proceeds to configure the library:

```
LCP successfully created
Library lib1 was successfully created.
```

13. At this point the following items have been configured:

    - The named library entry has been added to the OpenVault catalog.
    - An authentication entry has been added to the `/var/opt/openvault/server/config/core_keys` file.

- The library's configuration file is created (`/var/opt/openvault/clients/lcp/`*lname*`/`*linst*`/config`, where *lname* is the name chosen for this library and *linst* is the chosen instance name).

2.5.3.4 Configuring an ADIC DAS or EMASS Grau library

Configure an ADIC DAS or EMASS Grau library in a similar manner to the IBM 3494 library.

2.5.4 Enabling Remote LCPs, DCPs, and Administration

This optional configuration step enables the following features:

- Administration of the OpenVault server from remote OpenVault clients
- LCPs and DCPs that connect to the OpenVault server from OpenVault clients

This step enables remote connections. You must also run the `ov_admin` script on the remote OpenVault client(s) to configure libraries and drives on client host(s).

2.5.4.1 Enabling Remote Administration

It is possible to administer the OpenVault server from a client machine where you have installed the OpenVault administrative commands. To enable that, on the OpenVault host, choose option 23, **Manage Applications** from the **Admin CLI Tools from the OpenVault Configuration Menu**. The following menu is displayed:

```
Manage Applications and Admin CLI Tools Menu

   1 - Create a new Application
   2 - Delete an Application
   3 - Show all existing Applications

   4 - Activate another Application Instance for an existing Application
   5 - Deactivate an Application Instance
   6 - Show all activated Application Instances

   7 - Activate a Host to use the Admin CLI Tools
   8 - Deactivate a Host from using the Admin CLI Tools
   9 - Show all Hosts activated to use the Admin CLI Tools

   r - Return to Main Menu.
```

```
q - Exit.

 Which operation would you like to do:
```

Choose option 7 to activate a host to use the Administrative Command Line Tools. The ov_admin script then prompts:

```
For which Remote Host do you want to activate Admin CLI Tools? []
```

To identify remote client hostnames, consult the host worksheet that you generated in Section 2.4, page 20.

2.5.4.2 Enabling Remote Libraries

Option 21, **Manage Remote Libraries and LCPs** on the **OpenVault Configuration Menu**, enables LCPs that connect to the OpenVault server from OpenVault clients. When this option is chosen, ov_admin presents the following menu:

```
Manage remote Libraries and LCPs Menu

          1 - Create a new Library
          2 - Delete a Library
          3 - Show all existing Libraries

          4 - Activate another LCP for an existing Library
          5 - Deactivate a LCP
          6 - Show all activated LCPs

          r - Return to Main Menu.
          q - Exit.

        Which operation would you like to do:
```

1. Choose option 1 to create a new library. You are prompted:

   ```
   Enter the name of the Library you want to create []
   ```

2. If you chose a name for this library in the configuration worksheet, enter that name.

3. The ov_admin script then displays the following prompts and messages:

```
Library 9714 was successfully created
Do you want to activate a LCP for this Library? [Yes]
Enter the name of the Host where the LCP for Library "9714" will run [vega]
Enter the LCP's instance name [9714@vega]
What security key will the LCP use [none]
The LCP "9714@vega" for Library "9714" was successfully activated on "vega".
Press enter to continue...
```

4. Repeat this process for all remote libraries you want to configure.

Tip: Use the worksheets to record the names you select. You will be asked to enter them again when configuring the library or drives on a remote OpenVault client host.

---

5. After this, the `ov_admin` script prompts for remote libraries.

2.5.4.3 Enabling Remote Drives

To configure the OpenVault server host for remote drives, choose option 22, **Manage Remote Drives and DCPs** from the **OpenVault Configuration Menu**. The following menu is displayed:

```
Manage remote Drives and DCPs Menu

          1 - Create a new Drive
          2 - Delete a Drive
          3 - Show all existing Drives

          4 - Activate another DCP for an existing Drive
          5 - Deactivate a DCP
          6 - Show all activated DCPs

          r - Return to Main Menu.
          q - Exit.

        Which operation would you like to do:
```

1. Choose option 1 to create a new remote drive. You are then prompted:

```
Enter the name of the Drive you want to create []
```

2. If you chose a name for this library in the configuration worksheet, enter that name.
3. The following prompt appears:

```
The following Drive Groups currently exist:

          drives

  What Drive Group do you want to use (or create)? [drives]
```

4. Press Enter to choose the default, or enter the name of an existing drive group, or enter the name of a drive group to be created. If you enter the name of a drive group to be created, you are prompted for more information about it.
5. The following prompts and messages appear:

```
Drive drive1 was successfully created.
Do you want to activate a DCP for this Drive? [Yes]
Enter the name of the Host where the DCP for
     Drive "9714-top-dlt" will run [vega]
Enter the DCP's instance name [9714-top-dlt@vega]
What security key will the DCP use [none]
The DCP "9714-top-dlt@vega" for Drive "9714-top-dlt"
     was successfully activated on Host "vega".

Press enter to continue...
```

2.5.5 When to Import Media

After you configure locally attached drives and libraries, the script asks to import tapes that were discovered in the newly configured libraries. If you have libraries on remote OpenVault client hosts, wait until those libraries are configured, then invoke the import function as described in the section Section 2.7, page 60. If you do not have any remote libraries, you may import media right away.

2.6 Configuring the OpenVault Clients

After configuring the OpenVault server, it is time to configure remote OpenVault clients, in any order. Consult the host worksheet that you generated in Section 2.4, page 20, for the list of OpenVault client hosts.

To configure an OpenVault client host, follow these steps:

1. Log in to the remote host as `root`.
2. Start the `ov_admin` script:

   # **/usr/sbin/ov_admin**

3. The `ov_admin` script outlines the general configuration strategy:

```
OpenVault Configuration
  The general strategy for setting up OpenVault is to
    1) configure the OpenVault server
    2) configure LCP/DCPs on the server machine
    3) configure server for local Applications
    4) if needed, configure server for remote LCPs, DCPs, and Apps
    5) if needed, install and configure LCP/DCPs on remote machines
    6) from the server, setup/import media for each library
```

4. The `ov_admin` script then determines the name of the OpenVault server host, the OpenVault server port number, and the OpenVault security key that you chose while configuring the OpenVault server.

   a. The script prompts for the name of the OpenVault server host:

      ```
      What is the name of the OpenVault Server? [vega]
      ```

      Enter the name. The default presented in this prompt is the hostname of the machine on which you are running `ov_admin` script.

   b. The script prompts for the port number of the OpenVault server host:

      ```
      What is the port number OpenVault is using? [44444]
      ```

      If you chose another port number when you configured the OpenVault server, enter that port number now; otherwise, accept the default.

   c. The script prompts for the security key of the OpenVault server host:

      Enter the security key that you chose when you configured the OpenVault server. If you did not select security at that time, accept the default.

   ```
   What default security key would you like to use? [none]
   ```

Note: Specifying the exact values for the OpenVault server's hostname, port number, and security key is critical for proper functioning of all OpenVault components.

5. The ov_admin script then provides the menu options shown Example 2-2:

Example 2-2 OpenVault Configuration Menu Options

```
OpenVault Configuration Menu

      Configuration on Machines Running LCPs and DCPs
         1 - Manage LCPs for locally attached Libraries
         2 - Manage DCPs for locally attached Drives

      Configuration on Admin-Enabled Machines
        11 - Manage Cartridge Groups
        12 - Manage Drive Groups
        13 - Import Media

         q - Exit.

Which operation would you like to do?
```

6. Configure all libraries and drives that you plan to attach to this host.

2.6.1 Configuring Attached Drives on OpenVault Client Hosts

Select the **Manage DCPs for Locally Attached Drives** option from the **OpenVault Configuration Menu** to configure drives that are attached to the host. The ov_admin script displays the following menu:

```
OpenVault DCP Configuration Menu

        1 - Create a new DCP
        2 - Modify a DCP
        3 - Delete a DCP
        4 - Start a DCP
        5 - Check status of DCPs
        6 - Stop a DCP
```

```
        r - Return to Main Menu.
        q - Exit.

    Which operation would you like to do:
```

Choose option 1 to create a new DCP.

The set of dialogs presented are similar to the dialog when you configure drives on the OpenVault server. The main difference is that the script asks you to enter the OpenVault name for the drive, but does not offer a default name.

Consult your drive worksheet to make certain that the names match. A sample dialog is shown below:

Example 2-3 Drive Configuration on a Client Host

```
OpenVault -- DCP Creation Menu

          1 - /dev/rmt/tps3d1
          2 - /dev/rmt/tps3d2
          3 - /dev/rmt/tps3d3
          4 - /dev/rmt/tps3d4
          5 - /dev/rmt/tps7d2
          5 - /dev/rmt/tps7d3

        r - Return to Previous Menu.
        q - Exit.

For which Drive would you like to add a DCP: 1
Enter a name for the device [] redwood1

What instance name would you like to give
    to this DCP? [redwood1@vega]
What security key would you like to give to this DCP? [none]
What polling interval would you like this DCP to use? [30]
What directory should hold handles for this
     DCP? [/var/opt/openvault/clients/handles]
Configuring STK-redwood at /dev/rmt/tps3d1 to be "redwood1"


DCP Configuration Parameters:
```

```
Drive name                  :  redwood1
DCP name                    :  redwood1@vega
DCP type                    :  STK-redwood
OpenVault Server host name  :  ursa
OpenVault Server port number:  44444
Security key                :  none
DCP polling interval        :  30
Handle creation directory   :  /var/opt/openvault/clients/handles
Drive access path           :  /dev/rmt/tps3d1
Direct SCSI access path     :  /dev/scsi/sc3d1l0

Create the DCP now? [Yes]
DCP successfully created
```

The DCP has been started. If you have not already done so, you need to create the Drive and activate the DCP on the OpenVault server so that the DCP can connect to the server.

2.6.2 Configuring Attached Libraries on OpenVault Client Hosts

Choose the **Manage LCPs for Locally Attached Libraries** option from the `ov_admin` **OpenVault Configuration Menu** to configure attached libraries on an OpenVault Client Host.

The script scans the installed hardware and software to determine which libraries are available for configuration. Each detected library is offered for configuration. The set of dialogs presented are similar to the one when you configure libraries on the OpenVault server. The main difference is that the script asks you to enter the OpenVault name for the library, but does not offer a default name.

It is critical that you enter the OpenVault name for each library exactly as you entered it (during server configuration when enabling remote libraries).

1. Consult your library worksheet to make certain that the names match. A sample dialog for a SCSI-attached library is shown in Example 2-4.

Example 2-4 Library Configuration on a Client Host

```
OpenVault LCP Configuration Menu

         1 - Create a new SCSI LCP
```

```
         2 - Create a new network LCP

         r - Return to Main Menu.
         q - Exit.


    Which operation would you like to do: 1

    OpenVault -- SCSI LCP Creation Menu

        1 - /dev/scsi/sc7d1l0

        r - Return to Previous Menu.
        q - Exit.

    For which SCSI library would you like to add a LCP:1
    Enter a name for the device [] 9714
    What instance name would you like to give to this LCP? [9714@vega]
    What security key would you like to give to this LCP? [none]
    What polling interval would you like this LCP to use? [30]
    Configuring  STK-9700 at /dev/scsi/sc7d1l0 to be "9714"

    What is the default shape of the slots in the library? [DLT]
    For the drive at location "first drive from BOTTOM"
        Enter a drive name for the element address "1030": 9714-bottom-dlt
    For the drive at location "second drive from BOTTOM",
        Enter a drive name for the element address "1031": 9714-top-dlt

LCP Configuration Parameters:

Library name                :  9714
LCP name                    :  9714@vega
LCP type                    :  STK-9700
OpenVault Server host name  :  ursa
OpenVault Server port number:  44444
Security key                :  none
LCP polling interval        :  30
Number of drives            :  2
Direct SCSI access path     :  /dev/scsi/sc7d1l0
Default cartridge shape     :  DLT
```

```
Drives in the Library

Drive Name       Drive Address
-----------      -------------
9714-bottom-dlt  1030
9714-top-dlt     1031

Create the LCP now? [Yes]
LCP successfully created
```

The LCP has been started. If you have not already done so, you need to create the Library and activate the LCP on the OpenVault server so that the LCP can connect to the server.

2. The `ov_admin` script prompts you for OpenVault drive names contained in the library.

   - Type names exactly as you did when enabling those drives on the OpenVault server. Consult your library worksheet for details. You must match the location description string for each contained drive with the corresponding description string in the library worksheet.
   - If you have drives in this library that are connected to another host, enter the exact OpenVault name for these drives, even if you have not yet configured the drives on that host. Consult your library worksheet for this library to get this information.
   - Configuring a non SCSI-attached library on an OpenVault client host follows the same procedure as configuring one on the OpenVault server host.

3. At this point the library is configured and the LCP is started. To verify current status of the LCP, run the `ov_stat` command on the OpenVault server.

2.7 Importing Media

After you have configured the libraries on the OpenVault server and all OpenVault clients, you need to import media to make it available for application use. Importing media is how the OpenVault server learns about each piece of media. Each tape that applications use must be imported before it can be made available for allocations and mounts. See Chapter 3, page 69, for more information.

Caution: If you have media that contain data in a library, or if media are known to certain sensitive applications, take precautions so that other applications do not accidentally modify these media (for example, by defining a cartridge group for the dedicated used by only the one application). Refer to the application's documentation for importing media known to that application only. **Failure to do so could lead to data loss.**

It is best to invoke the import media function of the `ov_admin` script only after you have configured all libraries, including libraries on remote OpenVault client hosts.

The import function is available only on the OpenVault server host.

1. To import media into a library, you must identify a cartridge group for each cartridge. The `ov_admin` script imports all media found in a library into the same cartridge group. By default, media are imported into the default cartridge group called `carts`. If you would like to add more cartridge groups, do so now. To import media using multiple cartridge groups, see Chapter 6, page 109.

2. You must also identify a cartridge type for each cartridge. The `ov_admin` script assumes that all media in a library is of the same cartridge type. If this is not true, skip automatic import of media and follow the procedures described in Chapter 3, page 69.

3. The import function allows you to pre-allocate all media to an application. If you wish to import media that is pre-allocated to other applications, skip automatic import of media and follow the procedures described in Chapter 3, page 69.

   You may import media that is not pre-allocated. If you do, make sure that OpenVault applications know how to import media. For example, the media mounting application `ov_umsh` provided with OpenVault knows how to allocate media.

4. To use the import function, enter the `/usr/sbin/ov_admin` command as `root` on the OpenVault server host, and select the **Import Media** option 13, as shown in Example 2-5:

   Example 2-5 Importing Media

```
OpenVault Configuration Menu

  Configuration on Machines Running LCPs and DCPs
     1 - Manage LCPs for locally attached Libraries
```

```
    2 - Manage DCPs for locally attached Drives

  Configuration on Admin-Enabled Machines
    11 - Manage Cartridge Groups
    12 - Manage Drive Groups
    13 - Import Media

  Configuration on the OpenVault Server Machine
    21 - Manage remote Libraries and LCPs
    22 - Manage remote Drives and DCPs
    23 - Manage Applications and Admin CLI Tools

     q - Exit.

Which operation would you like to do: 13
Would you like to import media ? [Yes]
```

5. The `ov_admin` script checks for all configured libraries containing media to import, and shows each of the libraries individually.

   ```
   Would you like import media from the library 9710 [Yes]
   ```

   Accept the default if you are ready to import media in that library; otherwise, enter **`No`**.

6. The script continues:

   ```
   Would you like to add ALL cartridge to the SAME Cartridge group [Yes]
   ```

   If you decide to import all cartridges in this library into the same cartridge group, accept the default. Otherwise, enter **`No`** to have the script skip importing media from this library.

2.7.1 Selecting Cartridge Types

1. Known cartridge types are presented for your inspection, including types that might be unavailable. Choose the type that corresponds to the brand name on your cartridges.

   Note: Because 3590 cartridges are the same size as 3480 cartridges, they are said to have a 3480 slot type, which is sometimes a source of confusion.

2. Once you select the cartridge type, available cartridge groups are displayed. Enter the cartridge group of which all cartridges in this library should be members.

```
Cartridge Groups available are:
                      1. carts
Select the Cartridge group for the cartridges in this library:
```

3. Once you select a cartridge group, the script asks about pre-allocation. If you want to pre-allocate cartridges to an application, accept the default. If not, see Section 2.7.2, page 63.

```
Would you like to pre-allocate cartridges to specific application [Yes]
```

4. If you answer **Yes**, you are prompted for an application name, `dmf` in this example:

```
Enter name of application for cartridge pre-allocation [ov_umsh] dmf
```

   If the application name you enter does not exist, the `ov_admin` script creates it for you.

5. Once the import completes successfully, you see output similar to the following:

```
Created Application: dmf
Cartridge-group-application creation:
                     Application: dmf
                     Group: carts
                     Importing tapes with:
                         Cartridge Type = DLT2000
                         Cartridge Group = carts
                         Application = dmf
This may take a while...
Finished importing media from library, 9710
```

2.7.2 Not Pre-allocating Cartridges

If you do not wish to pre-allocate cartridges, enter **No** after the following prompt:

```
Would you like to pre-allocate cartridges to specific application [Yes]
```

Cartridges in this library are imported without being pre-allocated to any application. Once the import completes successfully, you see output similar to Example 2-6:

Example 2-6 Importing Media without Pre-allocating Cartridges

```
Importing tapes with:
          Cartridge Type = DLT-7000
          Cartridge Group = carts
This may take a while ...
Finished importing media from library, 9730
```

2.8 Custom Installation

On IRIX systems, it is possible to do a custom installation, which allows you to install only the subsystems that you need. Before doing a custom install, you must determine the types of all OpenVault-managed libraries and drives on all systems.

- For drives, see Section 2.8.1, page 64.
- For SCSI libraries, see Section 2.8.2, page 65.
- For other libraries, see Section 2.8.3, page 65.

Remember to repeat the steps explained in this section for every host with an attached library or drive that is to be managed by OpenVault.

2.8.1 Determining Attached SCSI Drives

On SGI ProPack for Linux and IRIX systems, you can use the `ov_scandev` command, which is part of OpenVault to determine the connected SCSI drives. The command display includes vendor name, path to the drive, and drive product name, as shown in Example 2-7. See the man page for more details.

Example 2-7 `ov_scandev` Drive Output (IRIX)

```
#  ./ov_scandev -D -p vendor,product,serial_number,wwn -l
/dev/rmt/5005076300002d33/lun0/c33p400000
         vendor : IBM
        product : 03590E11
  serial_number : <unknown>
            wwn : 5005076300002d33
/dev/rmt/500104f0004395f0/lun0/c33p1
         vendor : STK
        product : 9840
  serial_number : 331000031498
```

```
            wwn : 500104f0004395f0
/dev/rmt/500104f000425829/lun0/c33p1
         vendor : STK
        product : 9840
  serial_number : 331000030387
            wwn : 500104f000425829
```

This sample output is unrelated to Figure 2-1, page 16 and Figure 2-2, page 17.

2.8.2 Determining Attached SCSI Libraries

On SGI ProPack for Linux and IRIX systems, you can use the `ov_scandev` command, which is part of OpenVault to determine the connected SCSI libraries. As shown in Example 2-8, the command can display the library vendor name, library product name, and path. See the man page for details.

Example 2-8 `ov_scandev` Library Output (IRIX)

```
#  ./ov_scandev -L -p vendor,product,serial_number,wwn -l
/dev/scsi/sc21d5l0
         vendor : STK
        product : 9714
  serial_number : <unknown>
            wwn : <unknown>
/dev/scsi/5005076300002d33/lun1/c33p400000
         vendor : IBM
        product : 03590E11
  serial_number : <unknown>
            wwn : 5005076300002d33
```

This sample output is unrelated to Figure 2-1, page 16 and Figure 2-2, page 17.

2.8.3 Network Libraries

OpenVault supports several network libraries. If you plan to manage this type of library, you must install the appropriate `OpenVault.lcp` subsystems on IRIX systems. On SGI ProPack for Linux systems, all of the LCPs provided by SGI are included in the `openvault-sw` package.

For any library that you plan to manage with OpenVault, configure one and exactly one controlling LCP. Because the above-mentioned libraries are not SCSI-attached,

you may designate any OpenVault host system to run the LCP. You must install the appropriate `OpenVault.lcp` subsystem on that chosen system. For simplification, it might be best to designate the OpenVault server as the LCP host for non-SCSI-attached libraries.

Note: If you unintentionally install the LCP for a network LCP on an OpenVault host, the OpenVault `ov_admin` menus will present it as an option for installation. Do not select it unless you want to configure it.

2.8.4 Other Guidelines for Custom Installation

This section offers guidelines for selecting OpenVault components to install on IRIX systems. On SGI ProPack for Linux systems, all OpenVault server commands and executables, administrative tools, LCPs, and DCPs are included in the `openvault-sw` package; OpenVault sample source for applications and `include` headers are included in `openvault-dev` packages.

- For OpenVault server hosts:
  - Required

```
OpenVault.sw.config         OpenVault setup scripts
OpenVault.sw.core           OpenVault core servers
OpenVault.sw.admin          OpenVault administrative tools
OpenVault.sw.startstop      OpenVault scripts to start & stop daemons
OpenVault.upgrade.files     OpenVault scripts required for upgrading
                                    Openvault installations
```

  - Recommended

```
OpenVault.man.manpages      OpenVault manual pages
OpenVault.man.relnotes      OpenVault release notes
OpenVault.docs.adminguide   OpenVault Administrator's Guide
```

  - Optional

```
OpenVault.sw.user           OpenVault end-user tools
OpenVault.dev.examples      OpenVault sample source for applications
OpenVault.dev.include       OpenVault app C/C++ include headers
```

  - As needed

```
          OpenVault run-time libraries for DCPs
OpenVault.dcp.XXXX          Appropriate DCP susbsystem
OpenVault.lcp.XXXX          Appropriate LCP susbsystem
```

- For OpenVault client hosts:
  - Required

    ```
    OpenVault.sw.config         OpenVault setup scripts
    OpenVault.upgrade.files     OpenVault scripts required for upgrades.
    ```

  - Not recommended

    ```
    OpenVault.sw.core           OpenVault core servers
    ```

  - Recommended

    ```
    OpenVault.man.manpages      OpenVault manual pages
    OpenVault.man.relnotes      OpenVault release notes
    OpenVault.sw.admin          OpenVault administrative tools
    OpenVault.docs.adminguide   OpenVault Administrator's Guide
    ```

  - Optional

    ```
    OpenVault.sw.user           OpenVault end-user tools
    OpenVault.dev.examples      OpenVault sample source for applications
    OpenVault.dev.include       OpenVault app C/C++ include headers
    ```

  - As needed

    ```
    OpenVault run-time libraries for DCPs
    OpenVault.dcp.XXXX          Appropriate DCP susbsystem
    OpenVault.lcp.XXXX          Appropriate LCP susbsystem
    OpenVault.sw.startstop      OpenVault scripts to start & stop daemons
    ```

The `OpenVault.sw.startstop` and `OpenVault.sw.config` and `OpenVault.upgrade.files` subsystems are required on each OpenVault host that contains an OpenVault DCP or LCP.

The installation tool enforces these requirements.

Cartridge Life Cycle

This chapter describes the OpenVault cartridge life cycle and the administrative tools used to manage it. More information on the administrative tools is available in the man page for each command.

The *life cycle* of a cartridge is the chain of states and events that affect a cartridge from the time that it first becomes part of a system until in ceases to be a part of that system. The major events in the life of a cartridge include:

- Physical and logical introduction of the cartridge into the system
- Assignment of ownership (who or what application gets to use the cartridge)
- Use of the cartridge by applications
- Recycling of a cartridge when one owner no longer needs it
- Disposal of the cartridge either by sending it to another system or removing it for disposal when the media reaches the end of its service life

3.1 Cartridge States

Figure 3-1, page 70, illustrates the life cycle of a cartridge:

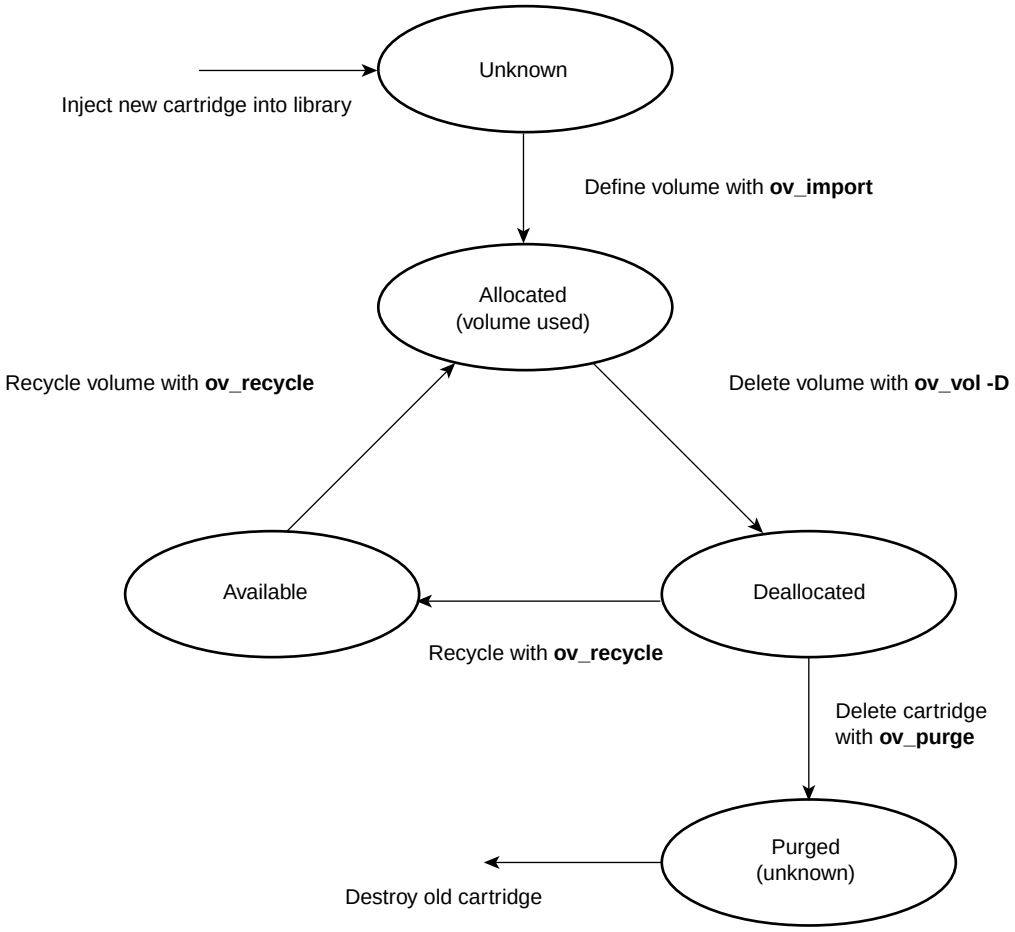

Figure 3-1 Cartridge Life Cycle (Simplified)

During this life cycle, these are the possible cartridge states:

Unknown — If no information is available about a cartridge, then its state is "unknown." A new cartridge (one that OpenVault has not seen before) is in the unknown state. OpenVault needs to become aware of a cartridge before it can do anything with it.

Defined — OpenVault needs to have certain information about a cartridge before it can be used. The majority of the information must come from a

person—the OpenVault administrator—who knows what the cartridge is and can describe it to OpenVault. This includes the physical cartridge label (PCL, usually an optically readable label on the cartridge), the cartridge type, and information on the number of sides on the cartridge. All this information constitutes the cartridge definition.

Available — In addition to knowing that a cartridge exists, OpenVault needs to know where information can be written on the cartridge, and whether an application has reserved that area. This information comprises the partition description for the cartridge. Once this information is entered the cartridge may be considered "available." That is, the cartridge may be assigned ("allocated") to an application that needs one.

Allocated — For applications, OpenVault is a service for allocating and mounting volumes. Applications make a request to write data in a known place and manner. That place is a logical volume on a partition on a cartridge. The application asks the server to allocate a volume for it. The server looks for cartridges with available partitions that are not used by any application. When it finds one, the server modifies cartridge ownership information to show that the requesting application has control over that cartridge. It marks the partition as unavailable, and creates volume information to describe the location of data stored on the cartridge. At this point the cartridge is considered "allocated" and the volume can be used as long as the application desires to do so.

Deallocated — When an application is finished with a cartridge, it can "deallocate" it. The server removes volume information for the cartridge; so the application can no longer request mounting of that volume. Information on data location is deleted and becomes unavailable. The application still owns the cartridge, however, giving the administrator some control over how to handle the cartridge thereafter. The "owned and deallocated" state can be used as a marker for a cartridge to be erased, destroyed (if the media is at the end of its useful life), or recycled without erasure. An administrator (or an administrative application) can mark the cartridge as available again without doing anything else to it. The cartridge becomes available for volume allocation, but only to the application that owns it (ownership remains as before). The cartridge must be recycled before other applications can use it.

Recycled — An administrator (or an administrative application) may recycle deallocated cartridges. This entails marking the cartridge as available for allocation and removing ownership information. Once the cartridge is recycled, it is again in the "defined" and "available" state, but other

information about the cartridge may be retained. This could include a count of the number of times that cartridge has been written so that its remaining life expectancy can be tracked.

Purged — All information about a cartridge can be removed from the system, returning the cartridge to "unknown" state. This is typically done only when a cartridge has permanently left the OpenVault system, as when a cartridge has reached the end of its life, or when it is removed from the system for transfer elsewhere. There is no "purged" state as such; this term simply provides a way of referring to cartridges for which there once was descriptive data, all of which has been removed.

3.2 Managing Cartridges

Cartridges are managed in OpenVault through an administrative interface (AAPI). A set of command line tools are included that format requests to the OpenVault server and display the results on a standard terminal window.

3.2.1 Physical Entry into a Library

Cartridges must be physically placed into a library before they can be used. There are two ways to accomplish this. One is to simply open the door to a library and place cartridges with barcodes in slots in the library. When the door is closed, the library control program (LCP) will command a scan of the slots in the library and report the contents of the library to the OpenVault server. This probably results in the library being offline while the door is open.

A second method is to use the inject/eject slot of the library if it is so equipped. Such a library will remain operational while cartridges are being injected or ejected. Some libraries have an inject slot that is always open and ready to receive cartridges. For these libraries, one can simply insert a cartridge into the slot. The LCP will recognize the event and handle the cartridge appropriately.

Other libraries need to have their inject/eject slot opened under program control. In this case you may use the `ov_inject` command to instruct a library on how many cartridges to accept. A typical command would be:

```
ov_inject -l biglib -n 3
```

This tells OpenVault that you wish to insert three cartridges into the inject slot on a library named `biglib`.

Note that this feature has not been fully implemented for most libraries at the time of this writing. For now, the most reliable method of placing carts into a library is to take the library offline, open the door, and insert cartridges directly into storage slots.

3.2.2 Examining the Contents of a Library

It is often useful to see what a library believes to be contained within it. The `ov_stat` command performs this function (among others), displaying the contents of the slots in a library as shown in Example 3-1:

Example 3-1 Library Contents

A typical `ov_stat` command would be:

```
ov_stat -L 9730 -s
```

The `-s` option specifies that a slotmap should be displayed, and the `-L9730` indicates that it should be for the library named `9730`. It produces the following output:

```
# ov_stat -L 9730 -s
Library Name   Broken   Disabled   State   LCP SoftState  LCP HardState
9730           false    false      ready   ready          ready
Library: '9730'
Library    Slot Name      Slot Type  Occupied   PCL        Cart ID
9730       slot 0          DLT        false
9730       slot 1          DLT        false
9730       slot 10         DLT        true       000199
9730       slot 11         DLT        true       ABN695
9730       slot 12         DLT        false
9730       slot 13         DLT        true       000195
9730       slot 14         DLT        false
9730       slot 15         DLT        false
9730       slot 16         DLT        true       000193
9730       slot 17         DLT        false
9730       slot 18         DLT        false
9730       slot 19         DLT        false
9730       slot 2          DLT        false
9730       slot 20         DLT        false
9730       slot 21         DLT        false
9730       slot 22         DLT        true       000185
9730       slot 23         DLT        false
9730       slot 24         DLT        false
```

```
9730       slot 25         DLT        false
9730       slot 26         DLT        false
9730       slot 27         DLT        false
9730       slot 3          DLT        false
9730       slot 4          DLT        false
9730       slot 5          DLT        false
9730       slot 6          DLT        false
9730       slot 7          DLT        false
9730       slot 8          DLT        true       000194
9730       slot 9          DLT        true       000192
Library    Bay Name        Slot Type  Total Slots  Free Slots
9730       bay 1           DLT        7            0
```

This example shows a library with 7 slots occupied, with the PCL listed for the cartridge in each occupied slot. The PCL information is needed to enter the data that OpenVault needs to manage cartridges. Note that at this point, the LCP has information about the PCLs of cartridges in this library, but the OpenVault server has no other information about the cartridges. Thus these cartridges are still “unknown” to OpenVault.

3.2.3 Creating Cartridge Data

The administrator (or an administrative application) must enter descriptive data about cartridges before they can be used. OpenVault provides two methods to enter that data:

- Individual commands to enter facts about the cartridge, partitions, and volumes on the cartridge (using `ov_cart`, `ov_part`, and `ov_vol`)
- `ov_import`, a tool that enters several of these facts at once

The individual commands will be discussed first as they illustrate the types of data needed. The `ov_import` command combines several steps of the individual commands and is discussed in Section 3.2.11, page 81.

Cartridge data is entered with the `ov_cart` command. When outside of a library, a new cartridge (one with no data on it) could be entered as follows:

```
ov_cart -n -B ABN695 -T DLT7000 -g default
```

This creates an entry in the OpenVault catalog for a cartridge with a PCL of `ABN695`. The cartridge is a DLT 7000 cartridge, which is indicated to OpenVault with the `-T` (cartridge type) option and type description string of `DLT7000`. There is a unique

description string for each type of cartridge known to the OpenVault system. Other examples include `8mm-160m`, `3590`, and `STKtimberline`. These strings tell OpenVault what kind of drive this tape uses. The `-g` option is used to specify a cartridge group. Cartridge groups are used to control which cartridges can be allocated by an application. In this case, the cartridge is being assigned to the group default, which would normally be used for cartridges which can be allocated by any application.

The `ov_cart` command generates output as follows:

```
New cart created with cartridge ID = 'wIS2MTXrMmQAAlYR'
```

The *cartridge ID* is a unique string that identifies a cartridge, and is used to define other cartridge data. A PCL alone cannot be used as a unique identifier because a PCL might not be unique on an OpenVault system—two different types of cartridges could have the same PCL. OpenVault requires that on a given system a PCL must be unique for a single cartridge type (there cannot be two DLT7000 tapes with the same PCL).

If the cartridge was not a new cartridge but rather already belonged to an application and had data on it, one additional item of data would need to be entered. This would be the name of the application that owns the cartridge. For example, if the above cartridge were used by the `xfsdump` application, the command would be:

```
ov_cart -n -B ABN695 -T DLT7000 -g default -A xfsdump
```

This ensures that no other application can allocate this cartridge and overwrite its data.

When a cartridge is created, OpenVault also creates additional descriptive data about the cartridge based on its knowledge of that cartridge type. This is primarily information about sides on the cartridge, including how many sides the cartridge has, the name of each side, and creation time. Cartridge type information can be seen by running the `ov_carttype` command with no arguments. For the DLT700, this information displays:

```
Type Name  Media Type  Media Length  Slot Type  Number Sides  Side1Name Side2Name
DLT7000    DLT Compact IV   100      DLT        1             SideA
```

Most magnetic tapes have only one side. Optical and removable magnetic disk cartridges have two sides. At the time of this writing, OpenVault supports single sided media only.

3.2.4 Displaying Cartridge Data

Once data on a cartridge is entered into the system, you can display it with the `ov_lscarts` command as shown in Example 3-2:

Example 3-2 `ov_lscarts` Cartridge Data

The first entry provides summary information on cartridges, and the second shows the PCLs of all cartridges known to the system.

```
# ov_lscarts
num carts num allocd num free
       15          7        8
# ov_lscarts '.*'
000185          000193          000197          ABN579          ABN695
000190          000194          000198          ABN580          ABN697
000192          000195          000199          ABN693
```

Without any arguments, `ov_lscarts` prints a summary of the number of cartridges in a system. This is because the number of cartridges in a system could be very large. If the default behavior was to show all cartridges, the screen could be completely filled with cartridge information. Thus `ov_lscarts` requires that a cartridge list be provided before it prints detailed information.

The argument '.*' in the second example is a regular expression that matches all cartridge names. This is the standard way of displaying all known cartridges.

You can obtain detailed information on cartridge(s) can be shown in Example 3-3:

Example 3-3 Detailed `ov_lscarts` Cartridge Data

```
# ov_lscarts -li ABN695
PCL      cart type   owner   state   cartID           part     volume
ABN695     DLT7000           ok      wIS2MTXrMmQAAlYR
```

In this example, the `-l` option specifies a long listing, and the `-i` specifies that the CartridgeID be printed also.

Note: The example does not contain any partition or volume information, because it has not yet been entered.

3.2.5 Creating Partitions

A *partition* is a defined area on the media where data can be written. For OpenVault to be able to use a cartridge, at least one partition must be defined on it. Magnetic tape cartridges usually have only one partition. Optical cartridges often have one partition per side. As SGI tape driver software typically does not support multiple partitions per cartridge, one partition is currently the limit for tapes on OpenVault.

Example 3-4 shows how you create a partition with the `ov_part` administrative command:

Example 3-4 `ov_part` Partitions Creation

```
# ov_part -n -C wIS2MTXrMmQAAlYR -s SideA -p "Part 1" -b DLT7000
New partition created:
  cartridge ID = 'wIS2MTXrMmQAAlYR', partition Name = 'Part 1'
```

The `-n` option indicates that this is a new partition on the cartridge with the CartridgeID `wIS2MTXrMmQAAlYR`. Note that this is the same CartridgeID reported by `ov_cart` above when the cartridge data was entered for the cartridge with the PCL of `ABN695`. The `-s` option identifies the side of the cartridge on which to put the partition. Since there is only one side on a DLT7000 cartridge, there is no choice of side here—the side name of `SideA` from the cartridge type table (as shown by `ov_carttype` above) must be used.

Finally, a name must be provided for the partition. This follows the `-p` option and in this case is `Part 1`. Any name may be specified here.

The `ov_part` command reports whether the partition creation was successful. If it was not, an error message is displayed.

At present there is no easy way for users to display partition information on a cartridge. The information is not needed unless you need to create a volume on the partition. If you need to do this, be sure to remember the name of the partitions you create. By convention, the standard name for the first partition is `Part 1` which must by typed in quotes on the command line. The quotes are needed so that the space between "Part" and "1" is not recognized as a delimiter between arguments on the command line.

3.2.6 Controlling Allocation Status

At this point, the cartridge is fully defined and available. An application that requests a volume be allocated to it could get ownership of this cartridge. That is what we

want to have happen for new cartridges that do not have data on them. However, if the cartridge already has data and we do not want it to be overwritten, we need to make sure that no application can allocate it.

The best time to do this is when the partition is created; otherwise, an application may be able to allocate the cartridge before you can type a new command to prevent it. Example 3-5 shows how you can set allocatable status on the `ov_part` command with the `-a` option when create the partition is created:

Example 3-5 Setting `ov_part` Allocatable Status

```
# ov_part -n -B wIS2MTXrMmQAAlYR -s SideA -p "Part 1" -b DLT7000 -a false
```

This command is identical to the `ov_part` command in Section 3.2.5, page 77, except for the addition of the `-a` option and its argument.

A better way to define cartridges with pre-existing data is to use the `ov_import` command, described in Section 3.2.11, page 81.

3.2.7 Allocating Volumes

The above steps together fully define a cartridge—OpenVault now has cartridge, side, and partition information. All that is needed for an application to use the cartridge is for a volume to be created on it. Example 3-6 shows how you use the `ov_vol` administrative command to create a volume.

Example 3-6 `ov_vol` Volume Allocation

This `ov_vol` command creates a volume named `VolOne` that will belong to an application named `ov_umsh` on the cartridge previously defined:

```
# ov_vol -n -v VolOne -a ov_umsh -C wIS2MTXrMmQAAlYR -s SideA -p "Part 1"
New volume created:
  volume name = 'VolOne', application name = 'ov_umsh'
```

The `-n` option specified that this is a new volume, with the following argument containing the name for the volume of `VolOne`. The `-a` option specifies the owner application, in this case being `ov_umsh`. The following three option/argument pairs specify where the volume is to be created—on the cartridge with CartridgeID `wIS2MTXrMmQAAlYR`, on its side `SideA` and partition "`Part 1`".

Applications may also create volumes by issuing an `allocate` request to OpenVault through the CAPI interface. OpenVault looks for an allocatable partition on a cartridge that is either not owned by any application or is owned by the requesting

application. It then creates volume data for one such partition, marks the partition as not allocatable (because there is now a volume allocated on it), and sets the cartridge ownership to the requesting application. See the *OpenVault Application Programmer's Guide* for further details.

3.2.8 Displaying Volume Data

Information on volumes can be displayed with the `ov_lsvols` command shown in Example 3-7:

Example 3-7 `ov_lsvols` Volume Data

```
# ov_lsvols -l VolOne
volume     cartID            owner          partition
VolOne     wIS2MTXrMmQAAlYR  ov_umsh        Part 1
```

The `-l` option specifies that a long listing be printed.

This command works similarly to the `ov_lscarts` command; issuing the command with no arguments causes summary information on volumes to be displayed, and a listing of all volumes requires a '.*' regular expression. The `-l` option specifies that a long listing be printed.

3.2.9 Deallocating Volumes

When an application has no further use for a volume it may give that volume back to OpenVault for reuse. It does this by deallocating the volume, which deletes volume data. The application may do this through the CAPI interface with the `deallocate` command, or an administrator may perform the deallocation using the `ov_vol` command as follows:

```
ov_vol -D -v VolOne -a ov_umsh
```

The `-D` option indicates that the volume should be deleted. The `-a` option indicates which application owns the volume that is to be deleted. The application name is needed because while volume names must be unique within one application, they do not have to be unique across all applications on an OpenVault system. Specifying the application name indicates which specific volume with the given name is to be deleted.

3.2.10 Recycling Cartridges

OpenVault retains a fair amount of information about a cartridge after a volume on it is deallocated. The owner information (that is, which application owned the cartridge while there was a volume allocated on it) remains, as does the side and partition information. This allows administrative operations to be performed on these cartridges. A typical operation might be to erase the tape before it is released for use by other applications or users in order to protect sensitive data. Such operations are optional (and beyond the scope of this document); there is no requirement that they be performed. After such steps are performed, the volume can be *recycled*, that is made available for allocation again.

The simplest method of recycling allows the owner of the cartridge to allocate another volume on it. To do this, you must tell OpenVault that this operation is allowable. This is done with the `ov_part` command as follows:

```
ov_part -C wIS2MTXrMmQAAlYR -s SideA -p "Part 1" -a true
```

The CartridgeID identifies the cartridge on which the partition resides. The `-s` and `-p` options indicate the side and partition name, as when the partition was created. The `-a` option allows setting of the allocatable status. Setting it to true allows the partition to be allocated, that is it allows a volume to be created on that partition.

Again, since the application that previously had a volume on the partition still owns the cartridge, only that application can allocate a new volume on that cartridge. Freeing up the cartridge so that any application can use it requires use of the `ov_recycle` command.

The `ov_recycle` command looks for cartridges that do not have any volumes on them and which are not available for allocation. The search can be restricted to cartridges belonging to a particular application, or to one or a list of cartridges. A typical `ov_recycle` command looks like this:

```
ov_recycle -r -A ov_umsh
```

The `-A` option indicates that all cartridges belonging to application `ov_umsh` that can be recycled should be recycled. Any such cartridges would have their owner information deleted, and their partitions would become allocatable again. This operation returns the cartridges to the defined and allocatable states described above, just as for a new cartridge with defined cartridge and partition information.

3.2.11 Simplified Entry of Media Information

A large amount of data can be entered at once, and data is shared between steps of the definition process so that the administrator does not have to read and re-enter intermediate results.

There is a simpler and faster method to define cartridge, partition, and volume data than the procedure using `ov_cart`, `ov_part`, and `ov_vol` that is described in Section 3.2.3, page 74, through Section 3.2.7, page 78. The `ov_import` command may be used to perform these operations simultaneously. Data is shared between steps of the definition process so that the administrator does not have to read and re-enter intermediate results.

The `ov_import` command accepts a series of command line options that specify a cartridge group name and partition bit format. Then it reads lines from standard input (that is, from the keyboard, pipe, or input file) containing all the information needed to create complete cartridge and volume definitions. The format of the input lines is:

```
PCL CARTRIDGETYPE [VOLUMENAME APPNAME]
```

The volume name and application name are optional, but if used, both must be used together and in order.

In Example 3-8, four cartridges are created with the `ov_import` command:

Example 3-8 `ov_import` Cartridge Creation

```
# ov_import -g default -b DLT7000
test001 DLT7000
test002 DLT7000
test003 DLT7000 vol1 ov_umsh
test004 DLT7000 vol2 ov_umsh
<EOF>
```

This `ov_import` command creates the cartridge data, a side and a partition on each cartridge with default names, and sets ownership and allocates a volume for the lines where volume name and owner were specified.

Example 3-9 shows the results with the `ov_lscarts` command:

Example 3-9 ov_lscarts Volume Listing

```
# ov_lscarts -lig 'test.*'
PCL      cart type  owner     group     state   cartID            part     volume
test001  DLT7000              default   ok      wIS2MTXt0jIABa6O  Part 1
test002  DLT7000              default   ok      wIS2MTXt0jcACesb  Part 1
test003  DLT7000    ov_umsh   default   ok      wIS2MTXt0koADWwM  Part 1   vol1
test004  DLT7000    ov_umsh   default   ok      wIS2MTXt0lEAC3I8  Part 1   vol2
```

This output shows that four cartridges now exist with "test" as the prefix of their PCL, that they all are DLT7000 cartridges, and that two of them have volumes and are owned by the application ov_umsh.

The ov_import command is most useful in defining cartridge information for a large number of cartridges at one time, such as when initially configuring an OpenVault system, or when adding a large number of cartridges to a library. In such cases, the ov_stat command can be used to get a list of PCLs of unknown cartridges in a library, and that list can then be edited into input for an ov_import command. This method minimizes the chances for human error in mistyping PCLs when creating cartridges, or entering incorrect CartridgeID strings for ov_part and ov_vol commands after a cartridge is created with ov_cart.

3.2.12 Removing Cartridges from Libraries

Administrators commonly remove cartridges for one of several reasons. These include disposal of cartridges after they have reached the end of their useful lives, moving cartridges to another library, or sending them to another site. As for the case of inserting cartridges in libraries, they can be removed from a library in two ways:

- First, you can simply open the door of a library and remove cartridges. When you close the door, the LCP scans the library and reports changes to the OpenVault server.
- The second method is to use the ov_eject command to tell the library which cartridges to move to the library's output port. A cartridge can be ejected by its PCL, CartridgeID, or position in a library, as follows:

  ```
  ov_eject -B PCL
  ov_eject -C wIS2MTXrMmQAAlYR
  ov_eject -l 9730 -b 'bay 1' -s 'slot 11'
  ```

If a cartridge is unknown to OpenVault (that is, present in a library but OpenVault has no data on it), then it must be specified either by PCL or by library slot position. If the PCL is missing or unreadable, then only the library slot form can be used.

3.2.13 Purging Cartridge Data

When a cartridge has been removed from an OpenVault system and is never expected to return (such as in the case of a cartridge that has reached the end of its useful life), there may be no further need to retain data about that cartridge. The data can be deleted from the OpenVault system with the `ov_purge` command. This command deletes all cartridge, side, and partition information about a cartridge as shown in Example 3-10:

Example 3-10 `ov_purge` Cartridge Data

A cartridge cannot be purged if there are allocated volumes on it. This is a safety precaution to ensure that data on cartridges that are still in use are not lost. The `ov_purge` command also asks for confirmation before it attempts to delete data as an additional check to help prevent inadvertent loss of data.

```
# ov_purge -C wIS2MTXrMmQAAlYR
Are you sure you want to purge all information for cartridge with
cartridge ID = wIS2MTXrMmQAAlYR (Y/N)? y
Deleted partition Part 1
Deleted cartridge wIS2MTXrMmQAAlYR
```

After this command is executed, no data remains in the OpenVault system about the cartridge, which returns to "unknown" state. If the cartridge is at the end of its useful life, the administrator should remove the cartridge from the library and destroy it, thus completing the life cycle for this cartridge.

Administering OpenVault

This chapter describes how to set up and configure OpenVault, as well how to perform some basic management tasks. The sections in this chapter are:

- Section 4.1 describes configuration files.
- Section 4.2, page 89, describes how to administer OpenVault.
- Section 4.3, page 97, describes how to monitor OpenVault.

4.1 OpenVault Configuration Files

OpenVault uses configuration files to initiate a communication session between device components on OpenVault clients and server. Configuration files are automatically created by the OpenVault `ov_admin` script, and contain information to bootstrap pieces of the system into full functionality. Once the various components have booted, the OpenVault server controls and manages clients and their components.

The following list defines terms used for OpenVault setup and administration:

Default access path

: Path used to access the drive via normal I/O channels, under `/dev/rmt`.

Drive handles

: Directory for creating `attach` nodes for applications, often `/tmp/mlm`.

Instance name

: You assign an arbitrary name to each drive and library. The instance name distinguishes between DCPs or LCPs controlling the same drive or library from different hosts (hopefully not simultaneously).

Passthrough driver

: Path used to access a device for direct SCSI control, under `/dev/scsi`.

Security key

A password for authorizing client or application access to OpenVault.

Port number

TCP port for OpenVault communication services, usually 44444.

Throttling

After a certain number of clients are connected, the connection rate is reduced to minimize contention.

OpenVault configuration files are described in detail in the following sections.

4.1.1 Server Configuration

When first configuring OpenVault, login to the OpenVault server system as superuser and run `ov_admin`, with its menu-based interface that helps guide you. For more information about setup procedures, see Chapter 2, page 13.

The OpenVault server configuration is stored in the `/var/opt/openvault/server/config/config` file.

4.1.2 Clients Configuration

OpenVault client software must run on all systems where OpenVault-managed drives and libraries are attached, in particular on systems not designated as the OpenVault server. These non-server systems are called OpenVault clients.

If you want to implement security, an authorization key must be created, as described in Section 4.1.4, page 87. Client security keys are part of their library, drive, and application configuration files. They are not in a separate file by themselves.

4.1.2.1 Setting Up Drives

Each drive needs its own drive control program (DCP) and configuration description, located in the `/var/opt/openvault/client/dcp/*/*/config` file. A drive-specific DCP should be installed using the standard OpenVault installation procedure.

For DCPs that are provided by SGI, DCP installation produces an interactive `ov_admin` script that requests the configuration information that is necessary to

create the configuration file. For DCPs supplied by other venders, consult their documentation.

The `ov_admin` script asks similar questions as for configuration of the OpenVault server, but limited to information needed for drive and DCP configuration.

4.1.2.2 Setting Up Libraries

Each library needs its own library control program (LCP) and configuration description, located in the `/var/opt/openvault/clients/lcp/*/*/config` file. A library-specific LCP should be installed using the standard OpenVault installation procedure.

For LCPs that are supplied by SGI, LCP installation produces an interactive `ov_admin` script that requests the configuration information that is necessary to create the configuration file. For LCPs supplied by other venders, consult their documentation.

The `ov_admin` script asks similar questions as for configuration of the OpenVault server, but limited to information needed for library and LCP configuration.

4.1.3 Applications Configuration

The `ov_admin` script asks you questions for configuring and storing the hostname of the OpenVault server used for the TCP interface. If you want to implement security, the application's OpenVault password must also be given. For details, see Section 4.2.5, page 91.

4.1.4 Setting Up Security

An ASCII authorization key file must be created on each AAPI/CAPI client and server system, and the client must be configured to use Open Vault with each key file. The authorization key file uses a *shared secret*—the client and server share a secret password. The server has one key file (`/var/opt/openvault/server/config/core_keys`) containing the passwords of each of the clients it knows about (drives, libraries, and applications). The clients each have an authorization key file listing the servers it knows about.

Example 4-1 shows a key file that can be used on either a client or a server.

Example 4-1 Client and Server Key Authorization File

```
#Host          Client        Instance     Language      Key
moon           System        OnlyInstance AAPI          a4sum
moon           Robbie        one          ALI           a2by4
moon           Herbie        alpha        ADI           gr8day
earth          networker     *            CAPI          un2
```

The meanings for each column in the key authorization file are described in Table 4-1.

Table 4-1 Key Authorization File Description

Column Title	Description
Host	IP hostname of the system that communicates with this system. If this is a CAPI client, give the OpenVault server hostname. If this is the OpenVault server, give the IP hostname of the client running the application.
Client	Name of the application.
Language	The language used for the connection, either AAPI, CAPI, ADI, or ALI.
Key	A password used to secure the connection, or “none” for no security.

On the server host, the shared secret key is stored in `/var/opt/openvault/server/config/core_keys`.

4.1.5 Setting Up Non-Robotic Libraries

Note: Non-robotic libraries are not supported in the OpenVault releases 1.*x*.

Non-robotic libraries are controlled and manipulated by a human operator rather than by a robotics interface. Organization of your cartridges is key to keeping your library accessible and usable. OpenVault can help by tracking the location of cartridges within a library. Assigning cartridges to a *bay* can further pinpoint cartridge location.

A bay is a storage location within the library. In the case of robotic libraries, bays can be a range of slots, a removable tray, or a storage silo. In the case of non-robotic libraries, a bay could be a bookshelf, a file drawer, or a storage closet. In these examples, a bookcase, file cabinet, or hallway is the library, partitioned into bays as you prefer.

Planning is important in determining how many bays and how many libraries are necessary for your storage needs. Be sure to plan adequate space to allow your library to grow and allow for further separation, based on detail (you may start with a "project," but then add more bays for phase 1, phase 2, and so forth).

Another major difference with a manual library is the need for communications with the operator. OpenVault needs to send messages to the operator, via a terminal, to prompt for action, such as loading a cartridge or removing a cartridge. Be sure to put the terminal that the operator uses in a convenient location so that timely communication is possible.

4.1.6 Setting Up Offsite Libraries

Note: Offsite libraries are not supported in the OpenVault releases 1.*x*.

An offsite library is a special type of manual library, differing in that an offsite library has no drive. Storage bins (which can be referred to as bays) contain the cartridges. OpenVault tracks the location of offsite storage cartridges. If a cartridge is needed from an offsite storage area, it must manually be located and moved into another library (see Section 5.2.2, page 107). The chosen library must contain a drive that is compatible with the cartridge type.

4.2 Administering OpenVault

Once OpenVault has been set up, as described in Section 4.1, page 85, you can start to configure it to meet your storage management needs. This section describes some basic configuration tasks, such as:

- Section 4.2.1setting logging levels.
- Section 4.2.3, page 91, registering applications.
- Section 4.2.4, page 91, unregistering applications.

- Section 4.2.5, page 91, setting up application security.
- Section 4.2.6, page 92, enabling application access to a drive.
- Section 4.2.7, page 92, managing cartridge groups.
- Section 4.2.7.2, page 95, introducing cartridges.

Tip: When using the OpenVault administrative commands, you can assign environment variable `OVSERVER` to name a default server. For example, if you add this line to your `.login` file, administrative commands use the OpenVault server *hostname*.

`setenv OVSERVER` *hostname*

If you want to use a different server, use the `-S` command-line option to specify another server (`-S` *newOVserver*). This overrides the `OVSERVER` environment variable.

4.2.1 Setting Logging Levels

You can configure the degree of system status reporting and message logging that you want to have. By default, system message logging sends all error messages to the file `/var/opt/openvault/server/logs/OVLOG.`*yyyymmdd*. Example 4-2 shows how to set the "information" log level.

Example 4-2 `information` Log Level

To set logging level to the information level:

```
ov_msg -s -t core -m information
```

See the `ov_msg`(1M) man page for details and Section A.2, page 125, for message log information.

4.2.2 OpenVault Timestamps

Internally OpenVault keeps time in UCT (universal coordinated time, also called GMT) and writes UCT timestamps into the message log file. The administrative commands translate UCT into the client's local time zone when displaying time values.

4.2.3 Registering Applications

Applications are client programs that can read data from or write data to (or both) removable media after OpenVault has found and mounted the desired media element. Reads and writes are performed using POSIX standard I/O facilities.

Applications can be added to (registered with) the OpenVault system at any time without the need to take OpenVault offline or perform a software upgrade. Registering an application introduces it to the OpenVault system so resources can be allocated for its use.

An authorization key, or password, can be added when registering an application to ensure secure transactions; see Section 4.2.5, page 91. Unless an application is registered, it cannot connect to the OpenVault server. To register an application, use the **Manage Applications** option from the `ov_admin` **OpenVault Configuration Menu**.

4.2.4 Unregistering Applications

Unregistering an application from the OpenVault system means the application can no longer connect to OpenVault. To unregister an application, use the **Manage Applications** option from the `ov_admin` **OpenVault Configuration Menu**.

4.2.5 Setting Up Application Security

Application security setup is similar to client security setup. Setting up an application with an authorization key ensures that, without the security key, no other application can connect to OpenVault masquerading as the originally authorized application.

As shown in the first and last lines of Example 4-1, page 88, the application name is used as a Client for the OpenVault server, which is named as the Host. The first line is an example that secures the OpenVault administrative tools and the last line is an example that secures the IRIX Networker application. The middle two lines appear only in the server keyfile and describe options for a library named `moon` and a drive named "moon".

The key that is assigned in the key authorization file, also known as the password, must already exist in the configuration file for the application. See Section 4.1.3, page 87, for details on setting up his file.

4.2.6 Enabling Application Access to a Drive

A drive is in exactly one drive group. Each application can be given access to one or more drive groups (all drives in a group). The `ov_admin` command can be used to enable a drive for a specific application, allowing the application to run on the drive. The **Manage Drive Groups** option of the `ov_admin` command allows the user to create a drive group, reassign a drive to another group, grant an application access to a drive group, and perform other related functions.

When using `ov_admin` to grant an application access to a drive group, one of the parameters you are asked to enter is the priority for the application's use of the drive group. OpenVault can be configured so an application will prefer to use drives in certain drive groups over drives in other groups. A priority is given to the application's use of each drive group and those priorities are used to sort the available drives into preference order. A higher priority number means that the application will prefer that drive group, while a lower number means that the application will prefer not to use that drive group.

4.2.7 Managing Cartridge Groups

Cartridge groups allow sets of cartridges to be allocated for specific applications or purposes. You may want to do this, for example, based on the capacity of the cartridge or its content (for example, the existing data on the cartridge is related to other cartridges in the group). Creating cartridge groups helps you organize your media and fulfills the demands of the application for available storage.

Figure 4-1 shows an environment with several cartridge groups.

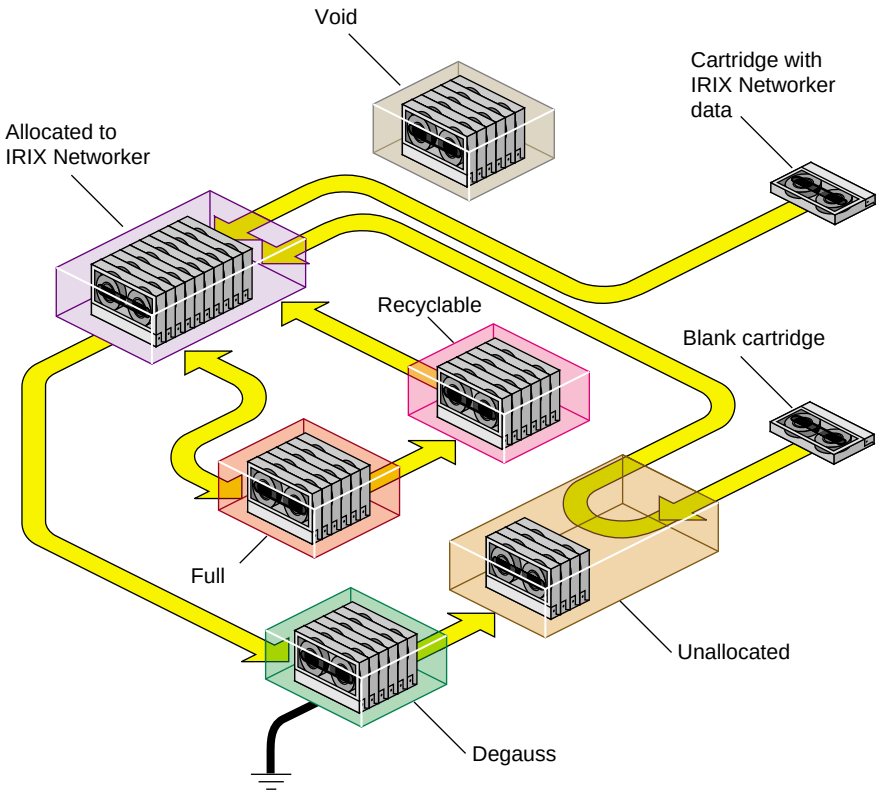

Figure 4-1 Example Cartridge Group (Engineering)

Cartridges move in and out of groups (shown by the arrows) based upon demands made by the application or by their place in the media life cycle (see Section 7.1.1.2, page 115). The cartridge groups in the figure are Void, Unallocated, and Degauss.

Void — A set of cartridges whose labels are not in the OpenVault library. This is a holding place until you move the cartridge to another group, such as Scratch, or find out why the catalog entry for the cartridge is missing.

Unallocated — A set of unallocated cartridges that are available for allocation by any application. These cartridges have been introduced into the OpenVault system and are ready to be allocated to some application.

Degauss — A set of cartridges that are ready to be demagnetized or destroyed. Cartridges are demagnetized to ensure all data is destroyed. Some of

these cartridges may need to be terminated if they are approaching the end of their media life span.

Full — IRIX NetWorker manages this set of cartridges. When a volume fills up, NetWorker marks it as full and holds it for a specified retention period.

Recyclable — IRIX NetWorker manages this set of cartridges, too. When the retention period expires and its cycle ends, a volume is marked for recycling.

4.2.7.1 Setting Up Cartridge Groups

Before setting up a cartridge group, plan where the group will reside and which drives or applications need to access it. Does the cartridge group require high throughput or high availability? Consider the network environment and any limitations it may impose when deciding where to locate the group. Consider the drives that will be servicing the group. What advantages or limitations do they impose? Also consider the growth potential for your storage requirements and choose a library that is adequate for its needs both now and for its projected future.

To create a cartridge group, follow the steps in Procedure 4-1.

Procedure 4-1 Creating a Cartridge Group

1. Be sure you have registered the application that will manage this group. See Section 4.2.3, page 91, for details.
2. Determine whether to assign drives to the application. If you decide to do so, see Section 4.2.6, page 92, for details.
3. Create the cartridge group. The **Manage Cartridge Groups** menu item of the `ov_admin` command is the preferred way to create a cartridge group. The **Manage Cartridge Groups** menu option can also be used to grant an application access to the cartridge group.
4. Introduce cartridges into the cartridge group. Continue with the next section, Section 4.2.7.2, page 95, for details.
5. If necessary, move cartridges between groups or within the library. The following sections describes situtions when you may need to move cartridges:
   - If a cartridge is no longer needed by an application, see Section 5.2.1, page 106.
   - If a cartridge can be moved back into the Unallocated group, see Section 5.2.3.1, page 108.

- If a cartridge should go to the Degauss group for erasure, see Section 5.2.3.2, page 108.
- If a cartridge needs to be moved within its own group, see Section 5.2.2, page 107.

4.2.7.2 Introducing Cartridges

Cartridges must be introduced into the OpenVault system before they can be used by an application. When a cartridge is introduced, an entry in the OpenVault catalog is created for the cartridge and the cartridge is assigned a cartridge ID. The entry, or cartridge ID, tracks information such as the PCL or bar code, application ID (if it's associated with an application), number of reads and writes, form factor, capacity, and so forth.

Client applications may choose their own names for cartridges. Because OpenVault client applications operate in separate name spaces, different applications may use the same name for different cartridges. Moreover, cartridges used by one application are not visible to or accessible from another application, unless the system administrator permits specific cartridges to be moved from one application to another.

Cartridges can be introduced in two ways, as shown in Figure 4-1, page 93:

- Cartridges without data (including data that can be overwritten)
- Cartridges containing data that is needed by an application (partition needed)

To introduce a cartridge without data, follow the steps in Procedure 4-2:

Procedure 4-2 Introducing a Cartridge without Data

1. To facilitate the cartridge identification, create its PCL by attaching a bar code or a label to the outside of the cartridge. The PCL is tracked in the catalog entry for this cartridge and is used as the default identification for the cartridge in most commands.
2. Signal the library to open its inject port so you can insert the cartridge. You must specify the library name (`-l` *libraryName*).

   `ov_inject -l` *libraryName*

3. Insert the cartridge into the inject port.
4. Create a new (`ov_cart -n -T`) cartridge entry in the OpenVault catalog so the cartridge can be recognized and accessed. The `-B` *label* gives the PCL affixed to

the cartridge. Also, assign the cartridge to a cartridge group (`-g` *cartridgeGroup*) so it can be used by an application needing a new cartridge:

`ov_cart -n -B` *label* **`-g`** *cartridgeGroup* **`-T`** *cartridgeType*

OpenVault checks its catalog to ensure the PCL is unique (new) and rejects the entry if that PCL already exists. Then `ov_cart` automatically creates sides on the cartridge, depending on the *cartridgeType* argument. The group name, form factor, media type, PCL and other identifying information are recorded in the OpenVault catalog for this cartridge.

Note: This command (`ov_cart -n -T`) is a good way to enter each new cartridge into the OpenVault system. Bulk insertion of cartridges into OpenVault may also be done using the `ov_import` command.

---

5. Create a partition on side 1 of the cartridge.

   `ov_part -n -C` *cartridgeID* **`-p`** *partitionName* **`-s SideA`**

To introduce a cartridge that already contains data, follow the steps in Procedure 4-3:

Procedure 4-3 Introducing a Cartridge with Data

1. Follow Procedure 4-2, page 95, for introducing cartridges without data, including step 5.
2. Create volumes for this cartridge. A volume allows the cartridge to be segmented so separate areas of the media can be used by an application:

   `ov_vol -n -v` *volumeName* **`-a`** *applicationName* **`-C`** *cartridgeID* **`-p "PART 1" -s SideA`**

   Use this command for each volume you are creating. OpenVault checks that the *applicationName* and *volumeName* combination is unique. Standard attributes are assigned to the volume and given default values if the values are not specified.

Note: This `ov_import` command optionally allows you to automatically create volume records for cartridges with data, in addition to creating the cartridge, side, and partition information for OpenVault.

You have now successfully introduced a cartridge and associated it with an application. Repeat this procedure for each new cartridge you are adding to the

OpenVault system. You can now perform operations on the cartridge, as described in Chapter 5, page 103.

4.3 Monitoring OpenVault

Monitoring OpenVault helps you spot potential problems so that you can take appropriate action, and allows you to track the OpenVault system usage. If reconfiguration is necessary, it can be done before it becomes critical.

Probably most helpful in daily operations, monitoring can help you track the data you have on your media and where the media is located. The following subsections describe these monitoring tasks:

- Section 4.3.1, page 98, describes checking server status.
- Section 4.3.2, page 100, describes checking media inventory.
- Section 4.3.3, page 101, describes how to list cartridge information.

OpenVault can be monitored on several levels and to varying degrees of detail. From the general to the specific, some of the areas that can be monitored are:

- OpenVault server status, including whether the server is up or down and by listing system error messages
- Library status, including names, types, whether the library is up or down, and slotmaps (showing occupancy and reservations)
- Drive status, including names, types, whether the drive is up or down, and cartridge occupancy
- Libraries and drives without active LCPs and DCPs
- Cartridge group status, including names of groups, number and names of cartridges in them
- Names of registered applications
- Cartridge status, including listing of all known cartridges, cartridge locations, and cartridge characteristics
- Volume status, including cartridge ID and PCL, and the owning application
- System messages, including by library, drive, and type (warning, error, operator)

Check the OpenVault man pages for a complete description of available commands and options (see Appendix B, page 131, for a complete listing of man pages).

4.3.1 Checking Server Status and Configuration

Using the `ov_stat` command to check the status of the OpenVault server allows you to view your system from a top-level viewpoint. Some things to check for are:

- Check that the default server is up:

  `ov_stat`

- Check that a specific server is up:

  `ov_stat -S` *serverName*

- Show the task queue for the default server:

  `ov_stat -q`

- Display status of all items for the server, LCPs, DCPs, and applications:

  `ov_stat -a`

- Show configuration (`-c`) and drive (`-d`) mode information for a drive:

  `ov_stat -D` *driveName* **`-d -c`**

```
Drive    Mode     Slot Type   Cart Type   Bit Format  Capacity  ...
...
```

- Show library (`-l`), configuration (`-c`), and slot map (`-s`) information for a library:

  `ov_stat -L` *libName* **`-l -c -s`**

```
Library Name  Broken  Disabled  State  LCP SoftState  LCP HardState
lib1          false   false     ready  inactive       ready
...
```

Note:
- A drive or library cannot spontaneously become Disabled. The system administrator disables a drive or library; it does not happen automatically.
- If a library or drive fails an internal hardware diagnostic, its LCP or DCP can report the device as Broken. No currently shipping LCP or DCP does this, however.
- The Access heading indicates whether the library believes it can mount and unmount cartridges in a particular drive. It is possible for a drive to be in perfect running order, but nonetheless unreachable by the library.

OpenVault tracks up to 8 status items for drives and libraries, as shown in Table 4-2.

Table 4-2 `ov_stat` Headings Explained

	Software	Hardware
LCP	State of the LCP: "ready", "not ready", "disconnected", "inactive", or "active".	Always set to "ready".
DCP	State of the DCP: "ready", "not ready", "disconnected", "inactive", or "active".	DCP's view of the drive: "loaded", "unloading", "unloaded", or "loading".
library	Always set to "ready".	Always set to "ready".
drive	Whether the drive being used by an application: "inuse" or "ready".	Drive state: "loaded", "unloaded", "loading", or "unloading".

The meaning of tokens shown in `ov_stat` output is as shown in Table 4-3:

Table 4-3 `ov_stat` Tokens Explained

Token	Meaning of Token
`ready`	Can accept commands, or is working properly, or is unallocated.
`not ready`	Temporarily unable to accept commands (the door might be open).
`disconnected`	The LCP or DCP is communicating with the MLM, but not with the hardware.
`inactive`	The LCP or DCP is not communicating with the MLM.
`inuse`	A drive has been allocated for use by an application.
`unloaded`	There is no cartridge in the drive.
`loading`	A cartridge is being loaded into the drive.
`loaded`	A cartridge is loaded in the drive.
`unloading`	A cartridge is being unloaded from the drive.
`active`	The LCP or DCP is becoming ready, but is not yet able to take commands.

4.3.2 Checking Media Inventory

Knowing your media inventory is essential for a well-organized site. OpenVault helps track the contents of libraries and cartridge groups so you know the whereabouts of your cartridges, including those that are currently loaded into drives.

The following `ov_lscarts` commands can be used to check the media inventory:

- Show location information for all cartridges, such as the library location, slot assignment, presence in slot, or if loaded in a drive, the name of that drive:

  ```
  ov_lscarts -w '.*'
  ```

- Show cartridges assigned to the IRIX Networker application, in long format:

  ov_lscarts -A networker -l '.*'

4.3.3 Listing Cartridge Information

Other information that is important to a smooth-running site is information regarding cartridge contents and usage, such as the number of reads, writes, mounts, and so forth.

The following `ov_lscarts` commands can be used to check information about cartridges:

- Show a summary of all cartridge information on the default server:

  ov_lscarts '.*'

- Show the application and cartridge group associated with each cartridge:

  ov_lscarts -g -a '.*'

4.3.4 Backing Up OpenVault

If your site performs system-wide backups, then the OpenVault database files will be included in these backups. If the OpenVault server is running at the time of the backup, the catalog files may not get backed up properly. These files can change in the middle of a backup session, especially if your backup scheme employs OpenVault.

To ensure backup of the OpenVault catalog files, follow the steps in Procedure 4-4:

Procedure 4-4 Backing Up OpenVault

1. Shut down OpenVault.
2. Copy the `/var/opt/openvault/server/dbase` directory to another location, using care to preserve the hard links in that directory. One way to do this is as follows:

   # **cd /var/opt/openvault/server**
   # **tar cvf - dbase | (cd *savelocationpath* ; tar xvf -)**

   savelocationpath is the name of the directory where the copies will be saved.
3. Restart OpenVault.

4. Perform the normal backup procedure, making sure to back up the copy you made of the `/var/opt/openvault/server/dbase` directory.

Of course, any single-user mode level-0 dumps, which you should be doing, will also backup the OpenVault catalog files.

To recover the catalog files, just restore them from tape to the temporary location, then copy them back into `/var/opt/openvault/server/dbase`, making sure that you preserve the hard links.

Operating OpenVault

This chapter describes the OpenVault commands that are used to perform operator tasks. The tasks are divided into normal, everyday operations and those that are performed occasionally, mainly for organizational reasons. The sections in this chapter describe the following tasks:

- Section 5.1, describes daily tasks.
- Section 5.2, page 106, describes occasional tasks.

5.1 Performing Daily Tasks

This section describes the most common operator storage management tasks. Mainly, these tasks revolve around manipulating cartridges and managing devices (libraries and drives).

5.1.1 Manipulating Cartridges

The operator manipulates cartridges by performing the operations as described in the following subsections.

5.1.1.1 Mounting Cartridges

A cartridge is mounted in a drive to allow a read or write operation to be performed by an application. When a cartridge is mounted, as performed with the default `ov_mount` command, it is placed into an available drive that is compatible with the cartridge's format. A device name is returned so the cartridge can be manipulated by an application. Once the cartridge is mounted into the drive, a shell process is started that allows an application to manipulate the drive. After the application finishes its operation, the shell process exits, and the cartridge is automatically unmounted.

The following are examples of the `mount` command options:

- A cartridge using its cartridge ID:

  `ov_mount -C -I` *sidename cartridgeID*

- A particular drive by its drive name, as recorded in the client configuration file (see Section 4.1.2.1, page 86):

  ov_mount -d *driveName* **-C** *cartridgeID* **-I** *sidename*

- A volume owned by an application:

  ov_mount -V *volName* **-A** *application*

Note: Occasionally, you may have to use ov_unmount to free a cartridge that is still in a drive because the application that mounted it has died. The unmount command uses all the options as the mount command, as described in Section 5.1.1.1, page 103.

5.1.1.2 Checking the Task Request Queue

When an OpenVault task request is initiated (for example, by a client application or the OpenVault ov_mount command), a task request (with its own ID) is generated and placed in the OpenVault server task queue. To check the pending task requests, use:

ov_stat -q

5.1.1.3 Canceling a Pending Task Request

Note: The ov_cancel command is not supported in the OpenVault releases 1.*x*.

If you want to cancel a pending task request (for example, the desired cartridge is not available), use the cancel command (ov_cancel *taskID)*, specifying the task by its ID number (as obtained with the ov_cancel command).

With the ov_cancel command, you can also send an explanation to the task that originated the task request, explaining why the operation is being canceled:

ov_cancel -r "*reason for cancellation*"

Tip: If the string contains spaces, enclose it in quotation marks.

5.1.2 Managing Devices

Sometimes the operator needs to step in and manually control a device to perform a task (such as taking a drive offline so it can be cleaned). This section describes some tasks the operator can perform to manage devices.

A device (a library or drive) is specified by its device name, which is the name used in the configuration file (see Section 4.1.2.1, page 86, and Section 4.1.2.2, page 87).

5.1.2.1 Disabling and Enabling Devices

You can temporarily disable a device to take it offline for a period of time in case servicing is necessary, such as when a drive needs cleaning. Permanently disabling a device may be necessary if it is malfunctioning and repairs need to be performed.

When the device is disabled, either temporarily or permanently, all communication to it is stopped and the OpenVault system considers it unavailable.

The following commands disable and then enable the device:

- To temporarily disable a drive and a library:

  ov_drive -T *name*
  ov_library -T *name*

- To permanently disable a drive and a library:

  ov_drive -D *name*
  ov_library -D *name*

- To enable a drive and a library:

  ov_drive -E *name*
  ov_library -E *name*

5.1.2.2 Cleaning a Drive

Note: Cleaning facilities are not supported in OpenVault release 1.*x*, and the `ov_clean` command is not included.

OpenVault helps monitor a drive's cleaning schedule by tracking the dates a drive has been cleaned and the number of read/write errors occurring on the drive. The `ov_clean` command will help find and load a cleaning cartridge, if one is available.

Drives are specified by the name recorded in the client configuration file (see Section 4.1.2.1, page 86):

- To display the cleaning information for a drive:

  ov_clean -i -d *driveName*

- To check whether a cleaning cartridge is available and which cartridge will be used if a cleaning cartridge is not specified:

  ov_clean -n -d *driveName*

- To perform the cleaning operation, using the cleaning cartridge OpenVault selects:

  ov_clean -d *driveName*

- To perform cleaning, using a particular cleaning cartridge, specified by PCL:

  ov_clean -d *driveName PCL*

5.2 Performing Occasional Tasks

This section describes the tasks that you probably perform occasionally, generally to provide organization for your cartridge storage and keep the OpenVault catalog synchronized with the movement of your cartridges and libraries. The OpenVault catalog tracks the location and status of all cartridges it is aware of.

5.2.1 Removing Cartridges from Libraries

When you want to remove a cartridge from a library (also known as ejecting), use the `ov_eject` command. The OpenVault catalog entry for that cartridge is not removed and all record of the cartridge is saved. The cartridge is either ejected (if the library can perform an eject) or an operator message is sent to the console, indicating to the operator that the cartridge can be physically removed. After a cartridge is ejected, it must be injected again (using `ov_inject`) before it can be used.

The following options can be used when ejecting a cartridge from a library:

- Eject a cartridge using its PCL (default):

  ov_eject -B*PCL*

- Eject a cartridge using its location (library, slot, and bay, if bays are present):

ov_eject -l *libraryName* **-s** *slotNumber* **-b** *bayName*

5.2.2 Moving Cartridges within Libraries

Note: Cartridge move is not supported in OpenVault release 1.*x*; that is, the ov_move command is not supported. To move cartridges within a library or between libraries, first eject them and then inject them into the library. See the ov_eject(1M) and ov_inject(1M) man pages for more information.

Occasionally you may want to reorganize cartridges in a library. For example, you may decide to group cartridges by application or to move them closer to their assigned drive.

Before moving any cartridges, it is helpful to generate a listing showing the contents of slots in the library. This should help you determine which slot addresses are immediately available for use and which cartridges may have to be moved to obtain their slot. The format of the slot address is specific to the LCP in use. To obtain a library slotmap:

ov_stat -s -L *library*

When using the ov_move command, you can specify cartridges by their PCL or by their location in the library.

- Move a cartridge using its PCL:

  ov_move *PCL destinationSlotAddress*

- Use the library location (including slot ID) instead of the PCL:

  ov_move -L *libraryName originalSlotAddress destinationSlotAddress*

5.2.3 Maintaining the Server Catalog

The OpenVault server catalog tracks the location and details of each cartridge known to the OpenVault system and the up or down status of the drives and libraries. Occasionally, you may need to step in and make some corrections to the catalog to update it.

5.2.3.1 Recycling Cartridges

When you want recycle a cartridge in OpenVault system for use by another application, use the `ov_recycle` command. All information relative to cartridge use (such as number of reads and writes) remains in the OpenVault catalog.

- Recycle the cartridge:

  `ov_recycle -r -B` *PCL*

- Recycle the cartridge, using its cartridge ID:

  `ov_recycle -r -C` *cartID*

- Recycle all cartridges owned by the given application:

  `ov_recycle -r -A` *application*

5.2.3.2 Destroying Cartridges

When you want to completely remove a cartridge from the OpenVault system, for example to destroy a cartridge when it is generating I/O errors at the end of its life cycle, use the `ov_purge` command, which erases the cartridge entry from the OpenVault catalog.

- Remove the cartridge entry, using its cartridge ID:

  `ov_purge -C` *cartID*

- Suppress the interactive verification; immediately remove the cartridge entry:

  `ov_purge -f -C` *cartID*

Note: Use the `ov_purge` command sparingly, because important usage information is lost forever.

Reconfiguring OpenVault

This chapter describes OpenVault methods for reconfiguration and performance tuning. The sections in this chapter include:

- Importing media into different cartridge groups
- Adding (configuring) or deleting (deconfiguring) drives
- Changing the drive group of a drive
- Changing the name of a library
- Adding remote drives, libraries, and applications at a later time
- Establishing OpenVault security after setup with “no security”
- Changing the OpenVault password for applications, libraries, and drives

6.1 Importing Media Into Cartridge Groups

To import media into different cartridge groups, use the `ov_import` command, perhaps automated with input scripts, to import cartridges into cartridge groups specified by the `-g` option. For example, to import four tapes into the DMF cartridge group, and two other tapes into the NetWorker cartridge group, run these commands:

```
# ov_import -g dmf -b DLT7000
test001 DLT7000 vol1 dmf
test002 DLT7000 vol2 dmf
test003 DLT7000 vol3 dmf
test004 DLT7000 vol4 dmf
Ctrl+D
# ov_import -g networker -b DLT7000
test0A DLT7000 volA networker
test0B DLT7000 volB networker
Ctrl+D
```

For more information, see Section 3.2.11, page 81.

6.2 Adding or Deleting Drives

When you add a drive to your system, OpenVault must recognize the drive in order to put it under management. The preferred method to do this is with the `ov_admin` command, via the **Manage DCPs for Locally Attached Drives** option. You can use this menu option to create an OpenVault record of (and DCP for) the drive. Alternatively, the `ov_drive` command provides options to create a drive.

To remove a drive from OpenVault management, use the **Manage DCPs for Locally Attached Drives** option from the `ov_admin` command. Alternatively, the `ov_drive` command provides options to delete a drive.

6.3 Changing the Drive Group of a Drive

The preferred method to change the drive group of a drive is with the `ov_admin` command. The **Manage Drive Groups** menu selection causes a submenu to display. The item **Reassign a drive to Another Drive Group** allows you to change the drive group of a drive.

6.4 Changing the Name of a Library

To change the name of a library, you must first delete that library (and its associated LCP), then create it anew with a different name. The preferred way to do this is with the `ov_admin` command. The **Manage LCPs for Locally Attached Libraries** option allows you to delete an LCP. After that step is accomplished, you can use the **Manage LCPs for Locally Attached Libraries** option again to create the LCP with a new name.

6.5 Adding Remote OpenVault Components

To add remote OpenVault components, inform the server about them, then run the `ov_admin` script on the remote client, as documented in Chapter 2, page 13.

6.6 Establishing OpenVault Security

When you initially configured OpenVault, you probably followed instructions in the documentation and created an installation without security. This implies that the `/var/opt/openvault/clients/admin/keys` file, the

/var/opt/openvault/dcp/*/*/config file and /var/opt/openvault/clients/lcp/*/*/config files, and the var/opt/openvault/server/config/core_keys file, specify "none" as the security key.

Note: Perhaps this is obvious, but passwords for specific applications in clients/admin/keys, for DCPs in clients/dcp/*/*/config, or for LCPs in clients/lcp/*/*/config, must be the same as the password given in the server/config/core_keys file for that component.

To establish security, become superuser and edit these files, substituting the password of your choice for the word "none" on lines reading key:

```
# cd /var/opt/openvault
# vi clients/admin/keys clients/dcp/*/*/config clients/lcp/*/*/config\
        server/config/core_keys
~
/none
```

6.7 Changing OpenVault Passwords

OpenVault authorization is aided by passwords specified in the security files described in Section 6.6. These passwords can all be the same, or they can be different from one component to the next. To change passwords, become superuser and edit these files, substituting the password of your choice for old password, either globally, or component by component:

```
# cd /var/opt/openvault
# vi clients/admin/keys clients/dcp/*/*/config lcp/*/*/config \
        server/config/core_keys
~
/key:
```

6.8 Reconfiguring Server Operation

The /var/opt/openvault/server/config/config file contains crucial OpenVault operational parameters, expected to remain fairly static, as shown in Table 6-1. Modify them with utmost care.

Table 6-1 OpenVault Server Parameters

Parameter	What It Controls
PORTNUM	TCP/IP port number on which OpenVault listens for connections
SEMAKEY	Semaphore key that OpenVault uses for communication between components
MAXSTARTS	Maximum number of connections before OpenVault rejects new ones
BOOTGRACE	After reboot, number of seconds OpenVault waits before ejecting drives
TASKRETRY	Number of seconds OpenVault waits before reevaluating blocked mounts
MPRETRY	Number of seconds OpenVault waits before retrying a failed mount

Tertiary Storage Management

This chapter describes some strategies for optimizing storage management:

- Section 7.1 describes the mass storage options available today.
- Section 7.2, page 116, talks about cabling tertiary storage devices to server machines.
- Section 7.3, page 119, discusses tertiary storage software for backup, archive, and hierarchical storage management.

7.1 Tertiary Storage Devices

This section discusses the hardware currently available for tertiary storage, also called nearline storage. The hardware used for secondary storage is usually magnetic disk, which offers the advantages of permanence, rapid random access, and decreasing cost. Laser-activated protein storage may eventually provide even higher capacity and lower power consumption than magnetic disk. Primary storage usually refers to chip-based electrical memory such as cache or random access memory (RAM).

7.1.1 Tape Drives

Tape drives, because of their rewritability and low cost per unit of data stored, are now the preferred method for backing up data to protect against data loss.

7.1.1.1 Tape Usage

Tape cartridges cost from US$10 for a 2 GB DAT tape to almost US$100 for a 35 GB DLT tape. Cost per megabyte (now about 1¢ including amortized tape drive investment) has been declining as tape capacity increases while cartridge price remains about constant.

By comparison, storage on magnetic disk, which of course provides rapid and random access, now costs under 10¢ per megabyte and is declining more rapidly than tape cost.

The typical magnetic tape lasts about five years, and can be rewritten hundreds of times. Just before a tape fails, its soft error rates rise. OpenVault can transmit the soft

error rate as reported by hardware. After a tape fails, there is no good way to recover the data stored on it. OpenVault can monitor the total number of reads and writes to a tape; so you can arrange transfer of data to new media before the tape fails.

Table 7-1 shows the characteristics of several popular tape drives now on the market.

Table 7-1 High Capacity Tape Drives

Drive	Native Capacity	Data Rate	Cartridge Load Time	Tape Size	Expected MTBF	Power Needed	Typical Price
AIT-2	50 GB	6.0 MB/sec	TBD	8 mm	250,0000 hours	+5V +12V	TBD
DDS-3 (DAT)	12 GB	1.0 MB/sec	30 secs	4 mm	35,000 hours	6 w	$1500
DLT 4000	20 GB	1.5 MB/sec	45 secs	1/2 in.	80,000 hours	25 w	$3600
DLT 7000	35 GB	5.0 MB/sec	40 secs	1/2 in.	200,000 hours	37 w	$8500
DLT 8000	40 GB	6.0 MB/sec	37 secs	1/2 in.	TBD	140 w	TBD
EXABYTE Mammoth	20 GB	3.0 MB/sec	20 secs	8 mm	200,000 hours	15 w	$4200
IBM Magstar 3570	5 GB	7.0 MB/sec	16 secs	1/2 in.	3 year uptime	40 w	$8500
IBM Magstar 3590	10 GB	9.0 MB/sec	16 secs	1/2 in.	3 year uptime	60 w	$25000
LTO Ultrium	100 GB	15 MB/sec	-	1/2 in.	-	-	$4000
Sony AIT	25 GB	3.0 MB/sec	7 secs	8 mm	200,000 hours	12 w	$5200

7.1.1.2 Preventative Maintenance

Tape drives should be kept in low-dust environments with moderate temperature and humidity. They should be cleaned when the cleaning light comes on, or in the absence of a cleaning indicator, at regular intervals as recommended by the manufacturer. If a tape drive has a cleaning indicator, it is best to clean the drive only when indicated, in order to reduce wear on the tape heads.

Using high quality media helps preserve tape heads and can reduce cleaning intervals. Always discard cleaning cartridges before they reach the end of tape.

Pay close attention to drive alerts and media faults and act quickly to resolve problems. Monitor your tape drives daily, and read the error logs. Tape drives usually give out subtle indicators before failing. More frequent error correction code (ECC) messages often indicate impending drive failure. Perform read-write confidence tests at regular intervals, because testing can identify failing hardware before data loss occurs.

7.1.2 Optical Drives

For software and data distribution, and for archival purposes, optical drives have now surpassed magnetic tape.

In moderate quantities, compact disc (CD) can be manufactured for under 50¢ a copy. Writable compact disc recordable (CDR) media now cost under $3 each. (Rewritable CDWR media now cost almost $20 each.) CDR burners have been declining in price to well under $500 today. Although the CD format is limited to about 650 MB, the digital versatile disc (DVD) format offers backward compatibility with CD, and much higher data capacity, variously quoted between 4 GB and 17 GB.

Although the cost per megabyte is much higher for CD than for magnetic tape, optical data can last virtually forever. The problem is that data must be staged before writing to disc, making the CDR inconvenient as a backup device—tapes can stop and start.

7.1.3 SCSI Media Changers

The SCSI 2 standard specified a range of commands for interfacing with removable media libraries, also called autochangers or jukeboxes. Such devices contain one or more tape or optical drives, and robotic mechanisms to exchange cartridges between storage slots and a drive. SCSI media changers can cost anywhere from under $4000 to much more, depending on the number of drives and slots configured in the device.

There are many manufacturers of SCSI media changers, and robotic designs vary, but most media changers include one or more of the tape drives listed in Table 7-1, page 114.

When media changers are configured with multiple drives, they can also have multiple SCSI interfaces so that drives can operate concurrently. It is most useful to have two or more drives in a media changer, and as many storage slots as possible. Device control may be done over a serial line, or by means of the lowest numbered SCSI interface.

7.1.4 Silo Libraries

When data transfer speed and aggregate storage capacity are at a premium, datacenters often choose proprietary silo libraries from ADIC DAS, EMASS Grau, IBM, or STK. These silo devices contain high-speed robotic mechanisms and multiple tape drives, and operate under control of a dedicated computer interface, rather than under the SCSI standard. Each tape drive usually has a direct SCSI connection to a network server, however.

In OpenVault documentation, silo libraries and SCSI media changers are classified as removable media libraries.

7.2 Connecting to a Host Computer

This section offers guidelines for connecting drives and removable media libraries to host computers such as SGI servers.

7.2.1 SCSI Connection Guidelines

A lot has been written about maximum SCSI cable length. This is not usually an issue because SCSI cables are available only in approved lengths. For single-ended SCSI, the maximum cable length is 6 meters, and 3 meters is the recommended maximum.

Differential SCSI improves signal integrity; so data can be transmitted farther and faster than with single-ended connections. Differential technology doubles the number of signal wires, with each second wire carrying an inverted signal—because measuring the difference between two signals is more reliable than measuring a single binary signal. The recommended maximum cable length for differential SCSI is 25 meters.

In practice, any type of SCSI bus should be as short as possible with as few connections as required. The SCSI-2 standard allowed up to eight SCSI addresses, whereas SCSI-3 allows up to 16 addresses. On IRIX systems, address zero is reserved for the controller.

External devices must be terminated with an externally mounted SCSI port terminator on the rear of the drive. Terminators are not required on internally mounted drives, because internal termination is handled on the SCSI drive backplane.

Active terminators may be used to improve signal integrity on either single-ended or differential SCSI busses. Active terminators are usually battery powered and come with an LED to indicate that they are working. Some have an external power supply.

Sometimes you hear that you should cable high-speed devices closest to the SCSI controller, with low-speed devices further out. In practice, however, most SCSI busses are limited by the speed of the slowest communicating device, no matter what its position.

Table 7-2 compares the bandwidth of different SCSI types.

Table 7-2 SCSI Types and Speeds

SCSI Type	Data Word Size	Clock Speed	Bandwidth	Cable Length
narrow	8 bits	5 MHz	5 MB/sec	6 meters
wide	16 bits	5 MHz	10 MB/sec	6 meters
fast/narrow	8 bits	10 MHz	10 MB/sec	3 meters
fast/wide	16 bits	10 MHz	20 MB/sec	3 meters
Ultra fast/wide	16 bits	20 MHz	40 MB/sec	1.5 meter

SCSI-1, SCSI-2, and SCSI-3 refer to different protocols, with higher protocol numbers having additional commands and interfaces. People often mix up protocol with speed, bandwidth, and even connector type. In general, SCSI-1 is 5 MHz, SCSI-2 is 10 MHz, and SCSI-3 (or UltraSCSI) is 20 MHz.

Figure 7-1, Figure 7-2, and Figure 7-3 illustrate three types of connectors that are in wide use today:

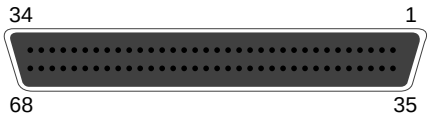

Figure 7-1 68-pin Wide SCSI-3 Connector

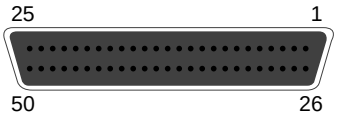

Figure 7-2 50-pin SCSI-2 Connector (Mini-micro)

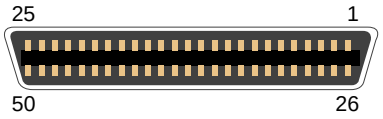

Figure 7-3 50-pin Centronics Parallel Connector

The first is the only type that can be used on fast and wide SCSI busses. The second and third types are functionally equivalent. The third type costs less than the second, but has a tendency to slow down the bus, and it should be used only with slow devices.

It is critical to put narrow devices at the end of a wide bus, with the wide bus terminated on the upper data lines and signals at the transition point. This results in fewer problems. SGI sells 68-pin to 50-pin (both mini-micro and Centronics) SCSI cables that have termination built in to the connector at the wide end.

Sustained bandwidth is typically no more than 80% of the peak bandwidth. It depends on the quality of disk drives and communicating drives. Transfer rates decrease when you put too many devices on the same SCSI bus.

For maximum bandwidth, it is best to place two fast devices on different SCSI controller. For example, if you have three DLT 7000 drives intended for an Origin2000, attach each one to a separate SCSI bus on the XIO card. This way there will be little bus contention, and the Origin2000 server will be able to drive them all near their rated throughput.

7.2.2 SGI Servers

OpenVault runs on SGI servers including the Origin 200 server, Origin 2000, and Origin 3000 server family. For server information, see the server hardware manuals.

7.3 Storage Management Applications

This section gives an overview of software currently used for tertiary storage.

7.3.1 Scheduled Backup

The principal purpose of backup is to provide fall-back data in case of disaster. As a side benefit, backup allows users to recover files that they delete accidentally. In practice, the side benefit occurs more often, but is less important in the overall scheme of things.

7.3.1.1 Full, Incremental, Differential, and Network Backups

In OpenVault database marketing literature, scheduled backup is often called *fully-automated lights-out* backup. This means that data backup occurs unattended, usually in the middle of the night when system load is low or nonexistent.

Suppose you have a server with 1 TB (terabyte, equal to 1000 GB) of data to safeguard. Dumping all the data to tape is called a *full* backup. Two DLT 7000 drives would take over 27 hours to copy this data, not including tape changing time (29 tapes are required). Fortunately not all data needs daily backup, because certain files seldom change.

The time-consuming and data-intensive nature of full backups led to the invention of *incremental* and *differential* backups. An incremental backup saves any data changed

since the last backup. A differential backup saves any data changed since a full backup (level zero), or since a differential backup of lower level. Differential backups are also called *level* backups. Figure 7-4 shows a scheme using incremental backups during the week, and various levels of differential backups during weekends. Unlike incrementals, repeated level-one backups would go all the way back to a full backup.

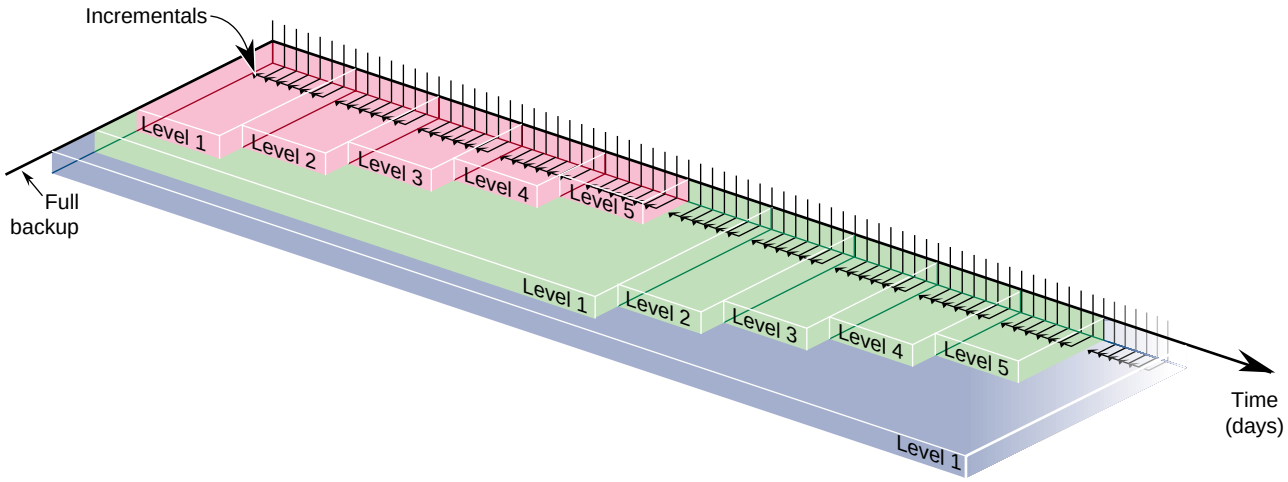

Figure 7-4 Incremental and Differential Backups

Because incremental and differential backups save only files that change, less data is involved, reducing backup time and overall tape consumption. The downside is that more tapes are required to restore a filesystem, because full backups must be overlaid with increasing levels of differential backups, then with incremental backups.

Scheduled backup software should be able to include and exclude specific sets of files. Some filesystems can be recreated from software distributions, except for a limited set of files that should be saves. On UNIX systems, core dumps do not typically need saving.

The advent of robotic tape libraries has made the distasteful task of changing tapes for weekend backups all but a thing of the past. It has also reduced total backup time, because robotic libraries can change tapes as soon as they become full, rather than waiting for an operator to load a new tape.

In most places, the backup window starts when the last employee goes home, and begins when the first employee arrives at work. (Backup is less reliable when files are changing.) Capacity planning should take into account the amount of data to be saved, the duration of the backup window, and the bandwidth of backup hardware.

Network backup has become critical now that productive work occurs on workstations and personal computers. Cost and performance considerations dictate that important data be kept on local disk. For convenience, and to ensure the integrity of backup data, workstations and personal computers can be backed up across the network. This is the usual client-server model—a server with huge tape capacity backs up a set of clients.

7.3.1.2 Backup Software

To reduce the workload for system administrators, backup software should provide convenient file retrieval for backup clients, including file search and version facilities. This brings up the security issue. Backup software must preserve normal system security, not allowing searching or recovery of other users' files. At the same time, administrators must be allowed to recover data for departed users, and to move data from one client to another when necessary.

Scheduled backup software should help you manage media, by including features for tape recycling, media aging, device cleaning, and perhaps bar code reading. Software should also be able to log backup failures due to power loss, hardware problems, and media bad spots, and notify administrators when intervention is required.

As stated above, the principal purpose of backup is to provide a cost-effective method for disaster recovery. For this to work well, you must create a disaster recovery plan and test all procedures to make sure they work correctly.

Storing backups offsite, in a fire vault, or both ways, is your only hope of recovery from fires and natural catastrophes. Clearly the frequency with which these are done equals the amount of work at risk: monthly offsites jeopardize a full month's data.

7.3.1.3 Supplemental Software

Database backup presents a challenge for backup software, because databases change internally, rather than on a per-file basis. Backing up a large database can take longer than the backup window allows. As a solution, database vendors often provide backup methods to save only changed data, and to roll in these changes if recovery is needed.

Taking a database offline for backup is simpler and faster, although many databases have 7-day, 24-hour uptime requirements. Saving from a live database is called *hot* backup.

Archiving involves taking a snapshot of data files as they reside on disk at a given time. The snapshot image is typically stored on removable media, such as tape or optical disc. Once the snapshot is safely stored, archived files may be deleted to conserve disk space. Whereas the goal of backup is to protect data against accidental loss or damage, the goals of archiving are to preserve data and to conserve online storage space.

Archiving is normally performed on data associated with specific projects, rather than on an entire system. Backup tapes are usually recycled or discarded, while archive media are intended to last a long time. For this reason, recordable CD is the ideal archive media, because it is more universal and permanent than tape.

7.3.2 Hierarchical Storage Management

HSM (hierarchical storage management) is a storage strategy that involves moving files from one medium to another, based on configurable a set of rules. One common rule is based on access rate—when a file becomes inactive, it get migrated. Storage hierarchy is usually governed by media cost and random access time. The goal of HSM is to conserve network storage resources, thereby providing users with a seemingly infinite storage capacity, at the lowest possible cost.

HSM was developed in the 1970s for use in mainframe applications, when disk storage was much more expensive than it is today, and tape storage was comparatively cheaper. According to one HSM manufacturer, between 60% and 80% of files on a typical system have not been accessed in 90 days; so HSM remains a viable strategy.

After migration, a stub file is left on disk as a link to the actual file on alternate media. When a user accesses a stub file, the HSM software locates the actual file on alternate media and restores the original data to disk. Most HSM systems are configured with three types of storage:

- Online (hard disk drives)
- Nearline (often magneto-optical jukeboxes for random access)
- Farline (usually high capacity tape drives)

While hard disks have file access time in the millisecond range, optical jukeboxes have access times in the multiple second range. Tape libraries have file access times that vary widely depending on where data exists on tape, on the order of several minutes.

Most HSM software can be configured with a list of directories not to migrate. Also, the administrator can set high and low watermarks for migration time at each storage level, thereby controlling latency to suit user preferences.

7.3.3 Enterprise Storage Control

In large networks of heterogeneous systems, the management of scheduled backups can be a major chore. Several products are available to help deal with enterprise issues.

SNMP (simple network management protocol) has features to help manage backups in a network environment. Many network management products integrate SNMP support.

Alexandria, a high-performance backup product by Spectra Logic, can coordinate server and database backups across large networks, and includes facilities for cross-backup and storage node sharing.

NetWorker, a backup product for heterogeneous networks by Legato Systems, optionally includes GEMS (global enterprise management system) for managing storage nodes and enterprise backup scheduling.

OpenVault Troubleshooting

This appendix explains how to resolve error conditions, and includes a description of the possible error messages.

Most OpenVault errors are found in the log file `/var/opt/openvault/server/logs/OVLOG.`*yyyymmdd*. They can also be listed with the `ov_msg -l` option; see the `ov_msg`(1M) man page.

A.1 Error Conditions

Errors such as tape read and write errors are associated with an application error, whereas conditions such as `cannot find tape` or `cannot associate with driver` are OpenVault errors.

A.2 Accessing OpenVault Messages

OpenVault system messages are appended to the logfile, `/var/opt/openvault/server/logs/OVLOG.`*yyyymmdd*.

To view all the warning, error, and operator intervention messages that are contained in the message log on the default server, use the `ov_msg` command; see the `ov_msg`(1M) man page.

A.3 Error Messages and Actions

Refer to the OpenVault release notes for a listing of the error messages that apply to the OpenVault system.

A.4 OpenVault Processes and Files

Table A-1 shows some important OpenVault files located in `/var/opt/openvault`.

Table A-1 OpenVault Configuration Files

OpenVault File	What it Controls
`server/config/core_keys`	File listing clients allowed OpenVault access, possibly with passwords.
`clients/admin/keys`	File storing security passwords for OpenVault applications.
`server/logs/OVLOG.`*yyyymmdd*	Repository of all events and errors that occur within OpenVault.
`server/config/config`	Configuration file for OpenVault server, MLM (media library manager).
`clients/dcp/`*tapeN*`/`*instX*`/config`	Configuration file for the drive named *tapeN* and instance name *instX*.
`clients/lcp/`*libN*`/`*instY*`/config`	Configuration file for the library named *libN* and instance name *instY*.
`clients/dcp/`*tapeN*`/`*instX*`/logfile`	Event and error log for the drive named *tapeN* and instance name *instX*.
`clients/lcp/`*libN*`/`*instY*`/logfile`	Event and error log for the library named *libN* and instance name *instY*.

Table A-2 shows some important OpenVault processes.

Table A-2 OpenVault Processes

OpenVault Process	What It Does
`ovroot`	If `chkconfig` is on, this process mediates and delegates service requests.
`MLM_catalog`	Grants services as requested, and manages the OpenVault catalog.
`MLM_capi`	Started by the server to manage each registered OpenVault application.
`MLM_aapi`	Started by the server to manage each OpenVault administrative application.
`MLM_dcp`	Started by the server to manage each attached drive and its related DCP.
`MLM_lcp`	Started by the server to manage each attached library and its related LCP.
DCP*tapeN*	The DCP (drive control program) that manages the drive named *tapeN*.
LCP*libN*	The LCP (library control program) that manages the library named *libN*.
ssi | lmcpd	Product-specific processes to manage ACSLS and IBM-3494, respectively.

A.5 Troubleshooting OpenVault Commands

When trying to troubleshoot OpenVault administrative commands, it's helpful to look inside the command to see what it is doing. The IRIX `par` command shows what system calls a program makes. This is useful, but OpenVault offers a better and more detailed way to see what an OpenVault command is doing. Before any troubleshooting can begin, debugging must be turned on in the OpenVault server, using the following command:

```
# /usr/sbin/ov_msg -s -t core -m debug
```

Once this is done, the OpenVault server starts writing a lot more messages to the OVLOG.*yyyymmdd* file; so you can run a command to see what it is doing. OVLOG messages contain a lot of useful information. Here is a sample log entry:

```
Apr 26 17:32:44 mlm_aapi:SYSTEM:onlyInstance: (debug) [177600] WRITTEN:
response whichtask[‘5’] accepted ;
```

This entry shows the following information:

Apr 26 17:32:44	Current date and time.
mlm_aapi	The OpenVault process.
SYSTEM	The application.
onlyInstance	There are no other such applications.
(debug)	Level of the message; can be info, error, debug.
[177600]	Process ID of the OpenVault process.
WRITTEN: ...	The commands being passed along.

When an OpenVault command is run (or when any connection is made to OpenVault), a process is created to handle the connection. This is listed in OVLOG (and in the process table) as mlm_aapi. Following this process, you can see what the MLM server sees when it talks to the calling command or application as shown in Example A-1.

Example A-1 mlm_aapi Calls and Their PIDs

To get a list of individual mlm_aapi calls and their PIDs, run the following command:

```
# grep ’welcome version’ OVLOG* | grep mlm_aapi
Apr 26 17:32:43 mlm_aapi:SYSTEM:onlyInstance: (debug) [177600] WRITTEN:
      welcome version[‘1.0’];
Apr 26 17:32:57 mlm_aapi:SYSTEM:onlyInstance: (debug) [177685] WRITTEN:
      welcome version[‘1.0’];
Apr 26 17:33:27 mlm_aapi:SYSTEM:onlyInstance: (debug) [177772] WRITTEN:
      welcome version[‘1.0’];
```

This listing shows the AAPI/CAPI reply welcome that the mlm_aapi sends to the calling command. There is only one welcome command per connection, so this should give a unique list of mlm_aapi processes.

Once the PID is found, grep can be used to locate all of the OVLOG messages that relate to that one command or connection. Example A-2, page 129, shows an ov_mount command:

Example A-2 ov_mount Search

```
# ov_mount -V A05444 -A dmf
```

To find the specific PID for that ov_mount command, run this grep command, which looks for any OpenVault commands showing the volume A05444 and using the application dmf.

```
# grep A05444 OVLOG* | grep dmf
Apr 26 14:52:24 mlm_aapi:SYSTEM:onlyInstance: (debug) [172088] READ:
    mount match [and( strEQ(VOLUME.'VolumeName' 'A05444') strEQ(VOLUME.
    'ApplicationName' 'dmf'))] report[VOLUME.'VolumeName' MOUNTLOGICAL.
    'MountLogicalHandle'] reportmode[value]  task['0'];
```

Make sure that the time listed in the results corresponds with the time that the ov_mount command was run. Now that you found the PID for this particular command, do a grep for all messages relating to that command:

```
# grep 172088 OVLOG*
Apr 26 14:52:23 mlm_aapi:SYSTEM:onlyInstance: (debug) [172088] WRITTEN:
    welcome version['1.0'];
Apr 26 14:52:23 mlm_aapi:SYSTEM:onlyInstance: (debug) [172088] /usr/OpenVault
    /mlm/MLM_aapi: submain(): process_peer() returned 1
Apr 26 14:52:23 mlm_aapi:SYSTEM:onlyInstance: (debug) [172088] READ: mount
    match[and( strEQ(VOLUME.'VolumeName' 'A05443') strEQ(VOLUME.
    'ApplicationName' 'dmf'))] report[VOLUME.'VolumeName' MOUNTLOGICAL.
    'MountLogicalHandle'] reportmode[value]  task['0'];
Apr 26 14:52:23 mlm_aapi:SYSTEM:onlyInstance: (debug) [172088] WRITTEN:
   response whichtask['0'] accepted ;
...
```

This shows the entire AAPI conversation from the view of the mlm_aapi or OpenVault server. All commands sent and received are shown exactly as transmitted.

OpenVault Man Pages

Table B-1 provides a brief introduction to the man pages for commands used to administer and operate OpenVault. Refer to the man page itself for a complete description of options and usage information.

Man pages with a `1M` designation on IRIX systems are designated by an `8` on SGI ProPack for Linux systems. For example, `ov_app.1m` on IRIX systems is `ov_app.8` on SGI ProPack for Linux systems.

Table B-1 OpenVault Man Pages

Name	Use	Description
`ov_admin`(1M)	Administration	Script for configuring OpenVault.
`ov_app`(1M)	Administrative	Adds, removes, or lists applications.
`ov_cart`(1M)	Data entry	Allows entry or modification of cartridge information (PCL, cartridge group, attributes, and so forth).
`ov_cartgroup`(1M)	Administrative	Adds or removes cartridge groups, and sets or modifies group priority and access permissions.
`ov_carttype`(1M)	Status	Manages OpenVault cartridge types.
`ov_dcstats`(1M)	Status	Displays drive and cartridge use statistics.
`ov_drive`(1M)	Administrative	Adds, removes, or lists drives.
`ov_drivegroup`(1M)	Administrative	Adds or removes drive groups, and sets or modifies group priority and access permissions.
`ov_eject`(1M)	Operation	Ejects cartridges from library, by location or by cart ID.
`ov_import`(1M)	Operation	Brings cartridges under OpenVault management.
`ov_inject`(1M)	Operation	Injects cartridges into a library—the library loading port opens so the operator can insert a cartridge.
`ov_library`(1M)	Administrative	Adds, removes, or lists libraries.

Name	Use	Description
ov_lscarts(1M)	Status	Provides cartridge status, including cartridge type, state, location, group, association with application, partition and volume mapping, and so forth.
ov_lsvols(1M)	Status	Provides volume status, including names, partitions, association with application, and containing cart ID.
ov_mount(1M)	Operation	Mounts a cartridge in preparation for read or write by an application. Options specify cart ID and a particular drive, media side, partition, volume, or application.
ov_msg(1M)	Administrative	Administers the message system. Allows for listing and deleting messages, and setting levels or limits.
ov_part(1M)	Data entry	Sets or modifies media partition information to prepare for volume setup. Information includes partition name, side, partition size, whether allocatable, and so forth.
ov_procs(1M)	Administrative	Displays running OpenVault processes.
ov_purge(1M)	Operation	Removes the catalog entry for the cartridge. This should be used only when a cartridge is to be destroyed.
ov_recycle(1M)	Operation	Makes unused cartridges available for reuse.
ov_scandev(1M)	Administrative	Provides information about devices on the host.
ov_shutdown(1M)	Administrative	Shuts down and restarts OpenVault.
ov_slotttype(1M)	Status	Manages OpenVault slot types.
ov_start(1M)	Administrative	Starts OpenVault components.
ov_stat(1M)	Status	Provides status about OpenVault servers and components, including up or down state, configuration information, slot maps of libraries, application registration status, task queues, missing DCP and LCP software, and so forth.
ov_stop(1M)	Administrative	Stops OpenVault components.

Name	Use	Description
ov_system(1M)	Administrative	Sets or retrieves information from the system table for system-level control.
ov_task(1M)	Operation	Removes a task request from the server task queue.
ov_umsh(1M)	Operation	User mount shell. Allows users to access media managed by the OpenVault media management system.
ov_unmount(1M)	Operation	Unmounts cartridge from a drive. Options specify cart ID and a drive, media side, partition, volume, or application.
ov_vol(1M)	Data entry	Allows entry or modification of volume information, including volume name, associated application, attribute name/value pairs, and so forth.

Index

D

E

F