

OpenVault™ Operator's and Administrator's Guide

Document Number 007-3211-003

CONTRIBUTORS

Written by Susan Thomas
Production by Allen Clardy
Engineering contributions by Curtis Anderson, Murali Sathyanarayana,
and Steve Philipson.

© 1997-1998, Silicon Graphics, Inc.— All Rights Reserved

The contents of this document may not be copied or duplicated in any form, in whole or in part, without the prior written permission of Silicon Graphics, Inc.

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure of the technical data contained in this document by the Government is subject to restrictions as set forth in subdivision (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 52.227-7013 and/or in similar or successor clauses in the FAR, or in the DOD or NASA FAR Supplement. Unpublished rights reserved under the Copyright Laws of the United States. Contractor/manufacturer is Silicon Graphics, Inc., 2011 N. Shoreline Blvd., Mountain View, CA 94043-1389.

Silicon Graphics and the Silicon Graphics logo are registered trademarks and IRIX Networker is a trademark of Silicon Graphics, Inc. UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company, Ltd. POSIX is a registered trademark of the Institute of Electrical and Electronic Engineers, Inc.

Exabyte is a registered trademark, and Mammoth a trademark, of Exabyte Corp. IBM is a registered trademark, and Magstar a trademark, of IBM Corp. Quantum is a registered trademark, and DLT a trademark, of Quantum Corp. Sony is a registered trademark, and AIT a trademark, of Sony Corp. StorageTek is a registered trademark of Storage Technology Corp.

Contents

	List of Examples	ix
	List of Figures	xi
	List of Tables	xiii
	About This Guide	xv
	Contents of This Guide	xv
	Audience	xvi
	Style Conventions	xvi
	Product Support	xvi
1.	Introducing OpenVault	1
	What OpenVault Does	1
	How OpenVault Fits In With Other Software	2
	OpenVault as Middleware	2
	Client-Server Model	3
	OpenVault Terms	3
	OpenVault Definitions	4
	OpenVault Architecture	5
	AAPI Programming Interface	7
	CAPI Programming Interface	7
	OpenVault Server	7
	ALI/LCP Interface	8
	ADI/DCP Interface	8
	How OpenVault Operates	8
	LCP Booting	8
	DCP Booting	9
	Cartridge Life Cycle	10

- 2. Installing and Configuring OpenVault 11**
 - OpenVault Installation Requirements 11
 - OpenVault Software Components 12
 - Configuration Terminal 13
 - Media in Cartridges 13
 - Licensing 13
 - 2.2 Sample Configurations 14
 - Custom Installation 17
 - Determining Attached SCSI Drives 18
 - Non SCSI-Attached Libraries 18
 - Other Guidelines for Custom Installation 19
 - Configuration Roadmap 21
 - Preparing OpenVault Devices and Hosts 22
 - Configuration Worksheet 23
 - Completing the Worksheet 26
 - Acquiring Hostnames for OpenVault Hosts 26
 - Generating Unique Names for Libraries 27
 - Collecting Information for Attached Drives 27
 - Collecting Information for SCSI-Attached Libraries 29
 - Collecting Information for ACSLS Libraries 30
 - Collecting Information for IBM-3494 Libraries 31
 - Collecting Information for EMASS-Grau Libraries 31
 - Planning Cartridge and Drive Groups 31
 - Media considerations 32
 - Selecting a Password 32
 - Install the OpenVault Flexlm license 32
 - Library and Drive Names 32
 - Library Names 32
 - Drive Names 33

- Server Configuration 34
 - Configuring OpenVault Core on the OpenVault Server 35
 - Configure Locally Attached Drives on the OpenVault Server 37
 - Configuring Locally-Attached Libraries on OpenVault Servers 38
 - Configuring SCSI- Attached Libraries 38
 - Configuring an IBM-3494 Library 42
 - Configuring an EMASS-Grau library 45
 - Configuring a StorageTek ACSLS library 45
 - Enabling Remote Administration, LCPs and DCPs 49
 - Enable remote applications 50
 - Enabling Remote Libraries 50
 - Enabling Remote Drives 51
- Client Configuration 52
 - Configure Attached Drives on OpenVault Client Hosts 54
 - Configuring Attached Libraries on OpenVault Client Hosts 54
- Importing Media 56
- 3. Cartridge Life Cycle 61**
 - Cartridge States 61
 - Managing Cartridges 64
 - Physical Entry Into a Library 64
 - Examining the Contents of a Library 64
 - Creating Cartridge Data 66
 - Displaying Cartridge Data 67
 - Creating Partitions 68
 - Controlling Allocation Status 68
 - Allocating Volumes 69
 - Displaying Volume Data 70
 - Deallocating Volumes 70
 - Recycling Cartridges 70
 - Simplified Entry of Media Information 71
 - Removing Cartridges From Libraries 73
 - Purging Cartridge Data 73

- 4. Administering OpenVault 75**
 - Before You Start 75
 - Installing OpenVault 76
 - Removing OpenVault 76
 - Setting Up OpenVault 76
 - Setting Up the Server 77
 - Importing Media 79
 - Configuring After Setup 80
 - Setting Up Clients 81
 - Setting Up Drives 81
 - Setting Up Robotic Libraries 81
 - Setting Up Applications 82
 - Setting Up Security 82
 - Setting Up Non-Robotic Libraries 83
 - Setting Up Offsite Libraries 83
 - Configuring OpenVault 84
 - Setting Logging Levels 84
 - Registering Applications 85
 - Unregistering Applications 85
 - Setting Up Application Security 86
 - Enabling Application Access to a Drive 86
 - Managing Cartridge Groups 87
 - Setting Up Cartridge Groups 88
 - Introducing Cartridges 89
 - Monitoring OpenVault 91
 - Checking Server Status and Configuration 92
 - Checking Media Inventory 93
 - Listing Cartridge Information 93

- 5. **Operating OpenVault** 95
 - Performing Daily Tasks 95
 - Manipulating Cartridges 95
 - Mounting Cartridges 95
 - Checking the Task Request Queue 96
 - Canceling a Pending Task Request 96
 - Managing Devices 97
 - Disabling and Enabling Devices 97
 - Cleaning a Drive 98
 - Performing Occasional Tasks 98
 - Removing Cartridges From Libraries 99
 - Moving Cartridges Within Libraries 99
 - Maintaining the Server Catalog 100
 - Recycling Cartridges 100
 - Destroying Cartridges 100
- 6. **Tertiary Storage Management** 101
 - Tertiary Storage Devices 101
 - Tape Drives 101
 - Preventative Maintenance 102
 - Optical Drives 103
 - SCSI Media Changers 103
 - Silo Libraries 103
 - Connecting to a Host Computer 104
 - SCSI Connection Guidelines 104
 - Silicon Graphics Servers 106
 - Storage Management Applications 107
 - Scheduled Backup 107
 - Supplemental Software 109
 - Hierarchical Storage Management 110
 - Enterprise Storage Control 111

- A. OpenVault Error Messages 113**
 - Error Conditions 113
 - Accessing the OpenVault Message Log 113
 - Error Messages 113
- B. OpenVault Reference Pages 115**
 - Index 117**

List of Examples

- Example 4-1** OpenVault Configuration Menu 80
Example 4-2 Client and Server Key Authorization File 82

List of Figures

Figure 1-1	OpenVault Architecture	6
Figure 2-1	Local-Only OpenVault Configuration	14
Figure 2-2	Local-and-Remote OpenVault Configuration	15
Figure 3-1	Cartridge Life Cycle (Simplified)	63
Figure 4-1	Example Cartridge Group (Engineering)	88
Figure 6-1	68-pin Wide SCSI-3 Connector	105
Figure 6-2	50-pin SCSI-2 Connector (Mini-micro)	105
Figure 6-3	50-pin Centronics Parallel Connector	105
Figure 6-4	Incremental and Differential Backups	108

List of Tables

Table 2-1	OpenVault Configuration Roadmap	21
Table 2-2	Host Worksheet	23
Table 2-3	Drive Worksheet (SCSI)	23
Table 2-4	Drive Worksheet (IBM-3494)	24
Table 2-5	Drive Worksheet (ACSLs)	24
Table 2-6	Library Worksheet (SCSI)	25
Table 2-7	Library Worksheet (IBM-3494)	25
Table 2-8	Library Worksheet (ACSLs)	26
Table 4-1	Key Authorization File Description	82
Table 6-1	High Capacity Tape Drives	102
Table 6-2	SCSI Types and Speeds	105
Table B-1	OpenVault Reference Pages	115

About This Guide

This *OpenVault Operator's and Administrator's Guide* describes how to administer and operate OpenVault. It also provides an introduction to tertiary storage management.

OpenVault is an emerging standard solution for tertiary storage management from Silicon Graphics. It provides a common interface for applications that manage removable media libraries, loadable drives, and OpenVault client applications.

This book does not describe how to develop OpenVault applications or device support—refer to the *OpenVault Applications Programmer's Guide* and the *OpenVault Infrastructure Programmer's Guide* for this information. Regular IRIX storage management utilities are described in the book *IRIX Admin: Backup, Security, and Accounting*.

Contents of This Guide

This guide contains the following chapters.

Chapter 1, "Introducing OpenVault," describes OpenVault architecture and operation.

Chapter 2, "Installing and Configuring OpenVault" details server and client setup.

Chapter 3, "Cartridge Life Cycle" discusses the treatment of media cartridges.

Chapter 4, "Administering OpenVault," presents procedures for system administrators.

Chapter 5, "Operating OpenVault," tells how to perform day-to-day operator tasks.

Chapter 6, "Tertiary Storage Management," is a conceptual introduction to this topic.

Appendix A, "OpenVault Error Messages," discusses OpenVault error conditions.

Appendix B, "OpenVault Reference Pages," lists OpenVault administration commands.

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

Style Conventions

This guide uses the following stylistic conventions:

<code>screen font</code>	Indicates system output, such as responses to commands appearing on the screen. Code samples and screen displays also appear in this font.
user input	Indicates what you must enter at a command line, such as a command and optional arguments to commands.
<i>variable</i>	Indicates a substitutable string (for example, <i>logFile</i>) that you replace with a specific value (for example, <i>myLog</i>).
<i>command</i>	Designates command and utility names.
<i>filename</i>	Indicates directory pathnames and filenames.
...	Denotes omitted material or indicates that the preceding optional items may appear more than once in succession.

Product Support

Silicon Graphics, Inc., provides a comprehensive product support and maintenance program for its products. If you are in North America and would like support for your Silicon Graphics products, contact the Technical Assistance Center at 1-800-800-4SGI. If you are outside North America, contact the Silicon Graphics subsidiary or authorized distributor in your country.

Introducing 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:

- “What OpenVault Does” on page 1
- “How OpenVault Fits In With Other Software” on page 2
- “OpenVault Terms” on page 3
- “OpenVault Architecture” on page 5
- “How OpenVault Operates” on page 8

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

- 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 and Chapter 5 describe how to perform administrator and operator tasks. Appendix B provides an overview of OpenVault administrative commands.

How OpenVault Fits In With Other Software

OpenVault, as middleware, operates between applications and devices, as shown in Figure 1-1. 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.

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:

1. for storage-based applications that use cartridges
2. for devices (libraries and drives) that store and retrieve information on cartridges

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.

OpenVault Terms

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

A-API	Administrative application programming interface. Used to make administrative requests of the system.
A-API/R	A-API 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.

OpenVault Definitions

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.
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.

OpenVault system	The set of hosts (server and clients) comprise the OpenVault system.
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 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 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.
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.
DCP	The OpenVault client process(es) that directly control a library under OpenVault management. The DCP interprets and services ADI requests on behalf of the OpenVault core.

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 shows the arrangement of OpenVault components.

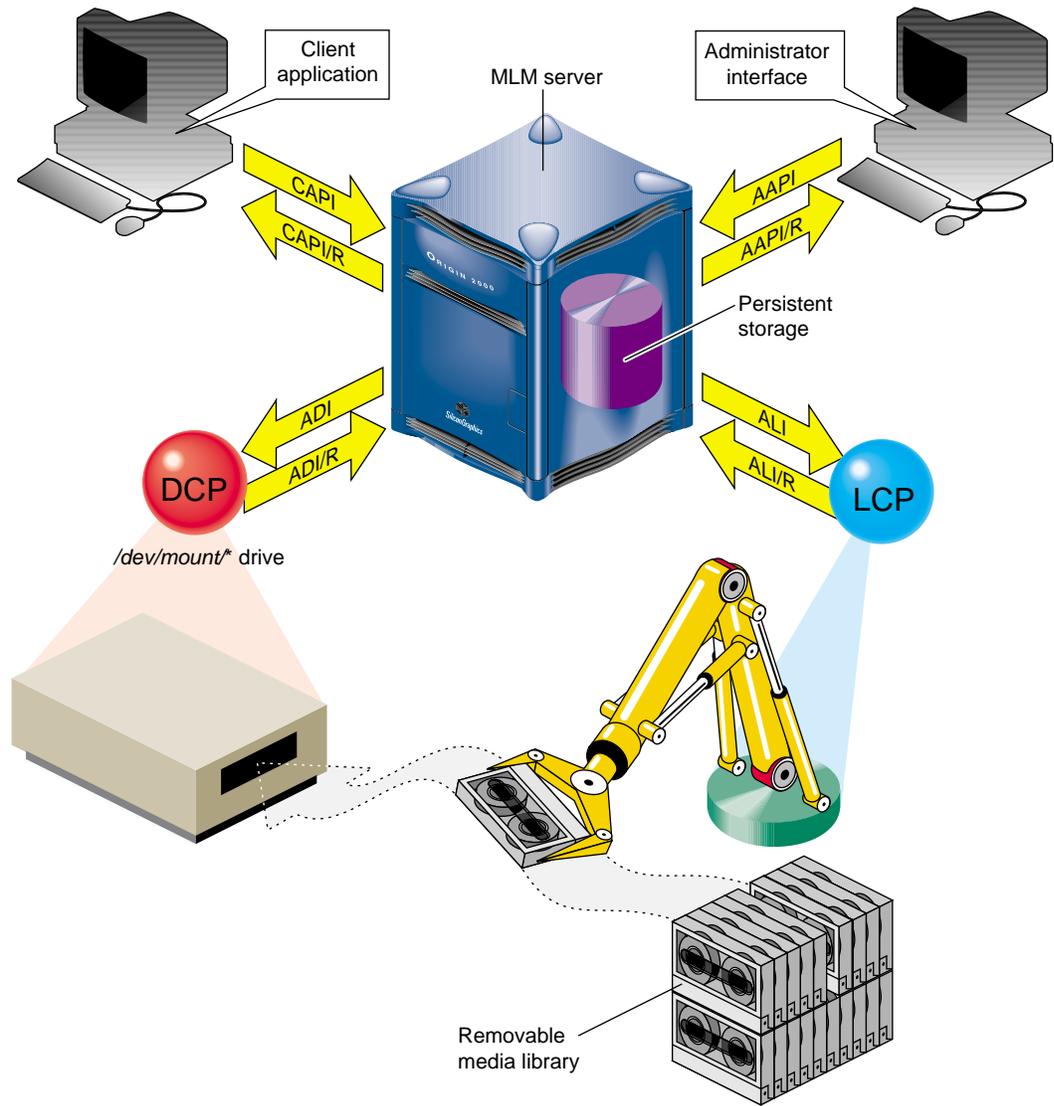


Figure 1-1 OpenVault Architecture

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.

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.

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.

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).

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 Programmers's Guide* describes how to program ALI/LCP and ADI/DCP.

How OpenVault Operates

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

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
- the 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 the OpenVault catalog can be updated, if necessary. At this point, the library can accept mount and unmount commands from OpenVault.

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
- the supported media formats (for example, EXABYTE-8mm-5GB)
- whether the listed access modes are supported
- whether a cartridge is loaded in the drive
- 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 the OpenVault catalog can be updated, if necessary.

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.

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 it ceases to be a part of that system. The major events in the life of a cartridge include:

- the 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

For more information on this topic, see Chapter 3, “Cartridge Life Cycle.”

Installing and Configuring OpenVault

This chapter describes how to configure OpenVault for the first time. There are two types of OpenVault hosts: the server, which contains the OpenVault core; and remote hosts, which do not contain an OpenVault core. Depending on which hosts have drives and libraries physically or logically connected, configurations fall into one of two categories:

- Local-only configuration

In this configuration, all drives, libraries, and applications that are managed by or serviced by an OpenVault Server reside on the OpenVault server host.

- Local-and-Remote configuration

In this 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, “Administering OpenVault.”

OpenVault Installation Requirements

This OpenVault release supports the following drives on IRIX versions 6.2 and above:

- IBM-3590 Magstar
- DLT-7000 and DLT-2000
- StorageTek STK-Timberline and STK-Redwood

This OpenVault release supports the following libraries on IRIX versions 6.2 and above:

- StorageTek STK-9710, STK-9714, and STK-9730
- StorageTek ACSLS controlled libraries.
- IBM-3494
- EMASS Grau

OpenVault Software Components

If you have not already installed OpenVault yet, you can install OpenVault either using the `inst` command or Software Manager. For more information on these installation programs, see the `inst(1M)` or `swmgr(1M)` man pages.

If you install the default OpenVault product subsystems, your installation includes the following software:

<code>OpenVault.sw.core</code>	OpenVault core servers
<code>OpenVault.sw.admin</code>	OpenVault administrative tools
<code>OpenVault.sw.libs</code>	OpenVault core runtime libraries
<code>OpenVault.sw.user</code>	OpenVault end-user tools
<code>OpenVault.man.manpages</code>	OpenVault manual pages
<code>OpenVault.man.relnotes</code>	OpenVault release notes
<code>OpenVault.dcp.libs</code>	OpenVault run-time libraries for DCPs
<code>OpenVault.dcp.DLT2000</code>	OpenVault DCP for the DLT2000 drive
<code>OpenVault.dcp.DLT7000</code>	OpenVault DCP for the DLT7000 drive
<code>OpenVault.dcp.EXB8505XL</code>	OpenVault DCP for the EXB-8505 drive
<code>OpenVault.dcp.IBM3590</code>	OpenVault DCP for the IBM-3590 drive
<code>OpenVault.dcp.STKredwood</code>	OpenVault DCP for the STK-redwood drive
<code>OpenVault.sw.config</code>	OpenVault configuration scripts
<code>OpenVault.sw.core</code>	OpenVault core servers
<code>OpenVault.sw.admin</code>	OpenVault administrative tools
<code>OpenVault.sw.libs</code>	OpenVault core runtime libraries
<code>OpenVault.sw.user</code>	OpenVault end-user tools
<code>OpenVault.man.manpages</code>	OpenVault manual pages
<code>OpenVault.man.relnotes</code>	OpenVault release notes
<code>OpenVault.dcp.libs</code>	OpenVault run-time libraries for DCPs
<code>OpenVault.dcp.DLT2000</code>	OpenVault DCP for the DLT2000 drive
<code>OpenVault.dcp.DLT7000</code>	OpenVault DCP for the DLT7000 drive
<code>OpenVault.dcp.EXB8505XL</code>	OpenVault DCP for the EXB-8505 drive
<code>OpenVault.dcp.IBM3590</code>	OpenVault DCP for the IBM-3590 drive
<code>OpenVault.dcp.STKredwood</code>	OpenVault DCP for the STK-redwood drive
<code>OpenVault.dcp.STKtimberline</code>	OpenVault DCP for the STK-timberline drive
<code>OpenVault.dcp.pseudo</code>	OpenVault DCP for the "pseudo" drive
<code>OpenVault.dev.examples</code>	OpenVault example sources for applications
<code>OpenVault.dev.include</code>	OpenVault app C/C++ include headers
<code>OpenVault.docs.adminguide</code>	OpenVault administrative guide
<code>OpenVault.docs.architecture</code>	OpenVault architecture specifications
<code>OpenVault.docs.developer</code>	OpenVault LCP/DCP/app developer guides

OpenVault.lcp.libs	OpenVault run-time libraries for LCPs
OpenVault.lcp.ATL2640	OpenVault LCP for the ATL-2640 library
OpenVault.lcp.EMASS_Grau	OpenVault LCP for the EMASS-Grau library
OpenVault.lcp.EXABYTE210	OpenVault LCP for the EXB-210 library
OpenVault.lcp.IBM3494	OpenVault LCP for the IBM-3494 library
OpenVault.lcp.STK9700	OpenVault LCP for the STK-97XX library ¹
OpenVault.lcp.STKACSLs	OpenVault LCP for the STK-ACSLs library ²
OpenVault.lcp.pseudo	OpenVault LCP for the "pseudo" library

1. This includes the STK-9710, STK-9714 and STK-9730 libraries.
2. Includes libraries that can be controlled via the Storage Tek ACSLS interface.

Configuration Terminal

Though it is not required, it is recommended that while configuring OpenVault is recommended that you use a graphics capable console attached to each host machine in the OpenVault system, i.e., capable of resizing and scrolling.

Media in Cartridges

This release of OpenVault supports bar-coded tape media only.

Licensing

You will be required to acquire a license for running OpenVault. This license needs to be installed on the OpenVault server host.

Licensing software must be installed before OpenVault, but only on the OpenVault server host machine. IRIX 6.2 users need to install the License Tools 2.1.1 or later package. IRIX 6.4 users require License Tools 3.0 or higher. To check whether License Tools are already installed, run the following command:

```
# versions -b license_eoe
```

To obtain a license visit the Web site <http://www.sgi.com/Products/license.html> or by Email, send a blank email message for the license template to license@sgi.com. Via FAX, in North America call 650-390-0537. The postal address is:

Silicon Graphics, Inc.
Attn. License Administration MS 134
2011 North Shoreline Blvd.
Mountain View, CA 94043-1389.

The OpenVault license you receive includes instructions on how to install it; you must add information provided on your license sheet to the `/var/flexlm/license.dat` file.

2.2 Sample Configurations

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

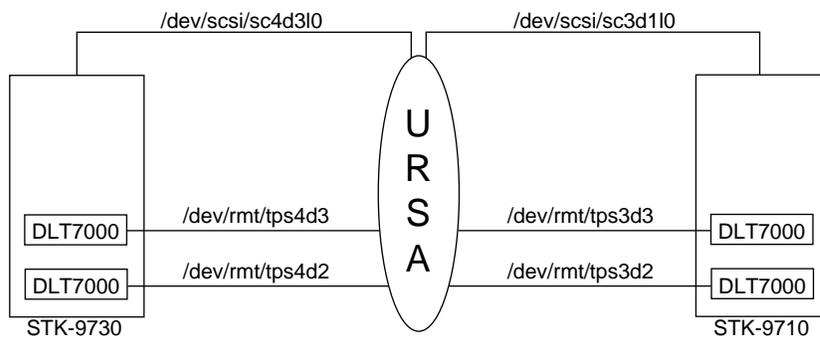


Figure 2-1 Local-Only OpenVault Configuration

In Figure 2-1, the OpenVault server host, `ursa`, contains two libraries, `STK-9710` and `STK-9730`. `STK-9710` is connected at device address `/dev/scsi/sc3d110`, and `STK-9730` is connected at device address `/dev/scsi/sc4d110`.

The `STK-9710` has the following drives:

- DLT7000 connected at address `/dev/rmt/tps3d2`, located physically as the bottom drive in the library.
- DLT7000 connected at address `/dev/rmt/tps3d3`, located physically as the top drive in the library.

The `STK-9730` has the following drives:

- DLT7000 connected at address `/dev/rmt/tps4d2`, located physically as the bottom drive in the library.
- DLT7000 connected at address `/dev/rmt/tps4d3`, located physically as the top drive in the library.

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

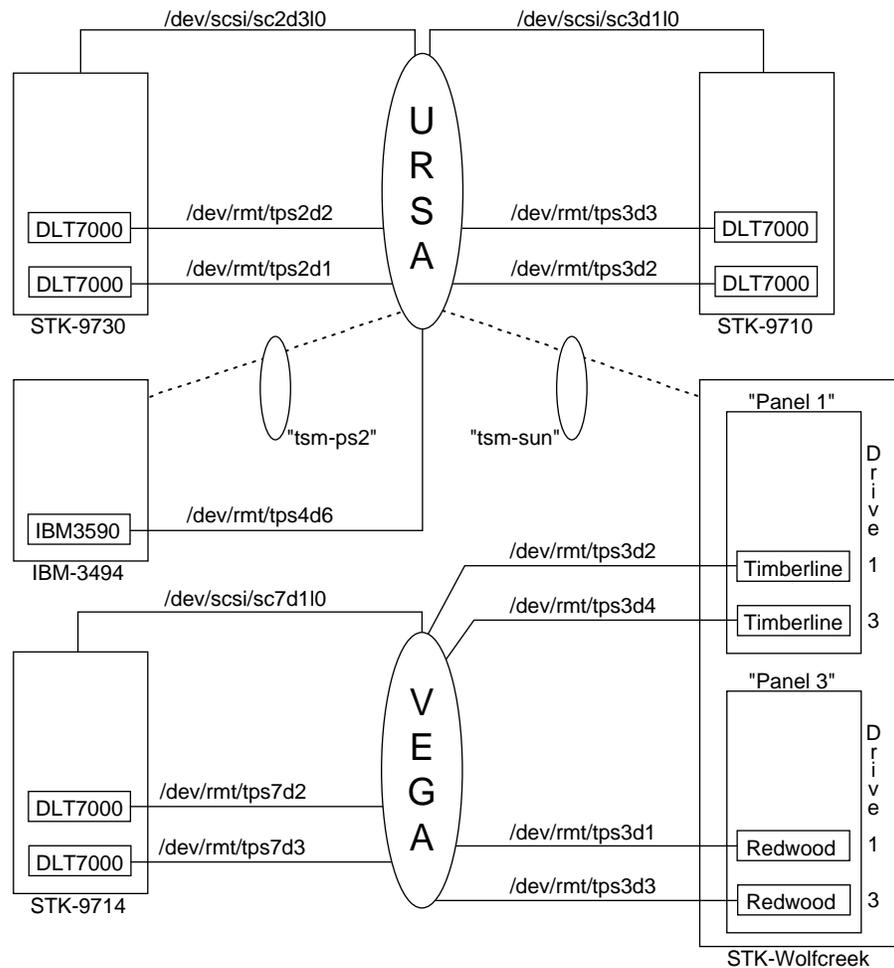


Figure 2-2 Local-and-Remote OpenVault Configuration

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

- STK-9710 connected at device address `/dev/scsi/sc3d110`
- STK-9730 connected at device address `/dev/scsi/sc2d310`
- StorageTek WolfCreek silo connected via the ACSLS server host, tsm-sun

- IBM-3494 connected via an IBM library server running on the host, tsm-ps2

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

- DLT7000 connected at address */dev/rmt/tps3d2*, located physically as the bottom drive in the library.
- DLT7000 connected at address */dev/rmt/tps3d3*, located physically as the top drive in the library.

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

- DLT7000 connected at address */dev/rmt/tps2d1*, located physically as the bottom drive in the library.
- DLT7000 connected at address */dev/rmt/tps2d2*, located physically as the top drive in the library.

The StorageTek ACSLS controlled WolfCreek silo has the following drives, all of which are connected to vega:

- STK-Redwood connected at address */dev/rmt/tps3d1*, located physically in LSM 2, Panel 3, as Drive 1
- STK-Redwood connected at address */dev/rmt/tps3d3*, located physically in LSM 2, Panel 3, as Drive 3
- STK-Timberline connected at address */dev/rmt/tps3d2*, located physically in LSM 2, Panel 1, as Drive 1
- STK-Timberline connected at address */dev/rmt/tps3d4*, located physically in LSM 2, Panel 1, as Drive 3
- The IBM-3494 on host ursa has the following drive, connected to the host ursa
- IBM-3590 connected at address */dev/rmt/tps4d6*, identified by the IBM Library Server as drive 0

Host vega is an OpenVault client host with 1 library: STK-9714 connected at device address */dev/scsi/sc7d110*. The STK-9714 has the following drives, all of which are connected to the host:

- DLT7000 connected at address */dev/rmt/tps7d3*, located physically as the bottom drive in the library.
- DLT7000 connected at address */dev/rmt/tps7d2*, located physically as the top drive in the library.

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 it.

Custom Installation

If you have not already installed OpenVault and would like to do a custom install, you must first determine the type of library and drive on each host that you plan to have OpenVault manage for you.

Use the *hinv* command to determine connected SCSI libraries; use the output of *hinv* with the *scsicontrol* command to determine the type of each attached SCSI library, as follows:

```
# /sbin/hinv|grep Juke| \
awk '{printf "/dev/scsi/sc%d%d10\n", $7, $3}' | \
xargs /usr/sbin/scsicontrol -i|grep Juke
/dev/scsi/sc2d310: Jukebox      STK      9730      1300
/dev/scsi/sc3d110: Jukebox      STK      9710      1805
```

For each Jukebox that *hinv* displays, issue a *scsicontrol* command like the one shown below. In this command, X is the controller number from the *hinv* output and Y is the unit number from the *hinv* output.

```
# scsicontrol -i /dev/scsi/scXdY10
```

Each line in the *scsicontrol* output shows the library's control patch, the library vendor name, the library product name and firmware revision number of the library.

From the above example, we see that the following libraries are present:

vendor	product	revision
STK	9730	1300
STK	9710	1805

Remember to repeat the process explained above for every host on which you have a SCSI library attached to, that you plan to control using OpenVault.

Determining Attached SCSI Drives

Use the *hinv* command to determine connected SCSI drives, and use the output of *hinv* with the *scsicontrol* command to determine the type of each attached SCSI drives, as follows:

```
# /sbin/hinv|grep Tape| \
awk '{printf "/dev/scsi/sc%d%d10\n", $8, $4}' | \
xargs /usr/sbin/scsicontrol -i|grep Tape
/dev/scsi/sc2d110: Tape          QUANTUM DLT7000          1E46
/dev/scsi/sc2d210: Tape          QUANTUM DLT7000          1E46
/dev/scsi/sc3d210: Tape          QUANTUM DLT7000          1E46
/dev/scsi/sc3d310: Tape          QUANTUM DLT7000          1E46
/dev/scsi/sc4d610: Tape          IBM      03590B1A          53F2
```

Notice that for each drive that *hinv* displays, we issue a *scsicontrol* command of 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/scXdY10
```

Each line in the *scsicontrol* output shows the drive’s control patch, the drive vendor name, the drive product name and firmware revision number of the drive

In the above example we see that we have the following drives:

vendor	product	revision
QUANTUM	DLT7000	1E46
IBM	03590B1A	53F2

Remember to repeat the process explained above for every host on which you have a SCSI drive attached to, that you plan to control using OpenVault.

Non SCSI-Attached Libraries

This version of OpenVault also supports the following non-SCSI libraries EMASS-Grau robotic library controlled through the DAS interface IBM-3494 robotic library controlled through the IBM’s library interface Any Storage Tek robotic libraries controlled through the ACSLS interface If you plan to configure OpenVault to manage one or more of these libraries, you must install the appropriate ‘OpenVault.lcp’ subsystems.

For each instance of these libraries you plan to configure with OpenVault you configure one and exactly one LCP to control the device. Since all of the above-mentioned libraries are not directly attached via SCSI, you chose any host in the OpenVault system as the host on which to run the LCP. You would then need to install the appropriate *OpenVault.lcp* subsystem on the identified hosts.

For simplification, it may be useful to install and configure LCPs non SCSI-attached libraries on the OpenVault server host machine.

Note: If you install the LCP for a non SCSI-attached LCP on an OpenVault host unintentionally, the OpenVault configuration script will ask if you would like to configure that LCP. Choose no to answer this question.

Other Guidelines for Custom Installation

Here are guidelines on selecting OpenVault components for installation:

- For OpenVault server hosts
 - Required:

OpenVault.sw.config	OpenVault configuration scripts
OpenVault.sw.core	OpenVault core servers
OpenVault.sw.admin	OpenVault administrative tools
OpenVault.sw.libs	OpenVault core runtime libraries
 - Recommended:

OpenVault.man.manpages	OpenVault manual pages
OpenVault.man.relnotes	OpenVault release notes
OpenVault.docs.adminguide	OpenVault administrative 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
OpenVault.docs.architecture	OpenVault architecture specifications
OpenVault.docs.developer	OpenVault LCP/DCP/app developer guides
 - As needed

OpenVault.dcp.libs	OpenVault run-time libraries for DCPs
OpenVault.dcp.XXXX	Appropriate DCP subsystem
OpenVault.lcp.libs	OpenVault run-time libraries for LCPs
OpenVault.lcp.XXXX	Appropriate LCP subsystem

- For OpenVault client hosts
 - Required:

OpenVault.sw.config	OpenVault configuration scripts
OpenVault.sw.libs	OpenVault core runtime libraries
 - Recommended:

OpenVault.man.manpages	OpenVault manual pages
OpenVault.man.relnotes	OpenVault release notes
OpenVault.sw.admin	OpenVault administrative tools
OpenVault.docs.adminguide	OpenVault administrative 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
OpenVault.docs.architecture	OpenVault architecture specifications
OpenVault.docs.developer	OpenVault LCP/DCP/app developer guides
 - As needed

OpenVault.dcp.libs	OpenVault run-time libraries for DCPs
OpenVault.dcp.XXXX	Appropriate DCP subsystem
OpenVault.lcp.libs	OpenVault run-time libraries for LCPs
OpenVault.lcp.XXXX	Appropriate LCP subsystem
 - Not recommended

OpenVault.sw.core	OpenVault core servers
-------------------	------------------------

The `OpenVault.dcp.libs` and `OpenVault.sw.libs` subsystems are required on each OpenVault host that contains an OpenVault DCP. The install tool that you use will enforce this requirement.

The `OpenVault.lcp.libs` and `OpenVault.sw.libs` subsystems are required on each OpenVault host that contains an OpenVault LCP. The install tool that you use will enforce this requirement.

The `OpenVault.sw.libs` subsystem is required on each OpenVault host that contains an application OpenVault LCP. The install tool that you use will enforce this requirement.

Configuration Roadmap

The OpenVault initial configuration script, *setup*, automatically performs the steps shown in Table 2-1. The details of each step are explained in the sections that follow.

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 step is required and 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 drives and libraries attached to remote OpenVault client hosts.	If you have remote libraries and drives, perform this step on the server host.
6. 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.
7. Configure drives attached to OpenVault client hosts.	If you have drives attached to remote OpenVault client hosts, perform this step on the remote host(s).
8. Configure libraries attached to OpenVault client hosts.	If you have libraries attached to remote OpenVault client hosts, perform this step on the remote host(s).
9. Import media into OpenVault.	After configuring all libraries, perform this step on the OpenVault server host.

Note: Not all steps may be relevant for your specific scenario.

Preparing OpenVault Devices and Hosts

The procedure that follows explains how to prepare the devices that OpenVault will manage for the OpenVault configuration process. Each drive that you plan to configure must be housed in a library that will be managed by OpenVault. OpenVault cannot manage stand-alone drives (that is, drives that are not housed in a robotic library).

In the procedure that follows, step 4 is required only if you are configuring IBM-3494 libraries.

1. Cable all SCSI drives to appropriate hosts. Ensure proper termination where required.
2. Cable all SCSI robots to appropriate hosts.
3. For each OpenVault host that has an OpenVault controlled drive attached to it, as *root* do the following:

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

Caution: The *mediad* daemon interferes with the operation of OpenVault, so it is essential that you prevent *mediad* from accessing OpenVault drives. Refer to the *mediad* man page on how to prevent *mediad* from accessing these drives.

4. Prepare the IBM-3494 libraries that you are planning to configure:
 - Enable the LCP host to communicate with the IBM-3494 server. The IBM-3494 library server runs on the PS/2 machine 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 IBM-3494 LCP. Refer to the IBM-3494 Library server documentation for details on how to do this.
 - On the OpenVault host that will run the IBM-3494 LCP, prepare the */etc/ibmatl.conf* file. Briefly, this file contains three fields:

```
<libraryname <PS/2 hostname <LCPhostname
```

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

If this file does not exist, you must create it.

Caution: It is critical that you remember this name. This is also going to the OpenVault name for this library.

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


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

5. Prepare the DAS PS/2 machine for a EMASS-Grau library to configure

Configuration Worksheet

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

Fill out Table 2-2 with OpenVault server client host information.

Table 2-2 Host Worksheet

OpenVault server host	OpenVault client host(s) [if any]

Fill out Table 2-3 for drives in a SCSI library.

Table 2-3 Drive Worksheet (SCSI)

Drive Device Path	Hostname drive is connected to	OpenVault name for drive	Library housed in	Description of drive's physical location (SCSI)

Fill out Table 2-4 for drives in an IBM-3494 library.

Table 2-4 Drive Worksheet (IBM-3494)

Drive Device Path	Hostname drive is connected to	OpenVault name for drive	Library housed in	Drive #

Fill out Table 2-5 for drives in an ACSLS library.

Table 2-5 Drive Worksheet (ACSL)

Drive Device Path	Hostname drive is connected to	OpenVault name for drive	Library housed in	LSM ID	Panel ID	Drive #

Fill out Table 2-6 for a SCSI library (use one worksheet per SCSI library).

Table 2-6 Library Worksheet (SCSI)

Library Device Path	Hostname library is attached to	OpenVault name for library

Description of drive's physical location within library	OpenVault name for this drive

Fill out Table 2-7 for an IBM-3494 library (use one worksheet per IBM-3494 library).

Table 2-7 Library Worksheet (IBM-3494)

Hostname or IP address of PS/2 controlling the 3494	OpenVault name for library

Drive # as described by mtlib command	OpenVault name for this drive

Fill out Table 2-8 for an ACSLS library (use one worksheet per ACSLS library).

Table 2-8 Library Worksheet (ACSLs)

ACSLs server host name	ACS ID	ACSAPI packet version	Hostname that LCP will run on	OpenVault name for library

Description of drive's physical location within library			OpenVault name for this drive
LSM ID	Panel ID	Drive #	

Completing the Worksheet

Follow the order outlined below to record information on the worksheet.

Acquiring Hostnames for OpenVault Hosts

You need to identify all those hosts that are going to comprise your OpenVault system. Identify only one host as the OpenVault server host.

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

Generating Unique Names for Libraries

Generate unique names for all libraries that are going to be managed by OpenVault. You may choose any naming scheme that best suits your site, as long as the names remain unique. If you prepared an IBM-3494 as described earlier, you would have had to generate a library name. For IBM-3494, you would have to use the name generated above.

Collecting Information for Attached Drives

Determine which drives are attached to each host machine using the *hinv* command:

```
# hinv | grep Tape
```

From our example configuration, we see the following drives on the hosts in question, *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:

1. Determine which OpenVault host the drive is physically cabled to. Record this information in the drive work sheet.

You may choose to physically follow the SCSI cables connected to the drive in order to determine this.

2. Determine the device path. Use the 'hinv' command as shown above from the host that the drive is connected to determine this.

It may be a good idea to insert a tape in the drive, unload all other drives that are on the same SCSI bus as this drive and issue a *mt* status command.

For example from our example configuration from Figure 2-2, if we were determining this information for one of the two STK-SD3 (STK-Redwood) drive connected to vega, we would insert a cartridge into drive in question, unload the other drives on SCSI controller 3 and issue an `mt -f <devicepath status` command. Here you could systematically try out both the `/dev/rmt/tps3d1` and `/dev/rmt/tps3d3` SCSI path.

By repeating this process and by elimination, we arrive at the desired information.

3. Record the name of the library that the device is physically housed in.
4. Generate a unique name for this drive and record this name.
5. Record the physical location of drive the library For examples, in SCSI libraries it is typical to find drives laid out 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”, “third drive from left” etc.,

Typically Storage Tek’s SCSI libraries order drives from the bottom, whereas Exabyte’s SCSI libraries order drives from the top, whereas some SpectrLogic libraries order drive left to right.

6. For drives in an ACSLS controlled robot, query ACSLS.

A physical description is not appropriate. You will 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
Identifier  State      Status      Volume      Drive Status
              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
ACSSA
```

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

7. For IBM-3494 robust drives, query the IBM-3494 server.

As with drives in an ACSLS controlled library, for drives in an IBM-3494 robot, a physical description is not appropriate. You will need to query IBM-3494 server for more information. A summary of how you can do this is shown below. Refer to the IBM-3494 documentation for details.

- Prepare the OpenVault host that will run the LCP for the IBM-3494 robot in question as described in section, “Collecting Information for IBM-3494 Libraries” on page 31.

As *root*, enter these commands:

```
# cd /usr/OpenVault/clients/lcp/IBM-3494
# ./mtlib -l <libname> -D
```

Here <libname> is the name that you generated earlier when you prepared the */etc/ibmatl.conf* file. From the example configuration listed in Figure 2-1.

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

The first field is the Drive #. Refer to the IBM-3494 documentation on how to map the drive number to a physical location.

- Record the drive number and physical location in the appropriate worksheets.
- [XXX Add info on how to determine locality info for EMASS-Grau library]

Collecting Information for SCSI-Attached Libraries

Use the procedure that follows to collect information for SCSI-attached libraries.

1. Record the device control path.

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:

```
# /sbin/hinv|grep Juke| \
awk '{printf "/dev/scsi/sc%dd%d10\n", $7, $3}'| \
xargs /usr/sbin/scsicontrol -i|grep Juke
/dev/scsi/sc2d310: Jukebox      STK      9730      1300
/dev/scsi/sc3d110: Jukebox      STK      9710      1805
```

We see above, that for each Jukebox that *hinv* displays, we issue a *scsicontrol* command of 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 instruction for every host with a SCSI library that will be controlled using OpenVault.
3. Gather information about contained drives.

For each SCSI library, as root, on the OpenVault LCP host:

```
# cd /usr/OpenVault/clients/lcp/<lcpctype>
```

Here <lcpctype> is the type of the LCP for example, the STK-9700.

```
# ./LCP* getlibinfo <lcpcontrolpath> | grep DRIVE
```

Here <lcpcontrolpath> is the */dev/scsi* control path as described in the step above. In our example local-only configuration in Figure 2-1 on *ursa*, we find the following for the STK-9730 library:

```
ursa 16# cd /usr/OpenVault/clients/lcp/STK-9700
ursa 17# ./LCP* getlibinfo /dev/scsi/sc2d310 | grep DRIVE
DRIVE 1030 - - first_drive_from_BOTTOM
DRIVE 1031 - - second_drive_from_BOTTOM
```

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

Collecting Information for ACSLS Libraries

Use the procedure below to collect information for ACSLS libraries.

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 that is running on the ACSLS server. For example if you are running ACSLS version 5.1 then use '4' as the packet version.

4. Record the drive information.

The method of determining drive information is described "Collecting Information for Attached Drives" on page 27.

Collecting Information for IBM-3494 Libraries

Use the procedure below to collect information for IBM-3494 libraries.

1. Record the hostname of the PS/2 system that is running the IBM-3494 server.
2. The method of determining drive information “Collecting Information for Attached Drives” on page 27 described earlier. Record the drive information.

Collecting Information for EMASS-Grau Libraries

[XXX add information]

Planning Cartridge and Drive Groups

The introductory chapter introduced the notion of Cartridge Groups and Drive Groups. The OpenVault configuration script initially creates one Cartridge Group, named carts and one Drive Group named, drives.

All media that you import via the initial configuration procedure, are by default, introduced into the carts Cartridge Group. If you would like to create more cartridge groups, so that you may import different media into different cartridge groups, you may do so by following the instructions provided in the imaginary “[Re-configuration & Tuning OpenVault](#)” chapter. However, be sure to not accept the “Import Media” option that is offered as part of initial configuration. The Import Media function is available as part of reconfiguration procedures. After creating the desired cartridge groups you may import media.

Note: When adding new cartridge groups, remember to enable existing and new applications to use those cartridge groups.

Similarly, all drives configured during the initial configuration procedure are added to 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 the instructions provided in the imaginary “[Re-configuration & Tuning OpenVault](#)” chapter.

Note: When adding new drive groups, remember to enable existing and new applications to use those drive groups.

Media considerations

[XXX fill this in]

Selecting a Password

If your specific site require security, it is recommended that you choose and remember a password. You will be asked to enter a password when you initially configure OpenVault. This password applies to the authentication of all applications, drives and libraries that you configure during the initial configuration process. If you would like individual libraries, drives, or applications to use passwords other than the default password that you selected during initial configuration, you may do so. Refer to the instructions provided in the imaginary *“Re-configuring & Tuning OpenVault”* chapter.

Install the OpenVault Flexlm license

The OpenVault license you receive includes instructions on how to install it. Typically this involves entering the information listed in the license sheet into the */var/flexlm/license.dat* file.

Library and Drive Names

The OpenVault configuration procedure requires you to identify and provide OpenVault names for libraries and drives. Each library and drive must have a unique OpenVault name. The configuration script requires you to enter these names in one or more places. It is very important that you use the OpenVault drive and library names consistently and correctly. The configuration worksheets can assist you in doing this.

Library Names

If you have a library connected to a remote OpenVault Client host, the OpenVault configuration script prompts for OpenVault name for the same physical library in the following places:

- When you configure the OpenVault Server host, you will need to enable this library by providing the OpenVault name for this library. At that time you have the choice of accepting a name generated by the configuration script or have the choice of entering a name you selected.

- When you actually configure the library on the OpenVault client host, you are asked for the OpenVault name of the library that you entered when you enabled this library on the OpenVault Server host.

In both cases, be sure to enter the exact OpenVault library names in both places.

If you have a library connected to the OpenVault Server host, at the time of configuring this library you will be asked to provide the OpenVault name of this library. The configuration script also offers a generated name as the default. You may override this default by entering a name you have selected.

Drive Names

If you have a drive connected to a remote OpenVault Client host, the OpenVault configuration script prompts for OpenVault name for the same physical drive in the following places:

- When you configure the OpenVault Server host, you will need to enable this drive by providing the OpenVault name for this drive. At that time you have the choice of accepting a name generated by the configuration script or have the choice of entering a name you selected.
- When you actually configure the drive on the OpenVault client host, you are asked for the OpenVault name of the drive that you entered when you enabled this drive on the OpenVault Server host.

In both cases, be sure to enter the exact OpenVault library names in both places.

If you have a drive connected to the OpenVault Server host, at the time of configuring this drive you will be asked to provide the OpenVault name of this drive. At this time, the configuration script also offers a generated name as the default. You may override this default by entering a name you have selected.

Regardless of where the drive is connected, at the time of configuring the library in which the drive is housed in you are asked to enter the OpenVault name of the drive.

Note: It is possible that one or more drives are connected to remote OpenVault clients but housed in a library connected to the OpenVault server host. In this case, the configuration of this library involves entering the drives names. At this point the configuration script does not offer a default. If you have selected names in the worksheet for this library then enter those names. If you have not recorded drive names in your worksheet, then you are required to create new names. Be sure to record the names and enter them exactly at later points in the configuration process.

In all three places, be sure to enter the exact OpenVault name of the drive.

Server Configuration

This section describes the steps that you would need to go through in order to configure an OpenVault Server. We will follow the roadmap presented in “Configuration Roadmap” on page 21. Obviously, depending on the specific configuration at your site, some questions may not apply.

As described in the roadmap in “Configuration Roadmap” on page 21, you will need to configure an OpenVault server first. In order to do so:

1. Login to the designated OpenVault server as *root*
2. Ensure that the OpenVault license is installed in the */var/flexlm/license.dat* file.
3. Go to the OpenVault install directory, usually
`cd /usr/OpenVault`
4. Execute the *./setup* script.

The *setup* script is the main OpenVault configuration script. This script is designed to drive the configuration process through the roadmap described in “Configuration Roadmap” on page 21.

In general, the *setup* script, configures OpenVault based upon installed software and hardware and your input to various questions. If you do not see the choice you are looking for, double check your installation to make sure the items are installed.

When prompted, and where possible, the setup script presents defaults for most prompts which are indicated by *[value]*. The default may be chosen at such prompts simply by pressing the Enter key. Help text may be obtained by entering *`?*` at many prompts.

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

This prompt is designed to inform the setup script whether it is being executed on an OpenVault host. Choose the default when running on the designated OpenVault host.

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

The default port number that the OpenVault server listens for is 44444. Choose the default if this is acceptable. If this port number is currently in use by another application

or if you prefer to use another port number, enter that number. Care must be taken to ensure that no other application is using that port number on this host.

Note: If you choose to use a different port number and you have OpenVault client host (that is, remote hosts) then you will be asked for the portnumber when you configure those hosts. At that time it is imperative that you enter the same value that you entered at the time of configuration of the Server. Failure to do so will prevent successful configuration of the client host.

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

By default, the OpenVault server and LCPs/DCPs and applications are configured without a security key. The security key is the mechanism by which the OpenVault server authenticates a new connection from a client (an LCP, DCP, or application). The default security key presented in this prompt is no security. If this is acceptable choose the default. If you have already selected a security key (password) then enter this value.

You may choose to configure OpenVault initially with default security and in order to simplify the initial configuration. This will enable you to get OpenVault up and running. Then at a later stage, you may modify OpenVault security by following the guidelines presented in the imaginary [“Re-Configuration & Tuning OpenVault” chapter](#).

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

Configuring OpenVault Core on the OpenVault Server

At this point the setup script prompts you with the following option:

```
OpenVault Configuration Menu
Automatic Configuration Option
    31 - Autoconfigure OpenVault on this host
        (setup OpenVault Server)
        (setup OpenVault Devices)
    q - Exit.
Which operation would you like to do: [31]
```

Select the default if you wish to continue with autoconfiguration, which allows you to configure the OpenVault server.

"Would you like to set up an OpenVault server on this system? [Yes]"

Select this option if this is the designated OpenVault server. If you selected the default option of setting up an OpenVault server on, then you will see the following messages displayed, indicated successful configuration of the OpenVault core.

```
Waiting for OpenVault to initialize ...
The OpenVault server was successfully started.
Created Application: ov_umsh
created cartridge group: carts
Cartridge-group-application creation:
    Application: ov_umsh
    Group: carts
created drive group: drives
Drive-group-application creation:
    Application: ov_umsh
    Group: drives
The OV server 'ursa' is UP.
The OpenVault server was successfully set up on this system.
```

At this point the following items have been configured:

- An OpenVault core including the catalog and an *ovroot* process that is ready to service connections from OpenVault clients.
- A single cartridge group named 'carts'
- A single drive group named 'drives'
- The user mounting application *ov_umsh*(1M) has been enabled to for executing on the OpenVault server host, that is, this machine.
- The user mounting application *ov_umsh* is eligible to use cartridges entered into the default cartridge group, 'carts'.
- The user mounting application *ov_umsh* is eligible to use drives entered into the default drive group, 'drives'.

The setup script then allows you to configure locally attached drives and libraries.

"Would you like to configure local LCPs and DCPs? [Yes]"

Answer yes if you are ready to proceed with configuration of locally attached drives and libraries.

Note: Configure drives and libraries at this time.

Configure Locally Attached Drives on the OpenVault Server

The OpenVault configuration script scans the hardware to determine which SCSI Tape drives are attached to this host. If any drives are present, it then proceeds to offer each drive for configuration.

```
Do you want to configure the <dtype> drive at <addr> as "tape1"? [Yes]"
```

This prompt offers you the choice of configuring the drive of type, <dtype>, for example, <dtype> could be a DLT7000, at address <addr>, for example, */dev/rmt/tps2d1*.

The configuration script proposes a name for the drive, "tapeX" as shown in the prompt above. Revert to the DRIVE worksheets that you have created in "Preparing OpenVault Devices and Hosts" on page 22. Search for the entry with the "Drive device pathname" equal to the <addr> shown above, and the "Hostname drive is connected to" field equal to the hostname of this machine.

Answer no for the following conditions:

- If you do not find a matching entry or do not want to configure this drive at all at this time
- If you wish to configure this drive at a later time.
- If you find a matching entry and you have chosen a name for this drive, (look under the column "OpenVault name for drive") and would like to use the name chosen in your worksheet.

If you find a matching entry and you have not chosen a name or wish to use the default name presented then press Enter or type yes.

If you answered no to the above prompt, the setup script allows you to enter a name of your choice or completely bypass configuring this drive.

If you answered no,

```
"Enter the name for the <dtype> drive at <addr>, or Enter to bypass  
this device []"
```

Note: To bypass configuration of this drive, press Enter.

If you wish to configure this device, enter the name from the entry you found in the DRIVE worksheet. It is very important that you type in the exact name listed in your worksheet.

If you chose to configure the drive, the configuration script proceeds to configure the drive and will display the output similar to the text shown below:

```
Creating drive 9730-bottom-dlt in drive group drives
Created Drive: 9730-bottom-dlt
Updating core keys file for the new drive
Configuring DLT7000 at /dev/rmt/tps2d1 to be "9730-bottom-dlt"
Starting the DCP for drive 9730-bottom-dlt
```

At this point, the following items are complete:

- The named drive entry has been added to the OpenVault catalog.
- An appropriate authentication entry has been added to the `/usr/OpenVault/core_keys` file.
- The drive's configuration file has been created (`/usr/OpenVault/dcp/<dtype>/<dtype>.config`, where `<dtype>` is the chosen name for this drive).
- The drive has been added to the drive group `'drives'`

Note: If you wish to move this drive to an alternative drive group follow the instructions provided in the imaginary ["Re-Configuring & Tuning OpenVault" chapter](#).

The configured DCP is automatically started.

The process described in this sub-section is repeated for every drive that is connected to the system. If you choose to add or delete drives at a later time you can configure or deconfigure such drives by following the instructions provided in the imaginary ["Re-Configuring & Tuning OpenVault" chapter](#).

Configuring Locally-Attached Libraries on OpenVault Servers

The OpenVault configuration script now examines the attached hardware for libraries and presents each library for configuration.

Configuring SCSI- Attached Libraries

```
"Do you want to configure the <ltype> library at <laddr> as "lib1"?
[Yes]"
```

This prompt offers you the choice of configuring the SCSI library of type, `<dtype>`, for example, `<type>` could be a STK-9730, at address `<laddr>`, for example, `/dev/scsi/sc2d310`.

The configuration script proposes a name for the SCSI library, “libX” as shown in the prompt above. Revert to the SCSI library worksheets that you have created in section 2.5. Search for the entry with the “Library device pathname” equal to the <laddr> shown above, and the “Hostname Library is connected to” field equal to the hostname of this machine.

Answer no under the following conditions:

- If you do not find a matching entry or do not want to configure this SCSI library at all at this time.
- If you wish to configure this SCSI library at a later time.
- If you find a matching entry and you have chosen a name for this SCSI library, (look under the column “OpenVault name for library”), and would like to use the name chosen in your worksheet.

If you find a matching entry and you have not chosen a name or wish to use the default name presented then press Enter or enter yes.

If you answered no to the above prompt, the setup script allows you to enter a name of your choice or completely bypass configuring this SCSI library.

If you answered no, you see this prompt:

```
"Enter the name for the <ltype> library at <laddr>, or Enter to bypass  
this device []"
```

If you wish to bypass configuration of this SCSI library, then type Enter.

If you wish to configure this device, then enter the name from the entry you found in the SCSI library worksheet. It is very important that you type in the exact name listed in your worksheet.

If you chose to configure the SCSI library, the configuration script proceeds to configure the drives contained in this SCSI library. The configuration script queries the device to determine information about the drives housed in this library.

Note: This action of querying the device to determine information about drives contained in this SCSI library might take a while, especially if the machine has just been rebooted or if the device has been power-cycled or if the main door to the library has just been closed.

At this point you will see output similar to the following, where <lname> is the name of the SCSI library that you chose, <laddr> is the SCSI library's device path, and <ltype> is the type of the SCSI library.

```
Creating library <lname>
Created Library: <lname>
Updating core keys file for the new library
Configuring <ltype> at <laddr> to be <lname>
```

Upon querying the information about contained drives, the setup script will ask you to enter the OpenVault name for each drives contained in this library, one after the other.

```
"For the drive at location <dlocation>
Enter a drive name for the element address <daddr>: "
```

Here <dlocation> is an English description of the location of the drive being prompted for. <daddr> is the drive's "slot" address, that is, the slot number that the SCSI robot uses to address the drive.

Here you will need to enter the OpenVault name for the drive at the given physical location within the SCSI library. Refer to the SCSI library worksheet for this SCSI library to determine the OpenVault name of this drive. You will need to match the description string <dlocation> with the column labelled "Description of drive's physical location within the library". Although your description may not syntactically match the description provided by the configuration script, choose the one that effectively has the same meaning.

Caution: It is critical to the proper functioning of OpenVault that you match the drive correctly with your worksheet and that you enter the OpenVault name of the drive. If you had not previously recorded a name on your worksheet, enter the name of the drive that you had previously chosen.

If you are running setup in a scrolling window, it is recommended that you scroll back to the point in the setup script where you configured this drive and use the name that you entered there.

The <daddr, that is, the drive's slot address is offered as an additional guide to the expert user. If you are very familiar with the SCSI libraries addressing scheme you may choose to use this as an unambiguous reference to the drive's physical location of the drive within the library.

If you do not want OpenVault to manage this drive then press Enter. The configuration script will bypass configuring this drive as one being contained in the library. OpenVault will not use this drive to perform mounts.

Note: If this drive is connected to a remote OpenVault client machine, then choose the name from your worksheet, as you have not yet configured this drive. If you have not yet recorded a name for this drive, then select a unique drive name and enter it at the prompt. Also be sure to record the name in the worksheet as you will be required to enter this name when you configure drives on the remote OpenVault client machine that the drive is attached to.

The configuration script will prompt for OpenVault names for each drive that is housed in this SCSI library.

After you have entered names of all the drives housed in this SCSI library (or bypassed) you will see output similar to the following:

```
OpenVault Server host name:      ursa
OpenVault Server port number:   44444
Library name:                   9710
LCP name:                       9710_lcp
Security key:                   none
LCP polling interval:          30
Library control path            /dev/scsi/sc3d110
Drives in the Library:
                                Name      Address
                                9710-bottom-dlt  1030
                                9710-top-dlt    1031
```

You are then prompted for confirmation:

```
"Create it now (Y|N)? [ ] "
```

If you are satisfied with the SCSI library configuration summary shown above, then enter 'Y'. If you would like to change one or more parameters, then enter 'N' and you will be prompted once again for the OpenVault names for each drive.

Note: The correct configuration script will NOT allow you to change the name of the library at this time. You may however do so at a later time. Refer to the imaginary ["Re-Configuring & Tuning OpenVault" chapter](#) for details on how to do this.

Upon successful confirmation, the configuration script proceeds to configure the SCSI library:

```
LCP successfully created
```

```
Press enter to continue...
Starting the LCP for library 9710
```

At this point, the following items are complete:

- The named SCSI library entry has been added to the OpenVault catalog.
- An appropriate authentication entry was added to the `/usr/OpenVault/core_keys` file.
- The SCSI library's configuration file has been created (`/usr/OpenVault/lcp/<lname>/config`, where `<lname>` is the chosen OpenVault name for this SCSI library).
- The configured LCP is automatically started.

Configuring an IBM-3494 Library

If you have the software for the IBM-3494 LCP installed on an OpenVault host, then the configuration script offers the option of configuring this library, regardless of whether you plan to configure one on the this host or not.

```
"Do you want to configure the IBM-3494 library at - as "lib1"? [Yes]"
```

This prompt offers you the choice of configuring an IBM-3494 library with name lib1.

The configuration script proposes a name for the library, "libX" as shown in the prompt above. Revert to the IBM-3494 library worksheets that you have created in "Preparing OpenVault Devices and Hosts" on page 22. If you have multiple IBM-3494 library that you need to configure, then select one.

Note: If you do not want to configure this library at all at this time, answer no.

If you have chosen a name for this library, (look under the column "OpenVault name for library"), and would like to use the name chosen in your worksheet, answer no.

If you have not chosen a name or wish to use the default name presented then press Enter or type yes.

Note: If this library is being configured on an OpenVault client, then you will not be prompted with a default library name; instead you will be asked to enter the name.

If you answered no to the above prompt, the setup script allows you to enter a name of your choice or completely bypass configuring this library.

```
"Enter the name for the IBM-3494 library at -, or <enter> to bypass  
this device []
```

Note: If you wish to bypass configuration of this library, then press Enter.

If you wish to configure this device, then enter the name from the entry you found in the library worksheet. It is very important that you type in the exact name listed in your worksheet.

If you chose to configure the library, the configuration script proceeds to configure the drives contained in this library. At this point you will see output similar to the following:

```
Creating library ibm3494  
Created library ibm3494  
Updating core keys file for he new library  
Configuring IBM-3494 at - to be "ibm3494"
```

At this point you will be prompted for the host name 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"
```

The configuration script then queries the device to determine information about the drives housed in this library.

Note: This action of querying the device to determine information about drives contained in this library might take a while, especially if the machine has just been rebooted or if the device has been power-cycled or if the main door to the library has just been closed.

At this point you will see output similar to the following:

```
waiting for lmcpd to become active ...
```

Upon querying the information about contained drives, the setup script will ask you to enter the OpenVault name for each drives contained 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. This is 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 of this drive. You will need to match the drive address string <daddr> with the column labelled "Drive # number as described by the mtlb command". Refer to the section "Collecting Information for Attached Drives" on page 27 for details on how to run the mtlb command.

Caution: It is extremely critical to the proper functioning of OpenVault that you match the drive correctly with your worksheet and that you enter the OpenVault name of the drive. If you had not previously recorded a name on your worksheet, enter the name of the drive that you had previously chosen.

If you are running setup in a scrolling window, scroll back to the point in the setup script where you configured this drive and use the name that you entered there.

Note: If you do not want OpenVault to manage this drive then press Enter. The configuration script will bypass configuring this drive as one being contained in the library. OpenVault will not use this drive to perform mounts.

If this drive is connected to a remote OpenVault client machine, then choose the name from your worksheet as you have not yet configured this drive. If you have not yet recorded a name for this drive, then select a unique drive name and enter it at the prompt. Also be sure to record the name in the worksheet as you will be required to enter this name when you configure drives on the remote OpenVault client machine that the drive is attached to.

The configuration script will prompt for OpenVault names for each drive that is housed in this library. After you have entered names of all the drives housed in this library (or bypassed) you will see output similar to the following:

```
OpenVault Server host name:    ursa
OpenVault Server port number:  44444
Library name:                  ibm3494
LCP name:                      ibm3494_lcp
TCP/IP Address of the Library: tsm-ps2
Security key:                   none
LCP polling interval:          30
Drives in the Library:         Name    Address
                               3590    0
```

You are then prompted for confirmation:

```
"Create it now (Y|N)?  []"
```

If you are satisfied with the library configuration summary shown above, then enter Y. If you would like to change one or more parameters, then enter N and you will be prompted once again for the OpenVault names for each drive.

Note: The correct configuration script will NOT allow you to change the name of the library at this time. You may however do so at a later time. Refer to the imaginary *“Re-Configuring & Tuning OpenVault”* chapter for details on how to do this.

Upon successful confirmation, the configuration script proceeds to configure the library:

```
waiting 30 seconds for lmcpd to become active ...
LCP successfully created
Press enter to continue...
Starting the LCP for library ibm3494
```

At this point, the following items are complete:

- If this Library was configured on the OpenVault server host, then the named library entry has been added to the OpenVault catalog.
- If this library was configured on the OpenVault Server host, then an appropriate authentication entry has been added to the */usr/OpenVault/core_keys* file.
- The SCSI library’s configuration file has been created, that is, */usr/OpenVault/lcp/<lname>/config*, where *<lname>* is the chosen OpenVault name for this library.
- The configured LCP is automatically started.

Configuring an EMASS-Grau library

[XXX fill this in]

Configuring a StorageTek ACSLS library

If you have the software for the StorageTek ACSLS LCP installed on an OpenVault host, then the configuration script offers the option of configuring this library, regardless of whether you plan to configure one on the this host or not.

```
"Do you want to configure the STK-ACSLs library at - as "lib1"? [Yes]"
```

This prompt offers you the choice of configuring an STK-ACSLs library with name lib1.

The configuration script proposes a name for the library, "libX" as shown in the prompt above. Revert to the STK-ACSLs library worksheets that you have created in "Preparing OpenVault Devices and Hosts" on page 22. If you have multiple STK-ACSLs libraries on this host that you need to configure, then select the worksheet one that you have not yet configured.

If you do not want to configure this library at all at this time, answer no.

If you have chosen a name for this library, (look under the column "OpenVault name for library"), and would like to use the name chosen in your worksheet, answer no.

If you have not chosen a name or wish to use the default name presented then press Enter or type yes.

Note: If this library is being configured on an OpenVault client, then you will not be prompted with a default library name, instead you will be asked to enter the name.

If you answered no to the above prompt, the setup script allows you to enter a name of your choice or completely bypass configuring this library.

```
"Enter the name for the STK-ACSLs library at -, or <enter> to bypass
this device []
```

Note: If you wish to bypass configuration of this library, then press Enter.

If you wish to configure this device, then enter the name from the entry you found in the library worksheet. It is very important that you type in the name exactly as it is listed in your worksheet.

If you chose to configure the library, the configuration script proceeds to configure the drives contained in this library. At this point, you will see output similar to the following:

```
Creating library wolfcreek
Created library wolfcreek
Updating core keys file for he new library
Configuring STK-ACSLs at - to be "wolfcreek"
```

At this point you will be prompted for the host name for the ACSLS Server for this library:

```
"What is the hostname of the ACSLS server? []"
```

Enter the version number of the ACSLS interface. You will need to identify the version of ACSLS that is installed on the ACSLS server subtract 1 from this number and enter the value for this prompt. Refer to the “Collecting Information for ACSLS Libraries” on page 30 for details on how to do this.

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

Enter the ID of the ACS that you are using for this LCP. Refer to the documentation for this library on obtaining the ACS ID.

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

The configuration script then queries the device to determine information about the drives housed in this library.

Note: This action of querying the device to determine information about drives contained in this library might take a while, especially if the machine has just been rebooted or if the device has been power-cycled or if the main door to the library has just been closed.

At this point you will see output similar to the following:

```
ONC RPC: csi_init(): Initiation Started
ONC RPC: csi_init(): Initiation Completed
Acquiring drive information from library (may take a while) ..
ONC RPC: csi_init(): Termination Started
ONC RPC: csi_init(): Termination Completed
```

Upon querying the information about contained drives, the setup script will ask you to enter the OpenVault name for each drives contained in this library, one after the other.

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

The location string displayed is the LSM ID, Panel ID, and Drive number for the drive. “Collecting Information for Attached Drives” on page 27 describes how to obtain information for drives in an ACSLS library. Refer to the ACSLS library worksheet for this library and select the drive entry with this LSM ID, Panel ID and Drive number. Enter the OpenVault name that you selected.

Caution: It is critical to the proper functioning of OpenVault that you match the drive correctly with your worksheet and enter the OpenVault name of the drive. If you had not previously recorded a name on your worksheet, enter the name of the drive that you had previously chosen.

If you are running setup in a scrolling window, it is recommended that you scroll back to the point in the setup script where you configured this drive and use the name that you entered there.

If you do not want OpenVault to manage this drive then press Enter. The configuration script will bypass configuring this drive as one being contained in the library. OpenVault will not use this drive to perform mounts.

Note: If this drive is connected to a remote OpenVault client host, choose the name from your worksheet, as you have not yet configured this drive. If you have not yet recorded a name for this drive, select a unique drive name and enter it at the prompt. Be sure to record the name in the worksheet; you will be required to enter this name when you configure drives on the remote OpenVault client host that the drive is attached to.

The configuration script prompts for OpenVault names for each drive that is housed in this library.

After you have entered (or bypassed) the names of all drives housed in this library, you will see output similar to the following:

```
OpenVault Server host name:    ursa
OpenVault Server port number:  44444
Library name:                  wolfcreek
LCP name:                      wolfcreek_lcp
Security key:                  none
ACSL Server Host name:        tsm-sun
ACSL Server Version:          4
ACSL ACS ID:                  0
LCP polling interval:         30
Drives in the Library:
```

Name	LSM	Panel	Drive
redwood1	2	3	1
redwood3	2	3	3
timberline1	2	1	1
timberline3	2	1	3

You are then prompted for confirmation:

```
"Create it now (Y|N)?  [ ] "
```

If you are satisfied with the library configuration summary shown above, enter 'Y'. If you would like to change one or more parameters, then enter 'N' and you will be prompted once again for the OpenVault names for each drive.

Note: The current configuration script will NOT allow you to change the name of the library at this time. You may however do so at a later time. Refer to the imaginary [“Re-Configuring & Tuning OpenVault” chapter](#) for details on how to do this.

Upon successful confirmation, the configuration script proceeds to configure the library:

```
LCP successfully created
Press enter to continue...
Starting the LCP for library ibm3494
```

At this point, the following items are complete:

- If this Library was configured on the OpenVault server host, then the named library entry has been added to the OpenVault catalog.
- If this library was configured on the OpenVault Server host, then an appropriate authentication entry has been added to the `/usr/OpenVault/core_keys` file.
- The SCSI library's configuration file has been created, that is, `/usr/OpenVault/lcp/<lname>/config`, where `<lname>` is the chosen OpenVault name for this library.
- The configured LCP is automatically started.

Enabling Remote Administration, LCPs and DCPs

The next step in the configuration roadmap allows you to enable the following features: administration of the OpenVault server from remote OpenVault clients; LCPs, to connect to the OpenVault server from remote OpenVault clients; and DCPs, to connect to the OpenVault server from remote OpenVault clients

Note: This step enables remote clients to connect to OpenVault. You must run the configuration script on the remote OpenVault host to configure libraries and drives on that host.

```
"Will remote OpenVault admin/LCP/DCP clients access this server? [No]"
```

If you plan to enable one or more of options a, b or c and enter yes; otherwise, accept the default. If you plan to add remote libraries, drives, administrative commands, or applications at a later time, refer to the instructions provided in the imaginary [“Re-Configuring & Tuning OpenVault” chapter](#).

The configuration script prompts for the name of a remote OpenVault client host, which is an OpenVault client that meets one or more of the following tests:

- You plan to connect libraries to that host
- You plan to connect drives to that host
- You plan to administer the OpenVault server from that host and installed the OpenVault administrative software subsystem.
- You plan to have applications running on this host that request services from the OpenVault server.

Refer to the Host worksheet that you generated in “Preparing OpenVault Devices and Hosts” on page 22 to identify remote client hostnames.

Enter the name of a remote OpenVault client host at the following prompt:

```
“Enter a remote OpenVault client machine name, or Enter to continue:
[ ]”
```

If you have already entered all the hosts, press Enter to continue to the next step in configuration. Otherwise, enter the name of the remote client host.

Enable remote applications

At this point the configuration script displays the following prompt:

```
“Will the machine vega require administrative access to this server?
[Yes]”
```

If you would like to administer the OpenVault server from the named host, accept the default; otherwise enter no”.

Enabling Remote Libraries

The configuration script then prompts for remote libraries.

```
“Are any libraries controlled by LCP’s on vega? [No]”
```

If you plan to have libraries connected to the named remote host, enter yes; otherwise accept the default.

If you answered yes to the previous question, the configuration script asks for OpenVault names of libraries on this host

```
"Do you want to configure a library on vega as "lib1"? [Yes]"
```

The configuration presents a library name, but if you have not chosen library names in the configuration worksheet, you can choose the offered name or select a new unique name.

It is very important that you record the name you select; you will be asked to enter this name when configuring the library on the remote machine. Refer to the Library worksheets that you have created in section 2.5 for library names and remote host names.

The configuration script continues to prompt for remote OpenVault libraries names. Press Enter when you have no more remote library names to enter.

Enabling Remote Drives

At this point, the configuration script asks for remote drives in a similar fashion:

```
"Are any drives controlled by DCP's on vega? [No]"
```

If you plan to have drives connected to the named remote host, then enter yes; otherwise, accept the default.

If you answered yes to the previous question, the configuration script asks for OpenVault names of drives on this host:

```
"Do you want to configure a drive on vega as "drive1"? [Yes]"
```

The configuration presents an OpenVault drive name. If you have not chosen drive names in the configuration worksheet, choose the offered name or select a new unique name.

Note: It is very important that you record the OpenVault drive name you select as you will be asked to enter this name when configuring the drive on the remote machine.

Refer to the Drive worksheets that you have created in "Preparing OpenVault Devices and Hosts" on page 22 for OpenVault drive names and remote host names.

The configuration script continues prompts for more remote drive names. Press Enter when you have no more remote OpenVault drive names to enter.

Now that you have configured locally attached drives and libraries, the configuration script prompts you if you would like to import the tapes that were discovered in the

newly configured libraries. If you have more libraries on remote OpenVault client hosts, you may want to wait for those libraries to be configured. If so, you may invoke the import function as described in the section “Importing Media” on page 56. If you do not have any remote libraries, then you may chose to import media right away.

Client Configuration

After configuring an OpenVault server, you are now ready to configure remote OpenVault clients. You will need to configure each remote OpenVault client in order.

Refer to the Hosts worksheet that you generated in “Preparing OpenVault Devices and Hosts” on page 22 for the remote OpenVault client hosts that you need to configure.

To configure an OpenVault client host, complete the steps listed below.

1. Login to the remote host as *root*
2. Change to the OpenVault directory:

```
# cd /usr/OpenVault
```

3. Start the setup script:

```
# ./setup
```

The setup script presents the general configuration strategy described below.

OpenVault Configuration

The general strategy for setting up OpenVault is to

- 1) configure the OpenVault server
- 2) configure LCP/DCP's on the server machine
- 3) configure server for local Applications
- 4) if needed, configure server for remote LCP's, DCP's, and Apps
- 5) if needed, install and configure LCP/DCP's on remote machines
- 6) from the server, for each library setup/import media

Where possible, defaults for each prompt are indicated by [value].

Help text may be obtained by entering `?' at many prompts.

Some menus will present only the available options depending upon the software, hardware, or options that are installed.

If you do not see the choice you are looking for, double check your installation to make sure the items are installed.

The setup 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.

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

Enter the name of the OpenVault server host. The default presented in this prompt is the host name of the machine that you are running configuration script on.

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

If you chose a different port number when you configured the OpenVault server, enter that port number, otherwise accept the default.

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

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.

Note: Specifying the exact values for the OpenVault server's host name, port number, and security key is critical for proper functioning of all the OpenVault clients on this host.

The configuration script then provides the menu options described below.

```
OpenVault Configuration Menu
  Configuration for Machines Running LCPs
    11 - Configure a new client LCP
  Configuration for Machines Running DCPs
    21 - Configure a new client DCP
  Automatic Configuration Option
    31 - Autoconfigure OpenVault on this host
  (setup OpenVault Devices)
    q - Exit.
Which operation would you like to do: [31]
```

Note: When configuring an OpenVault client host for the first time it is highly recommended that you select the Autoconfigure option (option 31). Selecting other options will allow you to configure LCPs and/or DCPs, however, those options are more manual in nature, at this time.

The Autoconfigure option allows you to configure the drives and libraries attached to this host.

```
"Would you like to configure local LCPs and DCPs? [Yes]"
```

If you are ready to configure the libraries and drives on this host, accept the default. If you proceed with autoconfiguring LCPs and DCPs on this host, configure all the libraries and drives that you planning to configure on this host

Configure Attached Drives on OpenVault Client Hosts

If you have any drives attached to this host, the configuration script will prompt for each attached drive in succession. The set of dialogs presented are similar to the dialog when you configure drives on an OpenVault server. The main difference is that the configuration script asks you to enter the OpenVault name for the drive and does not offer a default name.

Caution: It is critical that you enter the Openvault name for each drive exactly as you entered when you enabled this remote drive while configuring the OpenVault server. Refer to the Drive worksheet that you generated previously.

A sample dialog is provided below:

```
Enter the name for the STK-redwood drive at /dev/rmt/tps3d1,  
or <enter> to bypass this device [] redwood1  
Configuring STK-redwood at /dev/rmt/tps3d1 to be "redwood1"  
Starting the DCP for drive redwood1
```

Note: If you prefer not to configure a drive at this time, then type Enter at the prompt.

At this point, the offered drive has been configured and the DCP for this drive is started. Use the *ov_stat* command on the server to determine the status of the newly configured drive.

Configuring Attached Libraries on OpenVault Client Hosts

The configuration script will scan the installed hardware and software to determine which libraries are available for configuration. Each library that it detects is offered for configuration.

The questions asked by this script are similar to those presented when you configure a library on the OpenVault server. These questions are described earlier.

The difference between configuring a library on an OpenVault server and configuring a library on an OpenVault client is that the configuration script asks you to enter the OpenVault name of the library and does not offer a default name.

Caution: It is critical that you enter the OpenVault name for each library exactly as you entered when you enabled this remote drive while configuring the OpenVault server. Refer to the appropriate library worksheet that you generated previously.

A sample dialog for a SCSI-attached library is shown below.

```
Enter the name for the STK-9714 library at /dev/scsi/sc7d110, or
<enter to bypass this device [] 9714
Configuring STK-9700 at /dev/scsi/sc7d110 to be "9714"
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 Creation Parameters Confirmation
OpenVault Server host name:      ursa
OpenVault Server port number:    44444
Library name:                    9714
LCP name:                        9714_lcp
Security key:                    none
LCP polling interval:           30
Library control path             /dev/scsi/sc7d110
Drives in the Library:          Name      Address
                               9714-bottom-dlt 1030
                               9714-top-dlt   1031

Create it now (Y|N)? [] Y
LCP successfully created
Press enter to continue...
Starting the LCP for library 9714
```

Note: If you decide that you do not want to configure an offered library at this time, you may press Enter to bypass configuring this library.

You will be prompted for OpenVault names for the drives contained in the library. Be sure to enter the names exactly as you did when you enabled those drives on the OpenVault server. Refer to your Library worksheet for details. Again, you will need to match the location description string for each contained drive with the description string that you recorded in the appropriate Library worksheet.

If you have drives contained in this library that are connected to a different host, enter the exact OpenVault name for these drives. Do this even if you have not yet configured the drives on that host. Refer to the Library worksheet for this library to obtain 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.

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

Importing Media

After you have configured the libraries on the OpenVault server and ALL OpenVault clients, you may need to import media to make it available for applications to use. Importing media is the act of letting the OpenVault server know about each piece of media. See Chapter 3, "Cartridge Life Cycle" for more information.

Each tape that you would like applications to use must be imported before it is made available for allocations and mounts.

Caution: If you have media in a library that contain data or if there are media that are known to applications, take precautions so other applications may not accidentally access such media. Refer to the application's documentation on how to import media known to that application into OpenVault. **Failure to do so may lead to data loss.**

You may want to invoke the import media function of the configuration script after you have configured all libraries, including libraries on remote Openvault client hosts.

The import function is available only on the OpenVault server host. To import media in a library, you must identify a cartridge group that each cartridge is going to belong to. The configuration script will attempt to import all media found in a library into the same cartridge group. If you prefer to import media from the same library into more than one cartridge group, follow the manual procedure in the imaginary "Re-Configuration & Tuning" chapter.

By default, the only available cartridge group is the 'carts' group. If you would like to add more cartridge group, it is best that you do so now. Refer to the imaginary "Re-Configuration & Tuning" chapter for details on how to do this.

You must also identify a cartridge type for each cartridge. This release supports the following cartridge types:

```
STKredwood & STKtimberline
3590, 3490E, 3490, & 3480
DLT7000 & DLT2000
```

Note: At this time, the configuration script will assume that all media in a given library is of the same cartridge type. If this is not the case, you may skip auto import of media from this library and follow the manual procedure described elsewhere.

The import function allows you to pre-allocate all media to an application. If you wish to import media so that it is pre-allocated to more than one application, then you may skip auto import of media from this library and follow the manual procedure described in the imaginary [“Re-Configuration & Tuning” chapter](#).

You may import media so that the media is not pre-allocated. If you do, make sure the applications that use this instance of OpenVault know how to import media. For example, the sample user mounting application *ov_umsh* provided with OpenVault knows how to allocate media.

To use the import function, enter these commands as *root* on the OpenVault server host, and select the “Import Media” option, as shown below:

```
# cd /usr/OpenVault
# ./setup
```

```
OpenVault Configuration Menu
      Configuration for the OpenVault Server Machine
        1 - Manage Libraries, Drives and Applications
        2 - Manage Cartridge Groups And Drive Groups
        3 - Import Media
        4 - Enable remote applications
      Configuration for Machines Running LCPs
        11 - Configure a new client LCP
        12 - Modify an existing client LCP
        13 - Deconfigure an existing client LCP
      Configuration for Machines Running DCPs
        21 - Configure a new client DCP
        22 - Modify an existing client DCP
        23 - Deconfigure an existing client DCP
        q - Exit.
```

```
Which operation would you like to do: 3
```

```
Would you like to import media ? [Yes]
```

At this point, the configuration script checks for all configured libraries that contain media that must be imported; then it offers each of these libraries individually.

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

Select the default if you are ready to import media from the named library. If not enter no.

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

If you decide to import all cartridges in this library in the same cartridge group, accept the default. Otherwise, type Enter so the configuration script will skip importing media from this library.

The available cartridge types are presented:

1. D2-L
2. D2-M
3. D2-S
4. DDS3
5. DDS2
6. DDS1
7. STKredwood *
8. STKtimberline *
9. 3590 *
10. 3490E *
11. 3490 *
12. 3480 *
13. 8mm-160m
14. mammoth
15. DLT7000 *
16. DLT2000 *
17. DTF
99. MIXED_MEDIA

```
"Select the Cartridge type for the cartridges in this library:"
```

Note: The supported cartridge types are denoted with a '*'.

The available cartridge groups are displayed. Enter the cartridge group that you would like all cartridges in this library to be part of.

```
Cartridge Groups available are:
```

1. carts

```
Select the Cartridge group for the cartridges in this library:
```

If you want to pre-allocate cartridges to an application, accept the default at the following prompt:

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

If yes, then, you are prompted for an application name.

```
"Enter the name of the application that you like to pre-allocate the  
cartridges to [ov_umsh]" dmf
```

Note: If the application name that you enter, does not exist, the configuration script will create it for you.

When the import complete 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
```

If you do not wish the pre-allocate then enter no at the following prompt:

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

Cartridges in this library are then imported without being pre-allocated to any application. Upon successful completion, you see output similar to the following:

```
Importing tapes with:  
    Cartridge Type = DLT7000  
    Cartridge Group = carts  
This may take a while ...  
Finished importing media from library, 9730
```

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 reference 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 it ceases to be a part of that system. The major events in the life of a cartridge include:

- the 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

Cartridge States

During the life cycle shown in Figure 3-1, 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 where data are 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 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.

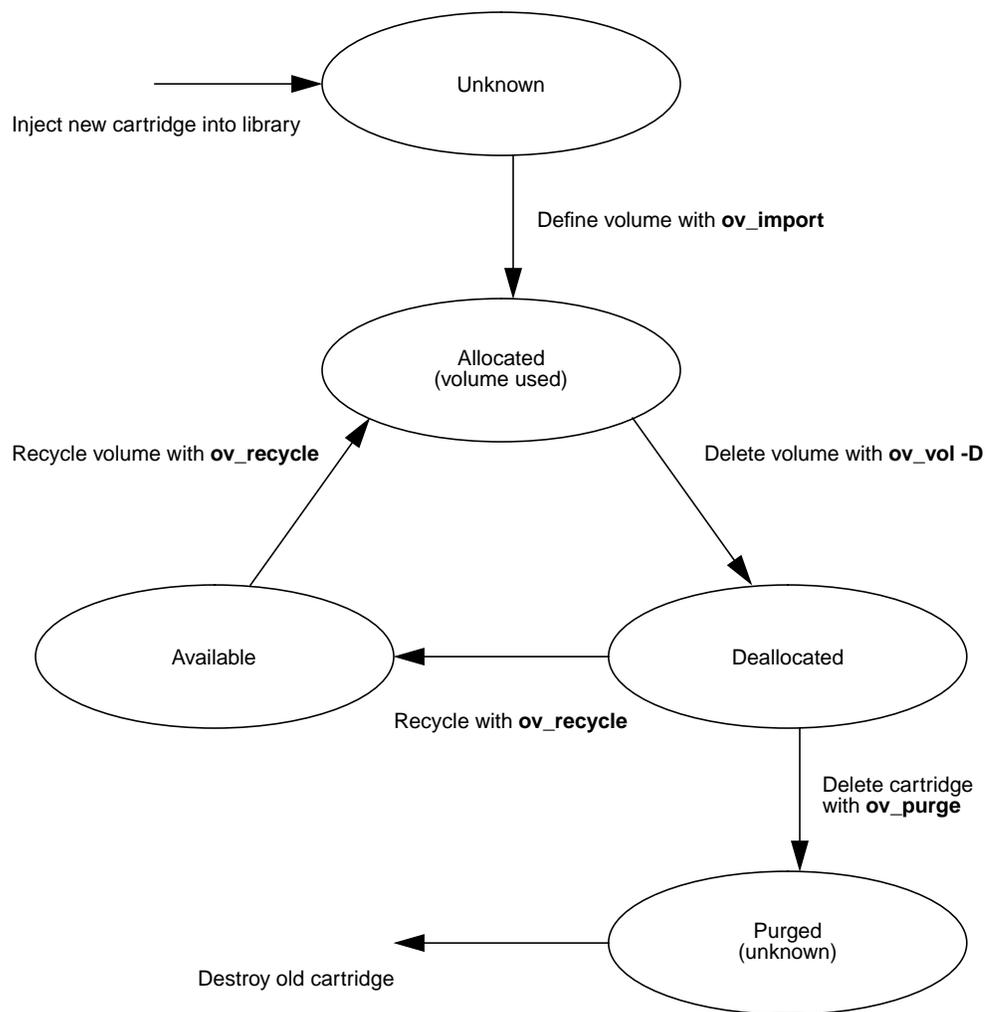


Figure 3-1 Cartridge Life Cycle (Simplified)

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.

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.

A second method is to use the inject/eject slot of the library if it is so equipped. 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.

Examining the Contents of a Library

It is often useful to see what a library believes to be contained within it. The command *ov_stat* performs this function (among others), displaying the contents of the slots in a library. A typical command would be:

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

The `-s` option specifies that a slotmap should be displayed, and the `-L 9730` indicates that it should be for the library named "9730." The output of this command looks like this:

```
# 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         DLT        false
9730     slot 1      DLT         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 shows a library with 7 slots occupied, with the PCL listed for the cartridge in each occupied slot. The PCL info 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.

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 data about the cartridge, partitions, and volumes on the cartridge (using *ov_cart*, *ov_part*, and *ov_vol*), and a tool that enters several of these data at once (*ov_import*). 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 will be discussed later.

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 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 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 data follows:

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.

Displaying Cartridge Data

Once data on a cartridge is entered into the system it can also be displayed. The `ov_lscarts` command is used for this purpose. Here are two examples of the use of this command. The first shows 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 info. 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. Detailed information on cartridge(s) can be shown as follows:

```
# ov_lscarts -li ABN695
PCL      cart type  owner  state  cartID          part  volume
ABN695   DLT7000         ok     wIS2MTXrMmQA1YR
```

In this example, the `-l` option specifies a long listing, and the `-i` specifies that the CartridgeID be printed also. Note that there is no partition or volume information, because we have not entered that data yet. Creating a partition is the next step.

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. Silicon Graphics tape driver software typically does not support multiple partitions per cartridge, so one is currently the limit for tapes on OpenVault. Partitions are created with the *ov_part* administrative command:

```
# ov_part -n wIS2MTXrMmQAALYR -s SideA -p Part1 -b DLT7000
New partition created:
  cartridge ID = 'wIS2MTXrMmQAALYR', partition Name = 'Part1'
```

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 this 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 be 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.

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 might be able to allocate the cartridge before we can type a new command to prevent it. The allocatable status can be set with the “-a” option when we create the partition as follows:

```
# ov_part -n wIS2MTXrMmQAALYR -s SideA -p Part1 -b DLT7000 -a false
```

This command is identical to the `ov_part` command above 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 below.

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.

The administrative command to create a volume is `ov_vol`. The following command creates a volume named “VolOne” that will belong to an application named `ov_umsh` on the cartridge defined in the examples above:

```
# ov_vol -n VolOne -a ov_umsh -c wIS2MTXrMmQAALYR -s SideA -p Part1
```

The `-n` option specifies 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 “Part1”. The output of this command is as follows:

```
New volume created:
  volume name = 'VolOne', application name = 'ov_umsh'
```

Applications may also create volumes by issuing an *allocate* request to OpenVault through the CAPI interface. OpenVault will look for an allocatable partition on a cartridge that is either not owned by any application or is owned by the requesting application. It will then create volume data for one such partition, mark the partition as not allocatable (because there is now a volume allocated on it), and set the cartridge ownership to the requesting application. See the *OpenVault Application Programmer's Guide* for further details.

Displaying Volume Data

Information on volumes can be displayed with the *ov_lsvols* command. This works similarly to the *ov_lscarts* command described above:

```
# ov_lsvols -l VolOne
volume      cartID          owner           partition
VolOne      wIS2MTXrMmQAALYR  ov_umsh        Part1
```

As for *ov_lscarts*, 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.

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 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.

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 wIS2MTXrMmQAALYR -s SideA -p Part1 -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 -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.

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 outlined above of using *ov_cart*, *ov_part*, and *ov_vol*. 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 this example, four cartridges are created with *ov_import*:

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

ov_import 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. The results can be seen with this *ov_lscarts* command:

```
# 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 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*.

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 ABN695
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.

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.

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. Here is an example of the use of this command:

```
# ov_purge -C wIS2MTXrMmQAALYR
Are you sure you want to purge all information for cartridge with
cartridge ID = wIS2MTXrMmQAALYR (Y/N)? y
Deleted partition Part1
Deleted cartridge wIS2MTXrMmQAALYR
```

After this command, no data remain in the OpenVault system about this 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:

- “Before You Start” on page 75
- “Installing OpenVault” on page 76
- “Setting Up OpenVault” on page 76
- “Configuring OpenVault” on page 84
- “Monitoring OpenVault” on page 91

Before You Start

Before installation, you must determine the extent of the OpenVault domain of control. First determine the best location for each OpenVault server and select the clients that you want for OpenVault. The domain of control is usually determined by physical location (building, floor) or administrative division (department, project). Be sure to choose a server that is highly reliable and accessible. Choose clients based on your need for the robotics libraries or drives that they contain.

Many installations attach a library and all its drives to the same machine used for the OpenVault server, so this machine is also the LCP and DCP client. Some installations use three different machines for the OpenVault server, LCP client, and DCP client.

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.

Before you install OpenVault, check that you have the necessary LCP and DCP software components to interface with all libraries and drives. Some LCP and DCP software may support a range of models, so verify before beginning installation.

Installing OpenVault

Use the *inst* command (or comparable install program) to load the OpenVault software on each server and client in the OpenVault environment. For more information about IRIX install programs, see *inst(1M)* and *swmgr(1M)*. For last minute product changes, refer to the OpenVault release notes.

You must install *OpenVault.sw.core* and *OpenVault.sw.libs* on the OpenVault server machine. For ease of administration, install *OpenVault.sw.admin* and *OpenVault.sw.user* everywhere. Install *OpenVault.*.libs* and the appropriate LCP software on each client machine attached to an OpenVault-controlled library. Install *OpenVault.*.libs* and the appropriate DCP on each client machine attached to an OpenVault-controlled drive. Depending on cabling, installation could involve one or more machines.

Removing OpenVault

To remove OpenVault software, or individual components, use the *versions* command; see *versions(1M)*. The following command removes all OpenVault software:

```
# versions remove OpenVault.*.*
```

Setting Up OpenVault

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 *setup* 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:

- 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.
- 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).
- security key A password for authorizing client or application access to OpenVault.

default access path

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

passthrough driver

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

drive handles Directory for creating *attach* nodes for applications, often */tmp/mlm*.

OpenVault configuration files are described in detail by the following sections:

- “Setting Up the Server” on page 77
- “Setting Up Clients” on page 81
- “Setting Up Applications” on page 82
- “Setting Up Security” on page 82

Setting Up the Server

To configure OpenVault, login to the OpenVault server machine as superuser and run *setup*, a menu-based interface that helps you along the way:

```
# /usr/OpenVault/setup
[explanatory messages appear]
What is the name of the OpenVault Server? [localhost]
What is the port number is OpenVault using? [44444]
What default security key would you like to use? [none]
```

Accept the defaults for server name (current hostname) and port number. If you accept the default authorization key *none*, OpenVault runs without security. If you specify a security key, the *setup* script creates a key authorization file on the OpenVault server. The setup of key authorization is described in the section “Setting Up Security” on page 82. The setup script continues:

```
Automatic Configuration Option
 31 - Autoconfigure OpenVault on this host
      (setup OpenVault Server)
      (setup OpenVault Devices)
q - Exit.
```

To autoconfigure the OpenVault server and any attached drives and libraries, accept choice 31. When *setup* asks to set up the OpenVault server on this system, answer yes. Messages like the following appear:

```
Waiting for OpenVault to initialize ...
Created Application: ov_umsh
created cartridge group: carts
Cartridge-group-application creation:
    Application: ov_umsh
    Group: carts
created drive group: drives
Drive-group-application creation:
    Application: ov_umsh
    Group: drives
The OV server 'localhost' is UP.
```

During the next phase, *setup* inventories hardware and autoconfigures attached drives and DCPs for them. Answer yes to all these questions:

```
Configure the DLT7000 drive at /dev/rmt/tps2d1 as "tape1"? [Yes]
...
Configure the DLT7000 drive at /dev/rmt/tps2d2 as "tape2"? [Yes]
```

During the next phase, *setup* autoconfigures attached libraries and LCPs for them. Unless you have a multiconnected library, or want to customize the library name, answer yes to such questions:

```
Configure the STK-9730 library at /dev/scsi/sc2d310 as "lib1"? [Yes]
Creating library lib1
Updating core keys file for the new library
Configuring STK-9700 at /dev/scsi/sc2d310 to be "lib1"
```

Now comes the tricky part. During cabling, you must have recorded which SCSI ports on the OpenVault server system were connected to which drives in the library. There is no way for OpenVault to make this correlation: you have to say how drives are cabled. OpenVault tries to give you hints by querying the library for information:

```
For the drive at location "first drive from top",
    Enter a drive name for the element address 1030: tape2
For the drive at location "second drive from top",
    Enter a drive name for the element address 1031: tape1
```

In this example, **tape2** was cabled from SCSI port *tps2d2* to the topmost drive, whereas **tape1** was cabled from SCSI port *tps2d1* to the second drive down.

After this, the *setup* script prints library information and asks if you want to create the LCP. Answer yes, and the LCP starts up. If there are other attached libraries, *setup* continues by asking you for configuration information about them. Some libraries may not be intended for use with OpenVault; when asked to configure these, answer no.

For the IBM 3494 library, OpenVault starts up the *lmcpd* daemon. For ACSLS controlled libraries, OpenVault activates *ssi* to acquire drive information from the library. For EMASS libraries, OpenVault starts up DAS. None of this absolves you of responsibility for telling OpenVault which drive names are associated with which drive in a library.

Importing Media

During the next phase, the *setup* script asks if you want to import media into the library (that is, make OpenVault aware of the media). Assuming the library contains cartridges, it is easiest to import media now, although you can do it later by running *setup* again.

```
Would you like import media from the library lib1 [yes]
Do you have tapes in library `lib1`, that DMF(TM) knows about [yes] no
```

OpenVault will get confused if the library already contains DMF-related media.

The *setup* script continues by asking about cartridge groups and cartridge types. Unless you want to create multiple cartridge groups for multiple applications, which can be done later, answer yes to the first question. Provide the cartridge media type after the second question.

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

```
Cartridge types available are:
```

```
...
7.          STKredwood
8.          STKtimberline
9.          3590
10.         3490E
11.         3490
12.         3480
...
15.         DLT7000
16.         DLT2000
...
99. MIXED_MEDIA
```

```
Select the Cartridge type for the cartridges in this library: 15
```

Next, *setup* informs you of its default cartridge group “carts” and asks if you want to preallocate these cartridges to an application, and if so, the name of the application.

```
Cartridge Groups available are:
```

```
1. carts
```

```
Select the Cartridge group for the cartridges in this library: 1
```

```
Would you like to pre-allocate cartridges to an application [yes]
Enter the name of that application for pre-allocation [] dmf
```

In this example, all cartridges are allocated to the DMF application, *dmf*.

```
Importing tapes with :
    Cartridge Type = DLT7000
    Cartridge Group = carts
    Application = dmf
This may take a while ...
Finished importing media from library, lib1
```

Configuring After Setup

After it has cycled through all the libraries importing media, the *setup* script deposits you in the configuration menu, where you can further administer media, cartridge groups, drive groups, libraries, drives, and applications. You can quit at this point.

```
OpenVault Configuration Menu

Configuration for the OpenVault Server Machine
  1 - Manage Libraries, Drives and Applications
  2 - Manage Cartridge Groups And Drive Groups
  3 - Import Media

Configuration for Machines Running LCPs
  11 - Configure a new client LCP
  12 - Modify an existing client LCP
  13 - Deconfigure an existing client LCP

Configuration for Machines Running DCPs
  21 - Configure a new client DCP
  22 - Modify an existing client DCP
  23 - Deconfigure an existing client DCP

q - Exit.

Which operation would you like to do: q
```

Example 4-1 OpenVault Configuration Menu

If any drives or libraries are connected to other systems besides the OpenVault server on which you just ran *setup*, you need to run *setup* on these client systems as well.

Setting Up Clients

You must run OpenVault software on all systems where drives and robotic libraries are attached and under OpenVault control, in particular on systems not operating as the OpenVault server. These non-server systems are referred to as OpenVault clients. Each type of drive and library needs its own software installed and configured, as described in “Setting Up Drives” and “Setting Up Robotic Libraries,” respectively.

If you want to implement security, an authorization key must be created, as described in “Setting Up Security” on page 82. Security keys are part of server, library, and drive configuration files—they are not in a separate file by themselves.

Setting Up Drives

Each drive needs its own drive control program (DCP) and configuration file. The drive-specific DCP needs to be installed with the OpenVault installation procedure; see “Installing OpenVault” on page 76 for more information.

For DCPs that are provided by Silicon Graphics, DCP installation produces an interactive *setup* script that requests the configuration information that is necessary to create the configuration file. For DCPs supplied by other vendors, consult their documentation.

The *setup* script asks the same questions it did when you autoconfigured the OpenVault server, except limited to information needed for drive and DCP configuration.

Setting Up Robotic Libraries

Each robotic library needs its own library control program (LCP) and configuration file. The library-specific LCP should have been installed with the OpenVault installation (see “Installing OpenVault” on page 76 for more information).

For LCPs that are supplied by Silicon Graphics, LCP installation produces an interactive *setup* script that requests the configuration information that is necessary to create the configuration file. For LCPs supplied by other vendors, consult their documentation.

The *setup* script asks the same questions it did when you autoconfigured the OpenVault server, except limited to information needed for library and LCP configuration.

Setting Up Applications

The *setup* 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 “Setting Up Application Security” on page 86.

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*, that is, the client and server share the secret password. The server has one key file (*/usr/OpenVault/var/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 that contains the servers it knows about. Example 4-2 shows an example of a key file that can be used on either a client or a server.

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

Example 4-2 Client and Server Key Authorization File

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. For example, 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.
Instance	The instance name of the application. If more than one version is running, use asterisk (*) as a wildcard for multiple instances of the same 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.

Any field in the key file can contain an asterisk "*" to specify all conditions. For example, if a server uses a * in the Host column, all OpenVault clients can connect with the server. If a client uses a * in the Host column, all OpenVault servers can connect with this client. If a client uses a * in the Key column, it indicates no password, and OpenVault runs with security disabled.

Note: Use asterisks carefully; too many asterisks in this file make the OpenVault system vulnerable to security violations. Also ensure that wildcard entries appear at the bottom of the file, after non-wildcard entries.

On the server host, the shared secret key is stored in `/usr/OpenVault/var/core_keys`.

Setting Up Non-Robotic Libraries

Note: This is not supported in the OpenVault 1.x releases.

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.

Setting Up Offsite Libraries

Note: This is not supported in the OpenVault 1.x releases.

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 “Moving Cartridges Within Libraries” on page 99). The chosen library must contain a drive that is compatible with the cartridge type.

Configuring OpenVault

Once OpenVault has been set up, as described in “Setting Up OpenVault” on page 76, you can start to configure it to meet your storage management needs. Some basic configuration tasks are:

- “Setting Logging Levels” on page 84
- “Registering Applications” on page 85
- “Unregistering Applications” on page 85
- “Setting Up Application Security” on page 86
- “Enabling Application Access to a Drive” on page 86
- “Managing Cartridge Groups” on page 87
- “Introducing Cartridges” on page 89

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.

Setting Logging Levels

You can configure the degree of system status reporting and message logging that you want to have. You can also specify a different name for the message log file. By default, system message logging sends all error messages to the file `/usr/OpenVault/var/OVLOG`.

An example follows to set the “information” log level. See `ov_msg(1M)` for details.

- To set logging level to the information level:

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

See “Accessing the OpenVault Message Log” on page 113 for message log information.

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. See `ov_app(1M)` for a description of options.

Unless an application is registered, it cannot connect to the OpenVault server.

An authorization key, or password, can be added when registering an application to ensure secure transactions; see “Setting Up Application Security” on page 86.

To register an application without using security, follow these steps:

1. Check to make sure that the application is not already registered.

```
ov_app -l '.*'
```

A listing appears of applications that are registered with the default OpenVault server. To check on other hosts, use the `-H host1, host2` option. If you don't see your application named, continue with step 2.

2. Register the application (*appName*) with the default server.

```
ov_app -c appName
```

Unregistering Applications

Unregistering an application from the OpenVault system means the application can no longer connect to OpenVault.

- To unregister the application called *networker* (IRIX NetWorker):

```
ov_app -d networker
```

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 4-2, 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 the section “Setting Up Applications” on page 82 for details on setting up his file.

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). You can enable a drive for a specific application, allowing the application to run on the drive. This might be necessary if the application uses special features of the drive.

1. To enable a drive for the *tar* application, use the following command:

```
ov_drive -c Herbie targroup false
```

This command creates the drive named Herbie in the group *targroup*. The *false* argument indicates that the drive is not disabled for group Herbie.

2. Create the drive group *tardrive* and add the *tar* application with a prime precedence rating **-p 1** (lower number drives are used first).

```
ov_drivegroup -c tardrive -A tar -p 1
```

Now the *tar* command tends to use drives in this drive group. A **-p precedence** value organizes all accessible drive groups into a preference list for the application.

Drive groups allow a set of drives to be specified for a particular use, most likely to be available for a particular application. You can set up drive groups by repeating the `ov_drivegroup -c` command for each drive group you want, allocating applications to each. Adding a `-p precedence` flag when you specify the application allows you to make certain applications more or less likely to use drives in that drive group.

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. 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 “Preventative Maintenance” on page 102). The cartridge groups in the figure are Unallocated, Void, and Degauss:

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.
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 OpenVault catalog entry for the cartridge is missing.
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.

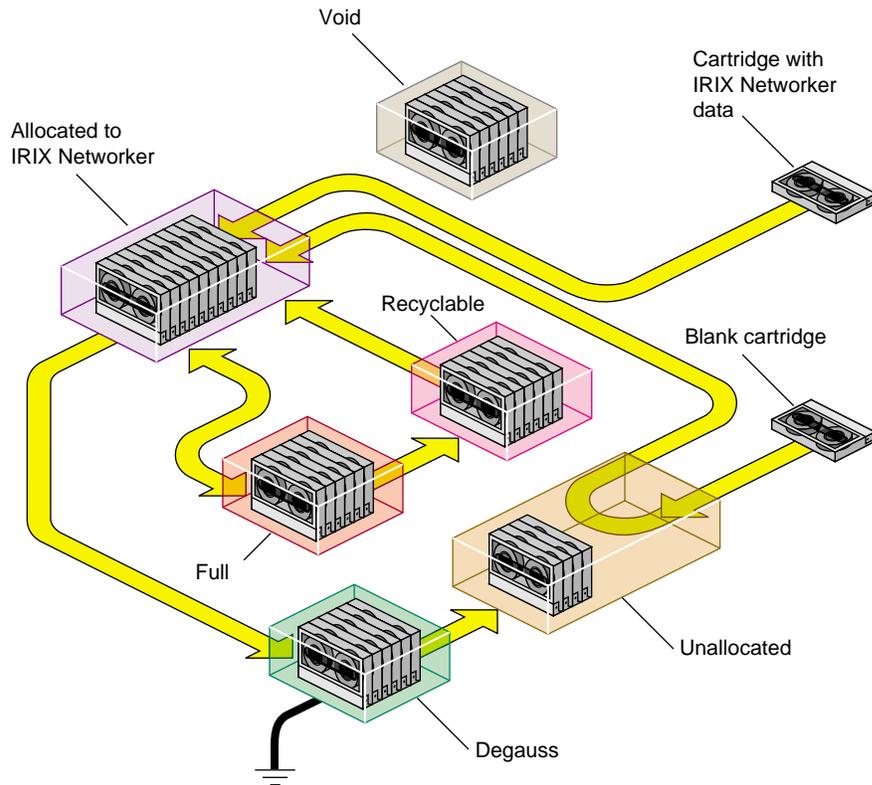


Figure 4-1 Example Cartridge Group (Engineering)

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.

Use the following procedure to create a cartridge group.

1. Be sure you have registered the application that will manage this group. See “Registering Applications” on page 85 for details.
2. Determine whether to assign drives to the application. If you decide to do so, see “Enabling Application Access to a Drive” on page 86 for details.
3. Create the cartridge group and assign its access permission. For example, the following command creates the “NWbackup” group with high priority (**-p 1**) and permits only the IRIX Networker application access to its cartridges (**-q 1**).

```
ov_cartgroup -c NWbackup -p 1 -A networker -q 1
```
4. Introduce cartridges into the cartridge group. Continue with the next section, “Introducing Cartridges” on page 89, for details.
5. If necessary, move cartridges between groups or within the library. Examples of when this might be necessary are when a cartridge
 - is no longer needed by an application—see “Removing Cartridges From Libraries” on page 99
 - can be moved back into the Unallocated group—see “Recycling Cartridges” on page 100
 - needs to be sent to the Degauss group for erasure—see “Destroying Cartridges” on page 100
 - needs to be moved within its own group—see “Moving Cartridges Within Libraries” on page 99

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:

- 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 these steps:

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 **-n** *label* gives the PCL affixed to the cartridge. Also, assign the cartridge to a cartridge group (**-p** *cartridgeGroup*) so it can be used by an application needing a new cartridge:

```
ov_cart -n label -p 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 cartridgeID -p partitionName -s SideA
```

To introduce a cartridge that already contains data:

1. Follow the procedure 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 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, “Operating OpenVault.”

Monitoring OpenVault

Monitoring OpenVault helps you spot potential problems so 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 elaborate on these tasks:

- “Checking Server Status and Configuration” on page 92
- “Checking Media Inventory” on page 93
- “Listing Cartridge Information” on page 93

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 reference pages for a complete description of available commands and options (see Appendix B for a complete listing of reference pages).

Checking Server Status and Configuration

Checking 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 all status 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`
- Show library (-l), configuration (-c), and slot map (-s) information for a library:
`ov_stat -L libName -l -c -s`

Checking Media Inventory

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

The following 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 '*'
```

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 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 '*'
```

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 are:

- “Performing Daily Tasks” on page 95
- “Performing Occasional Tasks” on page 98

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). The subsections described in this section are:

- “Manipulating Cartridges” on page 95
- “Managing Devices” on page 97

Manipulating Cartridges

The operator manipulates cartridges by performing the following operations:

- “Mounting Cartridges” on page 95
- “Checking the Task Request Queue” on page 96
- “Canceling a Pending Task Request” on page 96

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. By default, you specify a cartridge with its PCL (*ov_mount -C PCL*).

With the mount command options, you can specify:

- a cartridge using its cartridge ID:
`ov_mount -C cartridgeID`
- a particular drive by its drive name, as recorded in the client configuration file (see “Setting Up Drives” on page 81):
`ov_mount -d driveName -C PCL`
- a cartridge owned by an application:
`ov_mount -C PCL -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 “Mounting Cartridges.”

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
```

Canceling a Pending Task Request

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.

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 “Setting Up Drives” on page 81 and “Setting Up Robotic Libraries” on page 81).

The tasks described in this section are:

- “Disabling and Enabling Devices” on page 97
- “Cleaning a Drive” on page 98

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
```

Cleaning a Drive

Note: Cleaning facilities are not supported in OpenVault release 1.1.

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 also helps find and load a cleaning cartridge, if one is available.

Drives are specified by the name recorded in the client configuration file (see "Setting Up Drives" on page 81)

- 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`

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.

The tasks described in this section are:

- "Removing Cartridges From Libraries" on page 99
- "Moving Cartridges Within Libraries" on page 99
- "Maintaining the Server Catalog" on page 100

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 PCL`
- Eject a cartridge using its location (library, slot, and bay, if bays are present):
`ov_eject -l libraryName -s slotNumber -b bayName`

Moving Cartridges Within Libraries

Note: Cartridge move is not supported in OpenVault release 1.1. To move cartridges within a library or between libraries, first eject them and then inject them into the library. See `ov_eject` (1M) and `ov_inject`(1M) 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`

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. The tasks in this section describe these actions for the catalog:

- “Recycling Cartridges” on page 100
- “Destroying Cartridges” on page 100

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 PCL`
- Recycle the cartridge, using its cartridge ID:
`ov_recycle -C cartID`
- Recycle all cartridges owned by the given application:
`ov_recycle -A application`

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:
`ov_purge PCL`
- Remove the cartridge entry, using its cartridge ID:
`ov_purge -C cartID`
- Suppress the interactive verification; immediately remove the cartridge entry:
`ov_purge -f PCL`

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

Tertiary Storage Management

This chapter describes some strategies for optimizing storage management:

- “Tertiary Storage Devices” describes the mass storage options available today.
- “Connecting to a Host Computer” on page 104 talks about cabling tertiary storage devices to server machines.
- “Storage Management Applications” on page 107 discusses tertiary storage software for backup, archive, and hierarchical storage management.

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).

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.

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 6-1 shows the characteristics of several popular tape drives now on the market.

Table 6-1 High Capacity Tape Drives

Drive	Native Capacity	Data Rate	Cartridge Load Time	Tape Size	Expected MTBF	Power Needed	Typical Price
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
Exabyte Mammoth	20 GB	3.0 MB/sec	20 secs	8 mm	200,000 hours	15 w	\$4200
IBM Magstar 3570	5 GB	6.0 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

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.

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.

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 6-1.

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.

Silo Libraries

When data transfer speed and aggregate storage capacity are at a premium, datacenters often choose proprietary silo libraries from 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.

Connecting to a Host Computer

This section offers guidelines for connecting drives and removable media libraries to host computers such as Silicon Graphics servers.

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 6-2 compares the bandwidth of different SCSI types.

Table 6-2 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.

Three types of connectors are in wide use today:



Figure 6-1 68-pin Wide SCSI-3 Connector



Figure 6-2 50-pin SCSI-2 Connector (Mini-micro)

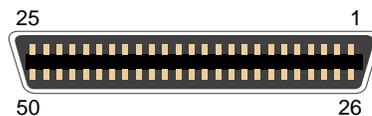


Figure 6-3 50-pin Centronics Parallel Connector

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. Silicon Graphics 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.

Silicon Graphics Servers

The Silicon Graphics Origin2000 enterprise server can contain one to four node cards, each with one or two CPUs. The Origin2000 server is capable of transfer rates at 5 GBps sustained, and up to 6.24 GBps peak. In rack-mounted configurations, as many as 16 modules (64 nodes, 128 CPUs) can be connected with CrayLink. Each Origin2000 module can contain 12 XIO cards; a three-slot PCI board can be substituted for one of the XIO cards. Each Origin2000 module can hold five 3.5-inch UltraSCSI devices, usually disk, and one half-height 5.25-inch SCSI device. For mass storage, the following XIO card configurations are supported:

- four-port UltraSCSI (three differential, one single-ended or differential)
- UltraSCSI, 100BaseT Ethernet, two 460 Kbps serial ports
- three PCI slots, either 32-bit or 64-bit PCI cards supported
- two-port Fibre Channel (copper or fiber)
- external VME adapter, with I/O expansion device

The Silicon Graphics Origin200 server tower includes two SCSI controllers with internal SCSI connectors. An Origin200 tower can accommodate six 3.5-inch UltraSCSI devices, usually disk, and two half-height 5.25-inch SCSI devices.

Two towers can be connected to double capacity. For maximum flexibility, you choose how to configure external SCSI connectors. For example, an UltraSCSI adapter can be placed in one of three PCI slots. The Origin200 PCI bus is capable of delivering 267 MBps throughput. For faster aggregate bandwidth, the Origin200 server can be connected with CrayLink to an I/O tower offering the same choice of XIO cards as the Origin2000 server.

The Silicon Graphics O2 workstation, although designed for desktop applications, is cabled for 68-pin Ultra SCSI (fast and wide) capable of delivering 40 MBps data throughput, more than enough for any tape drive currently on the market.

Storage Management Applications

This section gives an overview of software currently used for tertiary storage.

Scheduled Backup

The original 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.

In 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 6-4 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.

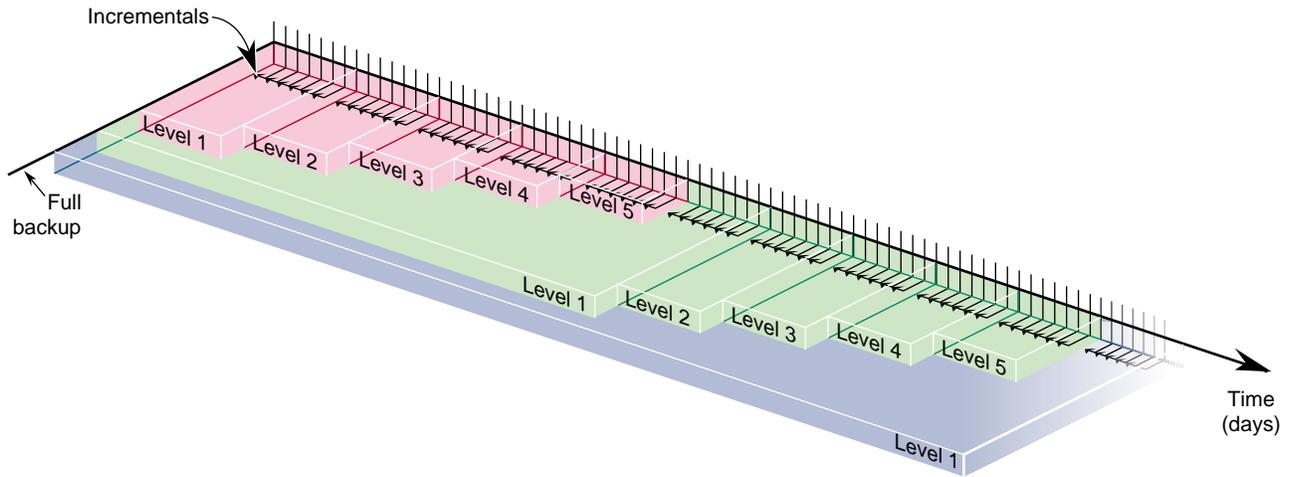


Figure 6-4 Incremental and Differential Backups

Because incremental and differential backups save only the files that change, much 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 saved. 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.

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.

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.

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.

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 Error Messages

This appendix explains how to resolve error conditions, and includes a description of the possible error messages. The sections in this appendix are:

- “Error Conditions” on page 113
- “Accessing the OpenVault Message Log” on page 113
- “Error Messages” on page 113

Most OpenVault errors are found in the log file `/usr/OpenVault/var/OVLOG`. They can also be listed with the `ov_msg -l` option; see `ov_msg(1M)`.

Error Conditions

Errors such as tape read and write errors are associated with an application error, whereas errors such as “cannot find tape” or “cannot associate with driver” are OpenVault errors.

Accessing the OpenVault Message Log

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 `ov_msg(1M)`.

Error Messages

Refer to the OpenVault release notes for a listing of the error messages that apply to the OpenVault system.

OpenVault Reference Pages

This appendix provides a brief introduction to the reference pages for commands used to administer and operate OpenVault. Refer to the reference page itself for a complete description of options and usage information.

Table B-1 OpenVault Reference Pages

Name	Use	Description
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.
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.

Table B-1 (continued) OpenVault Reference Pages

Name	Use	Description
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_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_shutdown(1M)	administrative	Shuts down and restarts OpenVault.
ov_slotttype(1M)	status	Manages OpenVault slot types.
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_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_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

A

- A-API, 3
- A-API interface, 7
- A-API/R, 3
- ADI, 3
- ADI/DCP interface, 8
- ADI/R, 3
- ALI, 3
- ALI/LCP interface, 8
- ALI/R, 3
- application, setup, 82
- applications
 - registering, 85
 - security, 86
 - setup, 82
 - unregistering, 85
- architecture, 5
- audience, xvi
- authorization key, 85, 86
- authorization key file, 82

B

- barcode, 3
- bay, 3, 83

C

- canceling a task request, 96
- CAPI, 4, 7
- CAPI/R, 4
- cartridge, 3
 - group, 3
 - ID, 89
 - information, listing, 93
- cartridge groups
 - managing, 87
 - setup, 88
- cartridges
 - ejecting, 99
 - introducing, 89
 - manipulating, 95
 - mounting, 95
 - moving within libraries, 99
 - removing from libraries, 99
- catalog, 3, 89
 - maintaining, 100
- checking media inventory, 93
- client host, 4
- clients, selecting, 75
- client/server model, 3
- client setup, 81
- configuration file
 - drive, 81
 - library, 81
- configuration files, 76
- configuration script

- drives, 81
 - library, 81
- configuring OpenVault, 84
- conventions, xvi
- creating a cartridge group, 89
- customer support, xvi

D

- daily tasks, 95
- DCP, 4, 5, 81
 - boot, 9
- debugging, 84
- Degauss, 87
- devices
 - disabling, 97
 - managing, 97
- disabling a device, 97
- DLT, 4
- domain of control, 75
- drive group, 4
- drives, enabling, 86
- drive setup, 81

E

- ejecting cartridges, 99
- enabling a device, 97
- enabling drives, 86
- error conditions, 113
- error messages, 113

F

- files

- authorization key, 82
- debugging messages, 84
- library configuration, 81
 - .login*, 84
- message log, 84

H

- human operator, 83

I

- installing OpenVault, 76
- introducing cartridges, 89
- introduction, OpenVault, 1

K

- key authorization file, 77

L

- LCP, 4, 5, 81, 99
 - boot, 8
- library, 4
 - contents, 99

M

- managing
 - cartridge groups, 87
 - cartridges, 95
 - devices, 97
 - server catalog, 100
- man pages, 115

manual libraries, setup, 83
media
 inventory, checking, 93
message log, 84, 113
middleware, 2
monitoring
 OpenVault, 91
mounting cartridges, 95
moving cartridges within libraries, 99

O

occasional tasks, 98
offline, device, 97
offsite libraries, setup, 83
OpenVault application, 5
OpenVault catalog, 5
OpenVault client, 5
OpenVault core, 5
OpenVault server, 4, 7
OpenVault system, 5
operation, OpenVault, 8
operator tasks, performing, 95
operator terminal, 83
optimizing storage management, 101
organization, storage, 98
OVSERVER, 84

P

password, 82, 85, 86
PCL, 4, 89
performing operator tasks, 95
planning storage needs, 83
prerequisites, 75

problems, whom to call, xvi
product support, xvi
purpose of OpenVault, 1

R

reference pages, 115
registering applications, 85
related documentation, xv
release notes, 76
removing cartridges from libraries, 99
robotic libraries, setup, 81

S

security, 82, 86
selecting clients, 75
server environment variable, OVSERVER, 84
server host, 4
server location, 75
server setup, 77
server status, 92
setting up, 82
 applications, 82
 cartridge groups, 88
 clients, 81
 drives, 81
 manual libraries, 83
 offsite libraries, 83
 OpenVault, 76
 robotic libraries, 81
 servers, 77
setup script, 77
shared secret, 4, 82
Silicon Graphics customer support, xvi
slot, 4

- addresses, 99
- slotmap, 4
- status, drive, 100
- storage management, optimizing, 101
- style conventions, xvi

T

- task request
 - canceling, 96
 - queue, checking, 96
- Technical Assistance Center, xvi

U

- unregistering applications, 85

Tell Us About This Manual

As a user of Silicon Graphics products, you can help us to better understand your needs and to improve the quality of our documentation.

Any information that you provide will be useful. Here is a list of suggested topics:

- General impression of the document
- Omission of material that you expected to find
- Technical errors
- Relevance of the material to the job you had to do
- Quality of the printing and binding

Please send the title and part number of the document with your comments. The part number for this document is 007-3211-003.

Thank you!

Three Ways to Reach Us

- To send your comments by **electronic mail**, use either of these addresses:
 - On the Internet: techpubs@sgi.com
 - For UUCP mail (through any backbone site): *[your_site]!sgi!techpubs*
- To **fax** your comments (or annotated copies of manual pages), use this fax number: 650-932-0801
- To send your comments by **traditional mail**, use this address:

Technical Publications
Silicon Graphics, Inc.
2011 North Shoreline Boulevard, M/S 535
Mountain View, California 94043-1389