

Tru64 UNIX

Logical Storage Manager

Part Number: AA-RH9BD-TE

September 2002

Product Version: Version 5.1B or higher

This manual describes how to configure and manage disk storage using the Logical Storage Manager (LSM) software. It includes information on LSM concepts and how to plan, set up, monitor, change, and troubleshoot an LSM configuration.

© 2002 Hewlett-Packard Company

Motif[®], The Open Group[™], and UNIX[®] are trademarks of The Open Group in the U.S. and/or other countries. All other product names mentioned herein may be the trademarks of their respective companies.

Confidential computer software. Valid license from Compaq Computer Corporation, a wholly owned subsidiary of Hewlett-Packard Company, required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

None of Compaq, HP, or any of their subsidiaries shall be liable for technical or editorial errors or omissions contained herein. The information is provided "as is" without warranty of any kind and is subject to change without notice. The warranties for HP or Compaq products are set forth in the express limited warranty statements accompanying such products. Nothing herein should be construed as constituting an additional warranty.

Contents

About This Manual

1 Overview

1.1	Overview of the LSM Object Hierarchy	1-1
1.1.1	LSM Disk	1-2
1.1.1.1	Disk Access Name	1-4
1.1.1.2	Disk Media Name	1-5
1.1.2	Disk Group	1-5
1.1.3	Subdisk	1-6
1.1.4	Data Plex	1-7
1.1.4.1	Concatenated Data Plex	1-8
1.1.4.2	Striped Data Plex	1-9
1.1.4.3	RAID 5 Data Plex	1-11
1.1.5	Log Plex	1-12
1.1.6	LSM Volume	1-13
1.1.6.1	Nonredundant Volumes	1-14
1.1.6.2	Mirrored Volumes	1-14
1.1.6.3	RAID 5 Volumes	1-17
1.1.6.4	Volume Usage Types	1-18
1.1.6.5	Volume Device Interfaces	1-18
1.2	Overview of LSM Interfaces	1-19
1.2.1	LSM Command-Line Interpreter	1-20
1.2.2	Storage Administrator Interface	1-23

2 Planning LSM Volumes and Disk Groups

2.1	Planning LSM Volumes	2-1
2.1.1	Planning Dirty-Region Logs	2-4
2.1.2	Planning RAID 5 Logs	2-5
2.1.3	Worksheet for LSM Volumes with Concatenated Plexes ...	2-7
2.1.4	Worksheet for LSM Volumes with Striped Plexes	2-8
2.1.5	Worksheet for LSM Volumes with a RAID 5 Plex	2-9
2.2	Planning Disk Groups	2-10
2.2.1	Guidelines for the Number and Usage of Disks	2-10
2.2.2	Guidelines for Connectivity and Availability	2-10
2.2.3	Worksheet for Planning Disk Groups	2-14

2.3	Finding Unused Storage Devices	2-15
2.3.1	Using the Disk Configuration GUI to Find Unused Disks	2-15
2.3.2	Using Operating System Commands to Find Unused Disks	2-16
2.3.3	Using the LSM voldisk Command to Find Unused Disks .	2-17

3 Installing and Initializing the LSM Software

3.1	Installing the LSM Software	3-1
3.2	Installing the LSM License	3-2
3.3	Initializing LSM	3-3
3.4	Using LSM for Critical File Systems and Swap Space	3-5
3.4.1	Creating Alternate Boot Disks (Standalone System)	3-6
3.4.1.1	Restrictions and Requirements	3-6
3.4.1.2	Encapsulating System Partitions (Creating System Volumes)	3-7
3.4.1.3	Mirroring System Volumes	3-10
3.4.1.4	Displaying Information for System Volumes	3-11
3.4.2	Using LSM Volumes for Cluster Domains	3-13
3.4.2.1	Migrating AdvFS Domains to LSM Volumes	3-14
3.4.2.1.1	Disk Space Considerations	3-14
3.4.2.1.2	Migrating AdvFS Domains	3-15
3.4.2.1.3	Mirroring Migrated Domain Volumes to the Original Disk (Optional)	3-20
3.4.3	Encapsulating Cluster Members' Swap Devices	3-21
3.5	Enabling the Automatic Data Relocation (Hot-Sparing) Feature	3-23
3.5.1	Configuring and Deconfiguring Hot-Spare Disks	3-24
3.5.2	Examples of Mail Notification for Exception Events	3-25
3.6	LSM Files, Directories, Device Drivers, and Daemons	3-29
3.6.1	LSM Files	3-29
3.6.2	LSM Directories	3-29
3.6.3	LSM Device Drivers	3-30
3.6.4	LSM Daemons	3-30

4 Creating LSM Disks, Disk Groups, and Volumes

4.1	Overview of Creating LSM Disks	4-1
4.1.1	Overview of Configuration Database Copies	4-2
4.1.2	Overview of Disk Offsets	4-2
4.1.3	Creating LSM Sliced Disks	4-3
4.1.4	Creating LSM Simple Disks	4-3
4.1.5	Backing Up Disk Label Information	4-4

4.2	Creating Disk Groups	4-4
4.2.1	Creating LSM Disks and Disk Groups Using the voldiskadd Script	4-5
4.2.2	Creating Disk Groups Using the voldg Command	4-8
4.3	Creating LSM Volumes for New Data	4-9
4.3.1	Overview of LSM Volume Attributes	4-9
4.3.2	Creating LSM Volumes with a Single Concatenated Plex ..	4-12
4.3.3	Creating LSM Volumes with Mirrored, Concatenated Plexes	4-12
4.3.3.1	Creating a Mirrored, Concatenated Volume in One Step	4-13
4.3.3.2	Creating a Mirrored, Concatenated Volume with Plexes on Different Buses	4-14
4.3.4	Creating LSM Volumes with a Single Striped Plex	4-14
4.3.5	Creating LSM Volumes with Mirrored, Striped Plexes	4-16
4.3.5.1	Creating a Mirrored, Striped Volume in One Step	4-17
4.3.5.2	Creating a Mirrored, Striped Volume with Plexes on Different Buses	4-18
4.3.6	Creating LSM Volumes with a RAID 5 Plex	4-19
4.3.7	Creating LSM Volumes for Swap Space	4-21
4.3.7.1	Creating Swap Volumes	4-22
4.3.7.2	Mirroring Swap Volumes	4-23
4.4	Creating LSM Volumes with Nondefault Properties	4-24
4.4.1	Creating a Striped Plex with Subdisks of Different Sizes ..	4-24
4.4.2	Creating a Striped Plex with Disks on Different Buses	4-27
4.4.3	Creating a RAID 5 Plex with Disks on Different Buses	4-31
4.5	Configuring File Systems to Use LSM Volumes	4-33
4.5.1	Configuring AdvFS Domains to Use LSM Volumes	4-33
4.5.2	Configuring UFS File Systems to Use LSM Volumes	4-34
4.6	Creating LSM Volumes for Existing Data	4-34
4.6.1	Encapsulating Disks or Disk Partitions	4-35
4.6.2	Creating LSM Volumes for AdvFS Domains	4-36
4.6.2.1	Encapsulating an AdvFS Domain	4-36
4.6.2.2	Migrating an AdvFS Domain	4-37

5 Managing LSM Objects

5.1	Managing LSM Disks	5-1
5.1.1	Displaying LSM Disk Information	5-1
5.1.2	Renaming LSM Disks	5-1
5.1.3	Placing LSM Disks Off Line	5-2
5.1.4	Placing LSM Disks On Line	5-2

5.1.5	Moving Data Off an LSM Disk	5-3
5.1.6	Moving LSM Objects Relocated by Hot-Sparing	5-5
5.1.7	Removing Disks from LSM Control	5-6
5.2	Managing Disk Groups	5-7
5.2.1	Displaying Disk Group Information	5-7
5.2.2	Adding LSM Disks to Disk Groups	5-8
5.2.3	Removing LSM Disks from Disk Groups	5-8
5.2.4	Deporting Disk Groups	5-9
5.2.5	Importing Disk Groups	5-10
5.2.6	Renaming Disk Groups	5-10
5.2.7	Creating a Clone Disk Group	5-11
5.3	Managing the LSM Configuration Database	5-12
5.3.1	Backing Up the LSM Configuration Database	5-12
5.3.2	Restoring the LSM Configuration Database from Backup	5-14
5.3.3	Changing the Size and Number of Configuration Database Copies	5-15
5.3.3.1	Reducing the Number of Configuration Database Copies on an LSM Disk	5-16
5.3.3.2	Removing Configuration Database Copies on LSM Disks with Small Private Regions	5-17
5.4	Managing LSM Volumes	5-18
5.4.1	Displaying LSM Volume Information	5-18
5.4.2	Backing Up LSM Volumes	5-19
5.4.2.1	Overview of the Fast Plex Attach Feature	5-21
5.4.2.2	Creating a Backup Volume Using the Fast Plex Attach Feature (volassist snapfast and snapback)	5-26
5.4.2.3	Creating a Backup Volume by Attaching a New Plex (volassist snapstart and snapshot)	5-29
5.4.2.4	Creating a Backup Volume by Detaching an Existing Plex	5-30
5.4.2.5	Backing Up a Nonredundant or RAID 5 Volume	5-31
5.4.2.6	Performing a Backup	5-32
5.4.2.7	Special Case: Backing Up LSM Volumes in an AdvFS Domain	5-32
5.4.3	Restoring LSM Volumes from Backup	5-35
5.4.4	Starting LSM Volumes	5-36
5.4.5	Stopping LSM Volumes	5-36
5.4.6	Removing LSM Volumes	5-37
5.4.6.1	Unencapsulating AdvFS Domains	5-38
5.4.7	Disabling the Fast Plex Attach Feature on LSM Volumes	5-40
5.4.8	Renaming LSM Volumes	5-40
5.4.9	Growing LSM Volumes	5-40

5.4.9.1	Growing LSM Volumes by a Specific Amount	5-41
5.4.9.2	Growing LSM Volumes to a Specific Size	5-42
5.4.10	Shrinking LSM Volumes	5-43
5.4.10.1	Shrinking LSM Volumes by a Specific Amount	5-44
5.4.10.2	Shrinking LSM Volumes to a Specific Size	5-44
5.4.11	Changing LSM Volume Permission, User, and Group Attributes	5-45
5.5	Managing Plexes	5-45
5.5.1	Displaying Plex Information	5-45
5.5.2	Adding a Data Plex (Mirroring LSM Volumes)	5-46
5.5.3	Adding a Log Plex	5-47
5.5.4	Adding an FPA Plex	5-48
5.5.5	Detaching a Plex	5-49
5.5.6	Dissociating a Plex	5-50
5.5.7	Reattaching Plexes	5-51
5.5.8	Changing the Plex Layout of LSM Volumes	5-51
5.5.8.1	Changing the Plex Layout from Concatenated to Striped	5-51
5.5.8.2	Changing the Plex Layout from Striped to Concatenated	5-53
5.6	Managing Subdisks	5-54
5.6.1	Displaying Subdisk Information	5-54
5.6.2	Joining Subdisks	5-54
5.6.3	Splitting Subdisks	5-55
5.6.4	Moving Subdisks to a Different Disk	5-55
5.6.5	Removing Subdisks	5-56

6 Troubleshooting

6.1	Troubleshooting LSM Objects	6-1
6.1.1	Monitoring LSM Events	6-1
6.1.2	Monitoring Read and Write Statistics	6-2
6.1.2.1	Displaying Read and Write Statistics	6-2
6.1.2.2	Displaying Failed Read and Write Statistics	6-3
6.1.3	Monitoring LSM Object States	6-3
6.1.3.1	Overview of LSM Kernel States	6-4
6.1.3.2	Overview of LSM Object States	6-4
6.2	Troubleshooting a Missing or Altered sysconfigtab File	6-7
6.3	Troubleshooting LSM Startup and Command Problems	6-8
6.3.1	Checking the Volume Configuration Daemon (vold)	6-8
6.3.2	Restarting the Volume Configuration Daemon (vold)	6-8
6.3.3	Checking the Volume Extended I/O Daemon (voliod)	6-9

6.4	Troubleshooting LSM Disks	6-10
6.4.1	Checking Disk Status	6-10
6.4.2	Recovering Stale Subdisks	6-10
6.4.3	Recovering From Temporary Disk Failures	6-11
6.4.4	Moving LSM Volumes Off a Failing Disk	6-11
6.4.5	Replacing Failed Disks	6-12
6.4.6	Replacing Failed Boot Disks	6-14
6.5	Troubleshooting LSM Volumes	6-17
6.5.1	Recovering LSM Volumes After a System Failure	6-17
6.5.2	Recovering LSM Volumes After Disk Failure	6-17
6.5.2.1	Recovering Mirrored and RAID 5 LSM Volumes	6-18
6.5.2.2	Recovering a Nonredundant Volume	6-18
6.5.2.3	Recovering a Mirrored Volume with No Valid Plexes ..	6-19
6.5.2.4	Recovering a Mirrored Volume with One Valid Plex ...	6-20
6.5.2.5	Recovering a RAID 5 Plex from Multiple Disk	
	Failures	6-20
6.5.2.6	Recovering a RAID 5 Log Plex	6-21
6.5.3	Starting Disabled LSM Volumes	6-22
6.5.4	Checking the Status of Volume Resynchronization	6-23
6.5.4.1	Changing the Rate of Future Volume	
	Resynchronizations	6-24
6.5.5	Calculating Sufficient Space to Create LSM Volumes	6-25
6.5.6	Clearing Locks on LSM Volumes	6-27
6.6	Troubleshooting Disk Groups	6-27
6.6.1	Resolving Mismatched Host IDs	6-27
6.6.2	Importing Disk Groups with Failed Disks (Forced	
	Import)	6-28

7 Special Cases

7.1	Upgrading an LSM Configuration	7-1
7.1.1	Increasing the Size of BCLs	7-2
7.1.2	Backing Up the LSM Configuration	7-2
7.1.3	Deporting Disk Groups (Optional)	7-3
7.1.4	Upgrading the LSM Software	7-3
7.1.5	Manually Converting Version 4.0 Disk Groups	7-4
7.1.6	Optimizing Restored LSM Configuration Databases	
	(Optional)	7-5
7.2	Adding a System with LSM to a Cluster	7-6
7.3	Moving Disk Groups Between Systems	7-10
7.3.1	Moving the rootdg Disk Group to Another System	7-11
7.3.2	Moving Disk Groups to Another System	7-12

7.3.3	Moving Disk Groups with nopriv Disks to Another System	7-13
7.4	Unencapsulating the Boot Disk (Standalone System)	7-16
7.5	Migrating AdvFS Domains from LSM Volumes to Physical Storage	7-20
7.6	Unencapsulating a Cluster Member's Swap Devices	7-20
7.7	Uninstalling the LSM Software	7-22

A The Storage Administrator GUI (lsmsa)

A.1	Overview of the Storage Administrator	A-1
A.1.1	Installing and Starting the Storage Administrator GUI ...	A-2
A.1.2	Storage Administrator Main Window Components	A-3
A.2	Performing Tasks with the Storage Administrator	A-5
A.2.1	Selecting Objects	A-5
A.2.2	Using the Console Menu and Selected Menu	A-5
A.2.3	Using the Right Mouse Button to Display a Pop-Up Menu	A-6
A.2.4	Using the Command Launcher Window	A-6
A.2.5	Exiting the Storage Administrator	A-6
A.3	Managing LSM Disks	A-6
A.3.1	Adding an LSM Disk	A-6
A.3.2	Adding a Hot-Spare Disk	A-8
A.3.3	Moving Data Off an LSM disk	A-9
A.3.4	Mirroring an LSM disk	A-10
A.3.5	Placing an LSM Disk Off Line	A-10
A.3.6	Placing an LSM disk On Line	A-11
A.3.7	Recovering LSM Volumes on an LSM disk	A-11
A.3.8	Removing LSM Disks from Disk Groups	A-12
A.3.9	Renaming an LSM disk	A-13
A.3.10	Replacing an LSM disk	A-14
A.3.11	Scanning for New Disks	A-15
A.4	Managing Disk Groups	A-15
A.4.1	Creating Disk Groups	A-15
A.4.2	Adding LSM Disks to Disk Groups	A-16
A.4.3	Adding Disk Partitions to Disk Groups	A-17
A.4.4	Deporting Disk Groups	A-17
A.4.5	Importing a Deported Disk Group	A-18
A.4.6	Recovering LSM Volumes in Disk Groups	A-20
A.4.7	Renaming Disk Groups	A-20
A.4.8	Moving Disk Groups to Another System	A-21
A.4.9	Destroying Disk Groups	A-21
A.5	Managing Subdisks	A-22
A.5.1	Joining Subdisks	A-22

A.5.2	Splitting Subdisks	A-23
A.5.3	Moving Subdisks	A-24
A.5.4	Removing Subdisks	A-26
A.6	Managing LSM Volumes	A-26
A.6.1	Creating LSM Volumes	A-27
A.6.2	Adding Mirrors to LSM Volumes	A-29
A.6.3	Adding Logs to LSM Volumes	A-30
A.6.4	Mounting UFS File Systems on LSM Volumes	A-31
A.6.5	Adding File Systems to LSM Volumes	A-32
A.6.6	Checking UFS File Systems	A-34
A.6.7	Unmounting UFS File Systems on LSM Volumes	A-35
A.6.8	Mapping and Analyzing LSM Volumes, Disks, and Subdisks	A-35
A.6.9	Disabling a Mirror in LSM Volumes	A-37
A.6.10	Repairing Disabled Mirrors	A-38
A.6.11	Creating Snapshots of Volume Data	A-39
A.6.12	Recovering LSM Volumes	A-40
A.6.13	Preparing LSM Volumes for Restoration	A-41
A.6.14	Removing Mirrors from LSM Volumes	A-41
A.6.15	Removing Logs from LSM Volumes	A-41
A.6.16	Removing LSM Volumes	A-42
A.6.17	Renaming LSM Volumes	A-43
A.6.18	Resizing LSM Volumes	A-43
A.6.19	Restarting LSM Volumes	A-45
A.6.20	Stopping LSM Volumes	A-45
A.7	Performing AdvFS Operations	A-46
A.7.1	Creating AdvFS Domains	A-46
A.7.2	Adding LSM Volumes to AdvFS Domains	A-48
A.7.3	Adding Filesets to AdvFS Domains	A-49
A.7.4	Defragmenting AdvFS Domains	A-50
A.7.5	Cloning AdvFS Filesets	A-51
A.7.6	Changing Fileset Quotas	A-52
A.7.7	Unmounting AdvFS Filesets	A-53
A.7.8	Renaming AdvFS Filesets	A-54
A.7.9	Mounting AdvFS Filesets	A-54
A.7.10	Removing AdvFS Filesets	A-55

B Understanding and Customizing the Storage Administrator GUI

B.1	Tracking Storage Administrator Activities	B-1
B.1.1	Overview of the Command Log File	B-1
B.1.2	Overview of the Access Log File	B-2
B.1.3	Overview of the Server Log File	B-2

B.2	Working with Dialog Boxes	B-3
B.2.1	Specifying Objects in Dialog Boxes	B-4
B.2.2	Specifying Object Sizes in Dialog Boxes	B-4
B.3	Viewing Objects and Object Properties	B-5
B.3.1	Overview of the Main Window	B-5
B.3.2	Viewing Volume Layout Details	B-5
B.3.3	Overview of Object Properties Dialog Boxes	B-6
B.3.4	Overview of the User Preferences Dialog Box	B-7
B.3.4.1	General Preferences	B-9
B.3.4.2	Main Window Preferences	B-10
B.3.4.3	Font Preferences	B-10
B.3.4.4	Color Preferences	B-10
B.3.4.5	Geometry Preferences	B-11
B.3.4.6	Tree/Table Preferences	B-12
B.3.4.7	Toolbar Preferences	B-12
B.3.4.8	Layout Details Preferences	B-12
B.3.5	Overview of the Alert Monitor Window	B-13
B.3.6	Overview of the Object Table Copy Window	B-13
B.3.7	Overview of the Object Search Window	B-14
B.3.8	Overview of the Task Request Monitor Window	B-15
B.4	Storage Administrator GUI Shortcuts	B-15
B.4.1	Sorting Objects	B-16
B.4.2	Clearing Alerts	B-16
B.4.3	Using Keyboard Shortcuts	B-16
B.4.4	Docking the Toolbar and Command Launcher	B-17

Glossary

Index

Examples

3-1	Sample Mail Notification	3-26
3-2	Complete Disk Failure Mail Notification	3-27
3-3	Partial Disk Failure Mail Notification	3-27
3-4	Successful Data Relocation Mail Notification	3-27
3-5	No Hot-Spare or Free Disk Space Mail Notification	3-28
3-6	Data Relocation Failure Mail Notification	3-28
3-7	Unusable Volume Mail Notification	3-28
3-8	Unusable RAID 5 Volume Mail Notification	3-28
4-1	Creating an LSM Disk Group with the voldiskadd script	4-7
4-2	LSM Volume Attribute Defaults File	4-11

6-1	Volume Recovery Section of /sbin/lsm-startup Script	6-24
-----	---	------

Figures

1-1	Typical LSM Hardware Configuration	1-3
1-2	Single Subdisk Using a Public Region	1-6
1-3	Multiple Subdisks Using a Public Region	1-7
1-4	Concatenated Data Plex	1-8
1-5	Volume with a Three-Column Striped Data Plex	1-10
1-6	Volume with a RAID 5 Data Plex	1-12
1-7	Volume with Concatenated and Mirrored Data Plexes	1-15
1-8	Volume with Striped and Mirrored Data Plexes	1-16
1-9	Two LSM Volumes Using Subdisks on the Same Disk	1-17
1-10	LSM Volumes Used Like Disk Partitions	1-19
2-1	Worksheet for LSM Volumes with Concatenated Plexes	2-7
2-2	Worksheet for LSM Volumes with Striped Plexes	2-8
2-3	Worksheet for LSM Volumes with a RAID 5 Plex	2-9
5-1	Volume Before Fast Plex Attach	5-21
5-2	Process of the volassist snapfast Command: Phase 1	5-22
5-3	Process of the volassist snapfast Command: Phase 2	5-23
5-4	Writes Occurring to Primary and Secondary LSM Volumes ...	5-24
5-5	Process of volassist snapback Command: Phase 1	5-25
5-6	Process of volassist snapback Command: Phase 2	5-26
A-1	Storage Administrator Main Window	A-3
A-2	Add Disk(s) Dialog Box	A-7
A-3	Evacuate Disk Dialog Box	A-9
A-4	Mirror Disk Dialog Box	A-10
A-5	Remove Disk Dialog Box	A-13
A-6	Replace Disk Dialog Box	A-14
A-7	New Disk Group Dialog Box	A-16
A-8	Add Disk Partition Dialog Box	A-17
A-9	Deport Disk Group Dialog Box	A-18
A-10	Import Disk Group Dialog Box	A-19
A-11	Rename Disk Group Dialog Box	A-21
A-12	Join Subdisks Dialog Box	A-23
A-13	Split Subdisk Dialog Box	A-24
A-14	Move Subdisks Dialog Box	A-25
A-15	New Volume Dialog Box	A-28
A-16	Add Mirror Dialog Box	A-30
A-17	Add Log Dialog Box	A-31
A-18	Mount File System Dialog Box	A-32
A-19	New File System Dialog Box	A-33
A-20	Check File System Dialog Box	A-34

A-21	Volume to Disk Mapping Window	A-36
A-22	Disable Mirror Dialog Box	A-38
A-23	Repair Mirror Dialog Box	A-39
A-24	Volume Snapshot Dialog Box	A-40
A-25	Remove Log Dialog Box	A-42
A-26	Rename Volume Dialog Box	A-43
A-27	Resize Volume Dialog Box	A-44
A-28	Storage Administrator Main Window	A-47
A-29	New Domain Dialog Box	A-47
A-30	Add Volume Dialog Box	A-49
A-31	Add Fileset Dialog Box	A-50
A-32	Defragment Domain Dialog Box	A-51
A-33	Clone Fileset Dialog Box	A-52
A-34	Fileset Quotas Dialog Box	A-53
A-35	Rename Fileset Dialog Box	A-54
A-36	Mount Fileset Dialog Box	A-55
B-1	Typical Storage Administrator Dialog Box	B-3
B-2	Volume Layout Details Window	B-6
B-3	Volume Properties Dialog Box	B-7
B-4	Preferences Dialog Box	B-8
B-5	Alert Monitor Window	B-13
B-6	Object Search Window	B-14
B-7	Task Request Monitor Window	B-15

Tables

1-1	LSM Commands Described in This Manual	1-21
2-1	Plex Layout Type Benefits and Trade-Offs	2-3
2-2	Default (Approximate) Sizes of DRL Logs	2-4
2-3	Completed Disk Group Worksheets for a Standalone System ..	2-13
2-4	Worksheet for Disk Group	2-14
3-1	LSM Software Subsets	3-2
3-2	LSM Commands for Providing Cluster Redundancy	3-14
3-3	LSM Device Special Files	3-29
3-4	LSM /etc/vol Subdirectories	3-29
3-5	LSM Block and Character Device Subdirectories	3-30
4-1	Common LSM Volume Attributes	4-10
6-1	Common volstat Command Options	6-2
6-2	LSM Volume Kernel States (KSTATE)	6-4
6-3	LSM Volume States (STATE)	6-4
6-4	LSM Plex States	6-6
6-5	LSM Subdisk States	6-7
6-6	vold Status and Solutions	6-8

6-7	Volume Recovery Procedures	6-17
B-1	Keyboard Shortcuts	B-16
B-2	Main Window Keyboard Shortcuts	B-16

About This Manual

This manual describes how to configure and manage disk storage using the Logical Storage Manager (LSM) software. It includes information on LSM concepts and how to plan, set up, monitor, change, and troubleshoot an LSM configuration.

Audience

This manual is intended for system administrators who configure and manage storage devices under LSM control. To use this manual you must be able to administer a standalone system running the HP Tru64 UNIX operating system software and its storage devices, or a TruCluster Server running the HP Tru64 UNIX operating system software and its storage devices.

New and Changed Features

The following changes have been made to this manual since the Version 5.1A release:

- Chapter 1 has been revised to better define LSM objects, including the new Fast Plex Attach log type.
- Chapter 2 has been expanded to discuss planning considerations specific to a TruCluster Server environment.
- Chapter 3 contains information on configuring LSM in a TruCluster Server environment. This information was previously included in the *Cluster Administration* manual.
- Chapter 5 has been expanded to describe the new Fast Plex Attach feature for backing up volumes.
- Chapter 6 has been revised to simplify the discussion of replacing failed disks and the subsequent volume recovery operations.
- Chapter 7 describes special cases in LSM management, including upgrading from a Version 4.0 system and uninstalling the LSM software.
- The following chapters and appendixes have been removed:
 - Former Chapter 7 — Error Messages
 - Appendix C — Using the voldiskadm Menu Interface
 - Appendix D — The Visual Administrator Interface (dxlsm)

– Appendix E — Using the Visual Administrator Interface

Information on the interfaces previously documented in these appendixes is available in the reference page for the relevant interface. For more information, see `voldiskadm(8)` and `dxlsm(8X)`.

Organization

This manual is organized as follows:

<i>Chapter 1</i>	Describes LSM features, terms, and concepts, and introduces the available interfaces to LSM.
<i>Chapter 2</i>	Presents factors to consider in planning your LSM configuration, describes how to find unused storage devices to use with LSM, and provides worksheets to aid you in planning LSM volumes and disk groups.
<i>Chapter 3</i>	Describes how to install and initialize LSM as part of a system or cluster upgrade (from Version 5.0 or higher to the current version) or as part of a full installation.
<i>Chapter 4</i>	Describes how to create new LSM disks, disk groups, and volumes, how to configure UFS file systems and AdvFS domains to use LSM volumes for their storage, and how to place existing data under LSM control.
<i>Chapter 5</i>	Describes common management tasks for LSM objects.
<i>Chapter 6</i>	Describes how to identify and recover from problems and how to replace failed disks under LSM control.
<i>Chapter 7</i>	Describes special cases in LSM management, such as upgrading a system with an existing LSM configuration from Version 4.0 to the current version, moving disk groups between systems, and removing (uninstalling) the LSM software from a single system or a cluster.
<i>Appendix A</i>	Describes how to start the Storage Administrator GUI and how to manage LSM objects using this interface.
<i>Appendix B</i>	Describes how to track Storage Administrator activities, how to use the Storage Administrator GUI, and how to customize the Storage Administrator GUI.
<i>Glossary</i>	

Related Documentation

The following operating system documents provide information related to LSM:

- *Installation Guide* describes how to install the LSM software as part of a full installation or separately from the operating system.

- *Release Notes* describe LSM problems and solutions that might not be documented elsewhere.
- *System Administration* describes general storage administration.
- *System Configuration and Tuning* describes how to plan, configure, and tune storage devices.
- *AdvFS Administration* describes how to use the AdvFS software with LSM.
- *Cluster Administration* describes the difference between how LSM performs in a standalone system versus in a TruCluster environment.

Icons on Tru64 UNIX Printed Manuals

The printed version of the Tru64 UNIX documentation uses letter icons on the spines of the manuals to help specific audiences quickly find the manuals that meet their needs. (You can order the printed documentation from HP.) The following list describes this convention:

- G Manuals for general users
- S Manuals for system and network administrators
- P Manuals for programmers
- R Manuals for reference page users

Some manuals in the documentation help meet the needs of several audiences. For example, the information in some system manuals is also used by programmers. Keep this in mind when searching for information on specific topics.

The *Documentation Overview* provides information on all of the manuals in the Tru64 UNIX documentation set.

The Tru64 UNIX and TruCluster Server documentation is available on the World Wide Web at:

<http://www.tru64unix.compaq.com/docs>

Reader's Comments

HP welcomes any comments and suggestions you have on this and other Tru64 UNIX manuals.

You can send your comments in the following ways:

- Fax: 603-884-0120 Attn: UBPG Publications, ZKO3-3/Y32
- Internet electronic mail: readers_comment@zk3.dec.com

A Reader's Comment form is located on your system in the following location:

```
/usr/doc/readers_comment.txt
```

Please include the following information along with your comments:

- The full title of the manual and the order number. (The order number appears on the title page of printed and PDF versions of a manual.)
- The section numbers and page numbers of the information on which you are commenting.
- The version of Tru64 UNIX that you are using.
- If known, the type of processor that is running the Tru64 UNIX software.

The Tru64 UNIX Publications group cannot respond to system problems or technical support inquiries. Please address technical questions to your local system vendor or to the appropriate HP technical support office. Information provided with the software media explains how to send problem reports to HP.

Conventions

This manual uses the following conventions:

#	A number sign represents the superuser prompt.
% cat	Boldface type in interactive examples indicates typed user input.
<i>file</i>	Italic (slanted) type indicates variable values, placeholders, and function argument names.
[]	In syntax definitions, brackets indicate items that are optional and braces indicate items that are required. Vertical bars separating items inside brackets or braces indicate that you choose one item from among those listed.
{ }	
...	In syntax definitions, a horizontal ellipsis indicates that the preceding item can be repeated one or more times.
cat(1)	A cross-reference to a reference page includes the appropriate section number in parentheses.

For example, `cat(1)` indicates that you can find information on the `cat` command in Section 1 of the reference pages.

Overview

The Logical Storage Manager (LSM) software is an optional integrated, host-based disk storage management application that lets you manage storage devices without disrupting users or applications accessing data on those storage devices. Although any system can benefit from LSM, it is especially suited to configurations with large numbers of disks or configurations that regularly add storage.

LSM uses Redundant Arrays of Independent Disks (RAID) technology to enable you to configure storage devices into a virtual pool of storage from which you create LSM volumes. You can configure new and existing UFS and AdvFS file systems, databases, and applications to use LSM volumes. You can also create LSM volumes on top of RAID storage sets.

The benefits of using an LSM volume instead of a disk partition include:

- Data loss protection, through mirroring (RAID 1) or striping with parity (RAID 5)
- Maximized disk usage, by seamlessly combining storage devices to appear as a single storage device to users and applications
- Performance improvements, through striping (RAID 0) over different disks and different buses
- Data availability in a TruCluster Server environment

TruCluster Server software makes multiple Tru64 UNIX systems appear as a single system on the network. The systems running the TruCluster Server software become members of the cluster and share resources and data storage. This sharing allows applications, such as LSM, to continue uninterrupted if the cluster member on which it was running fails.

This chapter introduces LSM features, concepts, terminology, and available interfaces. For more information on LSM terms and a list of all the LSM commands, see `volintro(8)`.

1.1 Overview of the LSM Object Hierarchy

LSM uses the following hierarchy of objects to organize storage:

- **LSM disk** — An object that represents a storage device that is initialized exclusively for use by LSM

- **Disk Group** — An object that represents a collection of LSM disks for use by an LSM volume
- **Subdisk** — An object that represents a contiguous set of blocks on an LSM disk that LSM uses to write volume data
- **Plex** — An object that represents a subdisk or collection of subdisks to which LSM writes a copy of the volume data or log information
- **Volume** — An object that represents a hierarchy of LSM objects, including LSM disks, subdisks, and plexes in a disk group. Applications and file systems make read and write requests to the LSM volume.

The following sections describe LSM objects in more detail.

1.1.1 LSM Disk

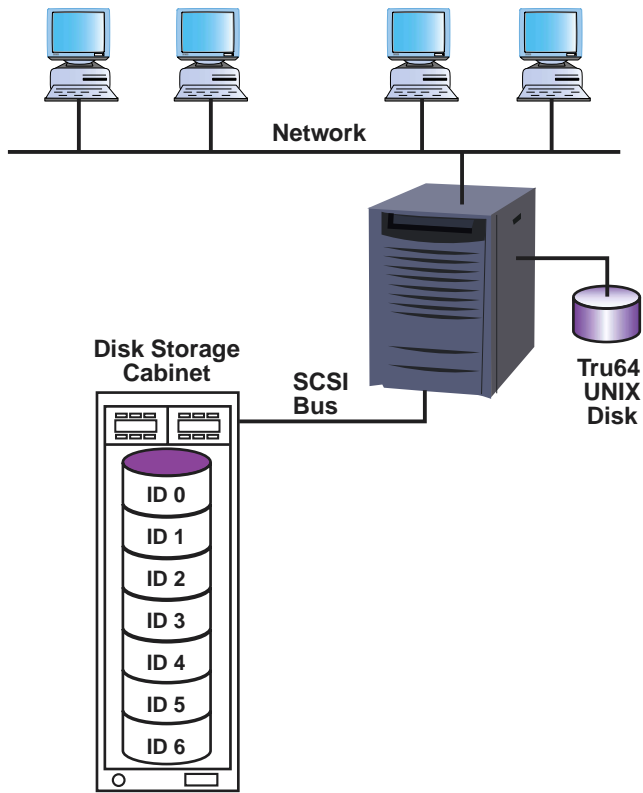
An LSM disk is any storage device supported by Tru64 UNIX, including disks, disk partitions, and hardware RAID sets, that you configure exclusively for use by LSM. LSM views the storage in the same way as the Tru64 UNIX operating system software views it. For example, if the operating system software treats a RAID set as a single storage device, so does LSM. In addition, LSM recognizes and supports hardware disk clones.

For more information on supported storage devices, see the *Tru64 UNIX QuickSpecs* web site at the following URL:

http://www.tru64unix.compaq.com/docs/pub_page/spds.html

Figure 1–1 shows a typical hardware configuration that LSM supports.

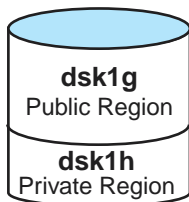
Figure 1–1: Typical LSM Hardware Configuration



ZK-1679U-AI

A storage device becomes one of the following LSM disk types when you initialize it for use by LSM:

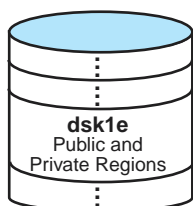
- A **sliced disk**, which is created when you commit an entire disk to LSM use.



ZK-1680U-AI

In a `sliced` disk, LSM organizes the storage into two regions on separate partitions — a large **public region** used for storing data and a **private region** for storing LSM internal metadata, such as LSM configuration information. The default size of the private region is 4096 blocks.

- A **simple disk**, which is created when you specify a disk partition for LSM use, including the `c` partition.

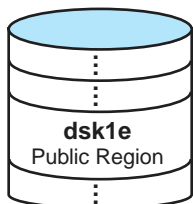


ZK-1681U-AI

In a `simple` disk, LSM organizes the storage into two regions on the same partition — a large public region used for storing data and a private region for storing LSM internal metadata, such as LSM configuration information. The default size of the private region is 4096 blocks.

Whenever possible, initialize the entire disk as a `sliced` disk instead of configuring individual disk partitions as `simple` disks. This ensures that the disk's storage is used efficiently and avoids using space for multiple private regions on the same disk.

- A **nopriv disk**, which is created when you encapsulate a disk or disk partition containing data you want to place under LSM control.



ZK-1682U-AI

In a `nopriv` disk, LSM creates only a public region for the existing data and no private region.

1.1.1.1 Disk Access Name

When you initialize a disk for LSM use, LSM assigns it a **disk access name** based on the device you specify. For example, if you initialize an entire disk (for example, `dsk4`), the disk access name is `dsk4`. If you initialize a disk partition (for example, `dsk4b`), the disk access name is `dsk4b`.

If you initialize multiple partitions of the same disk as separate LSM disks, each has its own disk access name; for example, `dsk2b` and `dsk2f`.

1.1.1.2 Disk Media Name

When you add an LSM disk to a disk group, it gets a **disk media name**, which can be either the same as the disk access name or a name you assign. Disk media names can include any combination of up through 31 alphanumeric characters but cannot include spaces or a slash (/).

For example, a disk with a disk access name of `dsk1` can also have a disk media name of `dsk1` or a name you assign, such as `finance_data_disk`.

LSM keeps track of the association of the disk media name and the disk access name. The disk media name provides insulation from operating system naming conventions. This association allows LSM to find the device if you move it to a new location (for example, to a different controller).

If you remove a disk from a disk group, it loses its disk media name. If you add the disk to a different disk group you can give it a different disk media name, or let it use the disk access name by default.

Within a disk group all the disk media names must be unique, but two different disk groups can have disks with the same disk media name.

1.1.2 Disk Group

A **disk group** is an object that represents a grouping of LSM disks. LSM disks in a disk group share a common **configuration database** that identifies all the LSM objects (LSM disks, subdisks, plexes, and volumes) in the disk group. LSM automatically creates and maintains copies of the configuration database in the private region of several LSM `sliced` or `simple` disks in each disk group.

The default size of the private region is 4096 blocks, and each LSM object requires one record. Two records fit in one sector (512 bytes). Therefore, the default private region size guarantees space for a configuration database that tracks 8192 objects (LSM disks, subdisks, plexes, and volumes).

LSM distributes these copies across all controllers for redundancy. If all disks in a disk group are located on the same controller, LSM distributes the copies across several disks. LSM automatically records changes to the LSM configuration and, if necessary, changes the number and location of copies of the configuration database for a disk group.

You cannot have a disk group that contains only LSM `nopriv` disks, because an LSM `nopriv` disk does not have a private region to store copies of the configuration database.

By default, the LSM software creates a disk group named **rootdg**. The configuration database for `rootdg` contains information for itself and all other disk groups that you create.

An LSM volume can use disks only within the same disk group. You can create all of your volumes in the `rootdg` disk group, or you can create other disk groups. For example, if you dedicate disks to store financial data, you can create and assign those disks to a disk group named `finance`.

1.1.3 Subdisk

A **subdisk** is an object that represents a contiguous set of blocks in an LSM disk's public region that LSM uses to store data.

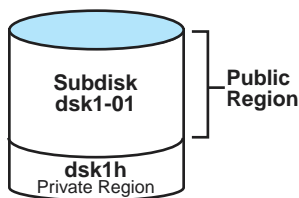
By default, LSM assigns subdisk names using the LSM disk media name followed by a dash (-) and an ascending two-digit number beginning with 01; for example, `dsk1-01`.

Alternatively, you can assign a subdisk name of up to 31 alphanumeric characters that cannot include spaces or the slash (/). For example, you can assign a subdisk name of `finance_disk-01` on a disk with a disk media name of `dsk3`.

A subdisk can be:

- The **entire public region**. Figure 1–2 shows that the entire public region of an LSM disk was configured as a subdisk named `dsk1-01`.

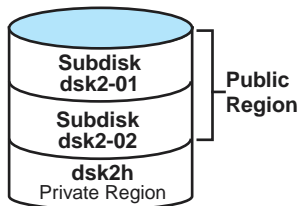
Figure 1–2: Single Subdisk Using a Public Region



ZK-1683U-AI

- A **portion of the public region**. Figure 1–3 shows a public region of an LSM disk that was configured as two subdisks named `dsk2-01` and `dsk2-02`.

Figure 1–3: Multiple Subdisks Using a Public Region



ZK-1684U-AI

1.1.4 Data Plex

A data plex is an object that represents a subdisk or collection of subdisks in the same disk group to which LSM writes volume data.

By default, LSM assigns plex names using the volume name followed by a dash (–) and an ascending two-digit number beginning with 01. For example, `volume1-01` is the name of the first (or only) plex in a volume named `volume1`.

Alternatively, you can assign a plex name of up to 31 alphanumeric characters that cannot include spaces or the slash (/). For example, you can assign a plex name of `finance_plex01`.

You can use one of three types of data plex depending on how you want LSM to store volume data on disk:

- **Concatenated data plex**

In a concatenated data plex, LSM writes volume data in a linear manner. When the space in one subdisk has been written to, the remaining data goes to the next sequential subdisk in the plex. Section 1.1.4.1 explains this plex type in more detail. A volume can contain two or more concatenated data plexes, in which case the volume is described as concatenated and mirrored.

- **Striped data plex**

In a striped data plex, LSM separates data into equal-sized data units (defined by the stripe width) and writes the data units to each disk in the plex. This attempts to balance the load across all disks. Section 1.1.4.2 explains this plex type in more detail. A volume can contain two or more striped data plexes, in which case the volume is described as striped and mirrored.

- **RAID 5 data plex**

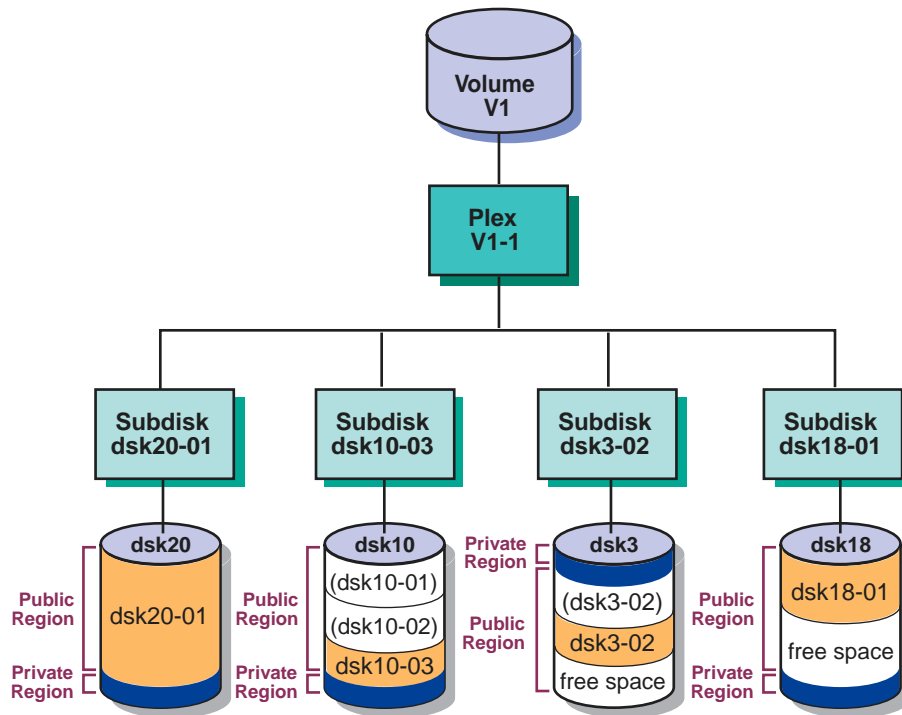
In a RAID 5 data plex, LSM calculates a parity value for the data being written, then separates the data and parity into equal-sized data units (defined by the stripe width), and intersperses the data and parity across

all disks. Section 1.1.4.3 explains this plex type in more detail. A volume can contain only one RAID 5 data plex, due to internal design constraints.

1.1.4.1 Concatenated Data Plex

In a concatenated data plex, LSM creates a contiguous address space on the subdisks and sequentially writes volume data in a linear manner. If LSM reaches the end of a subdisk while writing data, it continues to write data to the next subdisk, which might be on a different physical disk (Figure 1–4). LSM lets you use space on several disks that otherwise might be unusable. One disk's public region can contain subdisks used in several different volumes.

Figure 1–4: Concatenated Data Plex



ZK-1685U-AI

A single subdisk failure in a volume with one concatenated data plex will result in LSM volume failure. To prevent this type of failure, you can create multiple plexes (mirrors) on different disks. LSM continuously maintains the data in the mirrors. If a plex becomes unavailable because of a disk failure, the volume continues operating using another plex.

Using disks on different SCSI buses for mirror plexes speeds read requests, because data can be simultaneously read from multiple plexes.

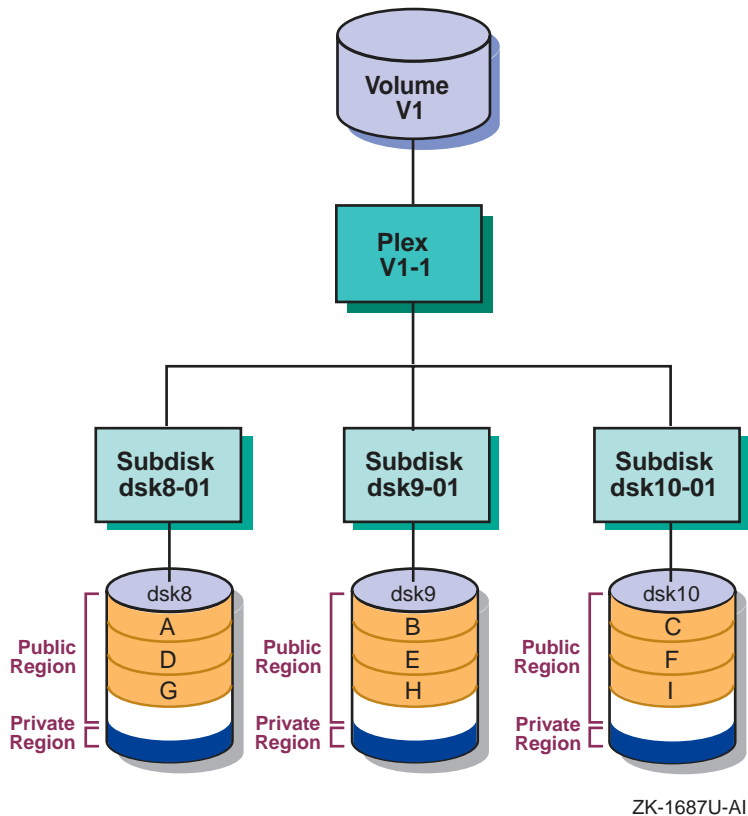
1.1.4.2 Striped Data Plex

In a striped data plex, LSM divides a write request into equal-size data units, defined by the stripe width (64K bytes by default) and writes each data unit to a different disk, creating a stripe of data across the columns (usually, the number of disks in the plex). You can define a different stripe width (data unit size) to achieve the best division of data across the columns.

LSM can simultaneously write two or more data units if the disks are on different SCSI buses.

Figure 1–5 shows a three-column striped plex. In this type of plex, an I/O write request is divided into equal-size units (A, B, C, D, and so on) and each data unit is written sequentially to a different subdisk (in a different disk column).

Figure 1–5: Volume with a Three-Column Striped Data Plex



If a write request does not complete a stripe (the number of data units is not evenly divisible by the number of columns), then the first data unit of the next write request starts in the next column.

If a write request is not evenly divisible by the data unit size, so that the last data unit in a write request does not map to the end of a column, the next write request completes the column then continues to subsequent columns.

As in a concatenated data plex, a single disk failure in a volume with one striped data plex will result in volume failure. To prevent this type of failure, you can create multiple data plexes (mirrors) on different disks. LSM continuously maintains the data in the mirrored data plexes. If a plex becomes unavailable because of a disk failure, the volume continues operating using another plex.

Using disks on different SCSI buses for mirror plexes speeds read requests, because data can be simultaneously read from multiple plexes.

1.1.4.3 RAID 5 Data Plex

In a RAID 5 data plex, LSM calculates a parity value for each stripe of data, then separates the data and parity into equal-size units defined by the stripe width (16K bytes by default) and writes the data and parity units on three or more columns of subdisks, creating a stripe of data and parity across the columns. The parity is contained in one data unit to ensure that each column of disks contains the entire parity value for any given data stripe.

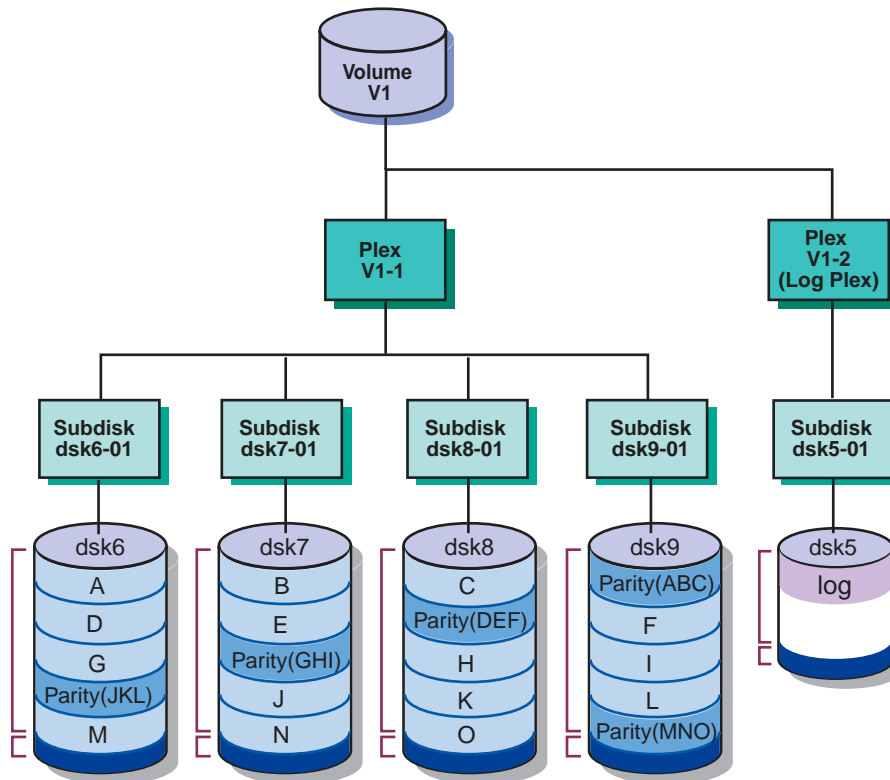
LSM writes the parity in a different column for each consecutive stripe of data. The parity unit for the first stripe is written to the last column. Each successive parity unit is located in the next column to the left of the previous parity unit location. If there are more stripes than columns, the parity unit placement begins again in the last column.

If a disk in one column fails, LSM continues operating using the data and parity information in the remaining columns to reconstruct the missing data. You can define a different stripe width (data unit size) to achieve the best division of data and parity across the columns.

LSM can simultaneously write the data and parity units if the columns are on different SCSI buses.

Figure 1–6 shows how data and parity information are written in a RAID 5 data plex.

Figure 1–6: Volume with a RAID 5 Data Plex



ZK-1690U-AI

1.1.5 Log Plex

A log plex contains information about activity in a volume. After a system failure, LSM recovers only those areas of the volume identified in the log plex as being dirty (written to) at the time of the failure.

By default, LSM creates a log plex for mirrored volumes (volumes with two or more striped or concatenated data plexes) and for volumes that use a RAID 5 data plex. Mirrored volumes use a Dirty Region Log (DRL) plex and an optional Fast Plex Attach (FPA) plex. RAID 5 volumes use a RAID 5 log plex.

- **Dirty Region Log (DRL) plex**

In a DRL plex, LSM keeps track of the regions of a volume that change due to I/O requests. When the system restarts after a crash, LSM resynchronizes only the regions marked as dirty in the log. This greatly reduces the time needed to resynchronize the volume, especially for volumes of hundreds of megabytes or more.

Regions are marked as dirty before the data is written. When the write completes, the region is not immediately marked as clean but instead allowed to stay dirty for a specific length of time. This reduces the overhead of marking the log if another write occurs to the same region. If a dirty region has had no activity for an extended period of time, it is marked as clean.

If you do not use a DRL plex, LSM copies and resynchronizes all the data to each plex to restore the plex consistency when the system restarts after a failure. Although this process occurs in the background and the volume is still available, it can be a lengthy procedure and can result in unnecessarily recovering data, thereby degrading system performance.

- **Fast Plex Attach (FPA) log plex**

A Fast Plex Attach log plex is used to support backups of mirrored volumes. An FPA log tracks the regions of a volume that change while one of its data plexes is detached. The detached plex is used to create a secondary volume for performing backups. When the plex returns to the original volume, only the regions marked in the FPA log plex are written to the returning plex, reducing the time required to resynchronize that plex to the volume.

- **RAID 5 log plex**

In a RAID 5 log plex, LSM stores a copy of the data and parity for several full stripes of I/O. When a write to a RAID 5 volume occurs, the parity is calculated and the data and parity are first written to the RAID 5 log, then to the volume. When the system is restarted after a crash, all the writes in the RAID 5 log are written (or possibly rewritten) to the volume. The RAID 5 log plex uses a special log subdisk.

In addition, for compatibility with Version 4.0, LSM supports a combination data and log plex. This type of plex is not used in Version 5.0 and higher.

1.1.6 LSM Volume

A volume is an object that represents a hierarchy of plexes, subdisks, and LSM disks in a disk group. Applications and file systems make read and write requests to the LSM volume. The LSM volume depends on the underlying LSM objects to satisfy the request.

An LSM volume can use storage from only one disk group.

LSM does not assign default names to volumes; you must assign a name of up to 31 alphanumeric characters that does not include spaces or the slash (/). Within a disk group the volume names must be unique, but two different disk groups can have volumes with the same name.

LSM volumes can be either redundant or nonredundant. A redundant volume provides high data availability, either through mirroring (two or more concatenated or striped data plexes) or through parity (RAID 5 data plex). The following sections describe these properties in more detail.

1.1.6.1 Nonredundant Volumes

A nonredundant volume has one data plex and therefore does not provide any data redundancy. The plex layout can be either striped or concatenated.

A nonredundant volume with one concatenated plex is called a simple volume, which can comprise space on one or more disks. This is the simplest volume type. A simple volume usually has the slowest performance of all the volume types.

1.1.6.2 Mirrored Volumes

A mirrored volume has two or more data plexes, which are either concatenated or striped, and a log plex (by default). Depending on the plex layout, this type of volume is also called a concatenated and mirrored volume or a striped and mirrored volume. Usually, all the data plexes in a volume have the same layout (all striped or all concatenated), but this is not a restriction.

Each data plex is an instance of the volume data. A mirrored volume provides data redundancy and improved read performance, as data can be read from any mirror. Mirrored volumes can have up to 32 plexes in any combination of data and DRL plexes, but by definition mirrored volumes have at least two data plexes. Mirrored volumes are redundant volumes, because each mirror (plex) contains a complete copy of the volume data.

Figure 1–7 shows a volume with concatenated and mirrored data plexes and a Dirty Region Log (DRL) plex (Section 1.1.5).

Figure 1–7: Volume with Concatenated and Mirrored Data Plexes

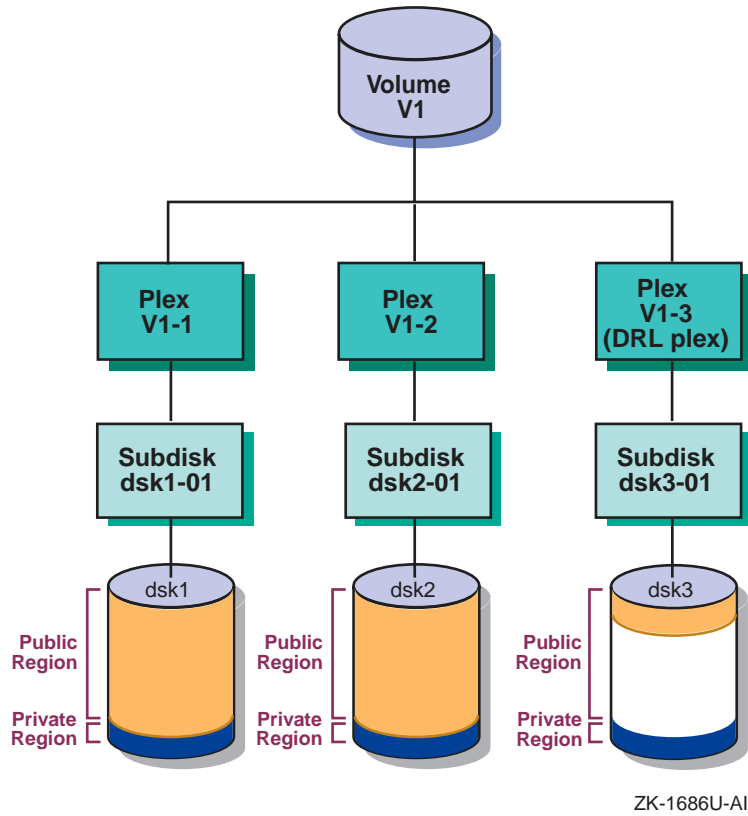
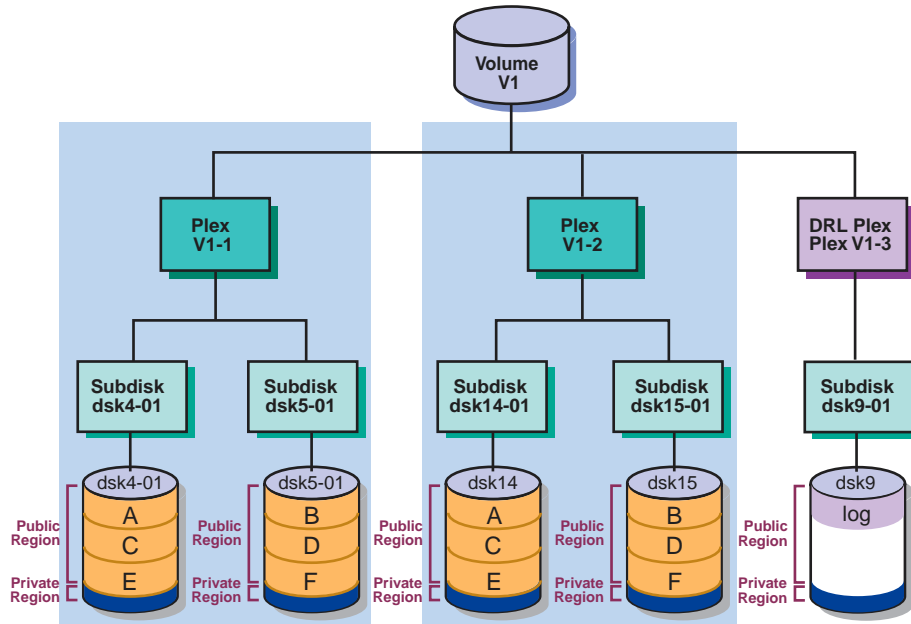


Figure 1–8 shows a volume with striped and mirrored data plexes and a DRL plex.

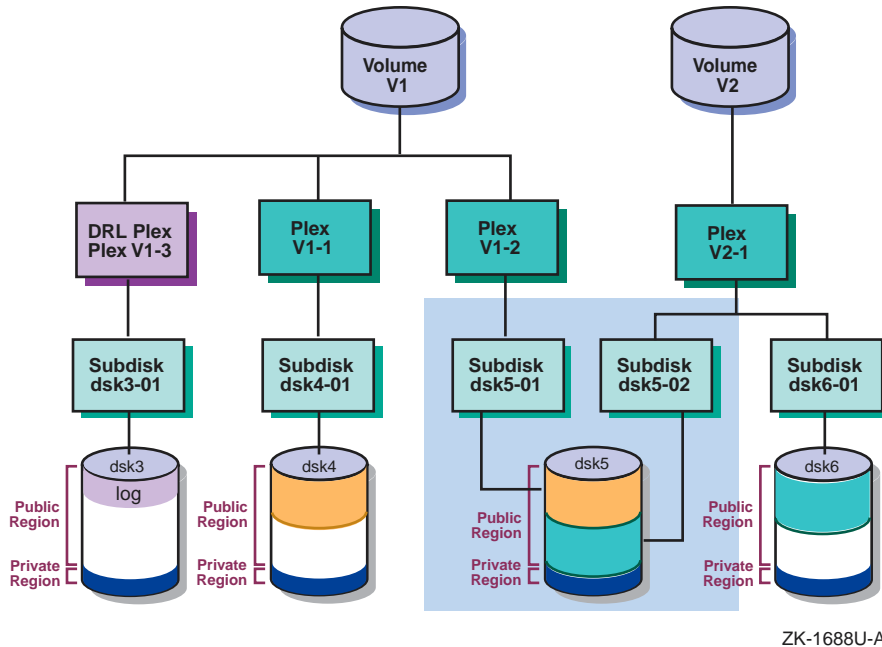
Figure 1–8: Volume with Striped and Mirrored Data Plexes



ZK-1689U-AI

Different LSM volumes can use disk space on the same disk, but in different subdisks.

Figure 1–9: Two LSM Volumes Using Subdisks on the Same Disk



In Figure 1–9, volume V1 uses space on disk dsk5 in subdisk dsk5-01. Volume V2 uses space on disk dsk5 as well, but in subdisk dsk5-02.

Volume V1 is striped and mirrored (uses two striped plexes), and volume V2 is a simple volume (uses one concatenated plex). If disk dsk5 fails, volume V1 continues running using plex V1-1. However, volume V2 will fail completely because it is not redundant.

1.1.6.3 RAID 5 Volumes

A RAID 5 volume has one RAID 5 data plex and one RAID 5 log plex. You can add multiple RAID 5 log plexes to the volume, but one is sufficient. RAID 5 volumes are redundant volumes, because the volume preserves data redundancy through the parity information.

Note

You cannot mirror a RAID 5 data plex.

The TruCluster Server software does not support RAID 5 volumes.

1.1.6.4 Volume Usage Types

An LSM volume has a usage type that defines a particular class of rules for operating on the volume. The rules are typically based on the expected content of the volume. The LSM usage types include:

- `fs` — For volumes that contain file systems. This is the default usage type.
- `gen` — For volumes used for swap space or other applications that do not use the system buffer cache (such as a database).
- `raid5` — For all RAID 5 volumes, regardless of what the volume contains.

In addition, LSM uses the following special usage types:

- `root` — For the `rootvol` volume, created by encapsulating the root partition on a standalone system.
- `swap` — For the primary swap volume, created by encapsulating the primary swap partition on a standalone system. For the swap volumes of cluster members, created by encapsulating the members' swap devices.
- `cluroot` — For the `cluster_rootvol` volume, created by migrating the clusterwide root file system domain to an LSM volume in a TruCluster Server cluster.

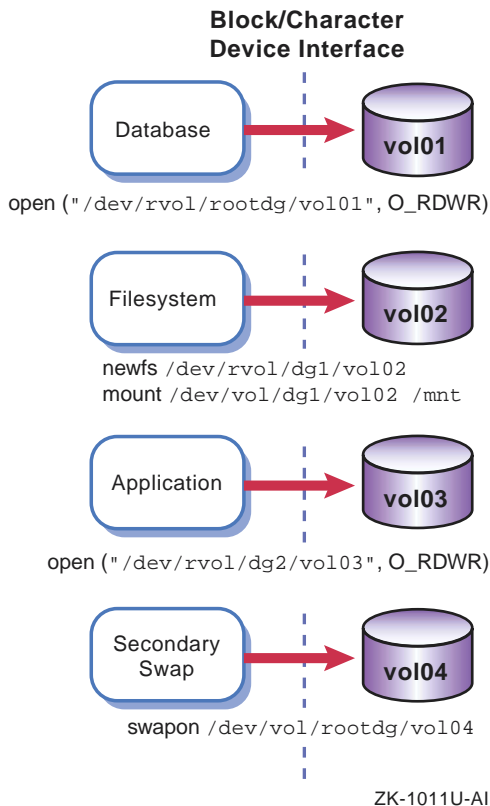
1.1.6.5 Volume Device Interfaces

Like most storage devices, an LSM volume has a block device interface and a character device interface.

- A volume's block device interface is located in the `/dev/vol/disk_group` directory.
- A volume's character device interface is located in the `/dev/rvol/disk_group` directory.

Databases, file systems, applications, and secondary swap use an LSM volume in the same manner as a disk partition because these interfaces support the standard UNIX `open`, `close`, `read`, `write`, and `ioctl` calls (Figure 1–10).

Figure 1–10: LSM Volumes Used Like Disk Partitions



1.2 Overview of LSM Interfaces

You can create, display, and manage LSM objects using one of the following interfaces:

- A command-line interpreter (CLI), where you enter LSM commands at the system prompt. This manual focuses chiefly on LSM CLI commands. The CLI provides the full functionality of LSM. The other interfaces might not support some LSM operations.
- A Java-based graphical user interface (GUI) called LSM Storage Administrator (`lmsa`) that displays a hierarchical view of LSM objects and their relationships.
- A menu-based, interactive interface called `voldiskadm` that supports a limited number of LSM operations on disks and disk groups.

To perform a procedure through the `voldiskadm` interface, you choose an operation from the main menu and the interface prompts you for

information. The `voldiskadm` interface provides default values when possible. You can press Return to use the default value or enter a new value or enter `?` at any time to view online help.

For more information, see `voldiskadm(8)`.

- A bit-mapped GUI called Visual Administrator (`dxlsm`) that uses the Basic X Environment.

The Visual Administrator lets you view and manage disks and volumes and perform limited file system administration. The Visual Administrator displays windows in which LSM objects are represented as icons.

Note

The Visual Administrator (`dxlsm`) has been replaced by the Storage Administrator (`lsmসা`).

For more information, see `dxlsm(8X)`.

In many cases, you can use the LSM interfaces interchangeably. That is, LSM objects created by one interface are usually manageable through and compatible with LSM objects created by other LSM interfaces; however, the Fast Plex Attach feature is available only through the CLI.

1.2.1 LSM Command-Line Interpreter

The LSM command-line interpreter provides you with most control and specificity in creating and managing LSM objects. The other interfaces (`lsmসা`, `voldiskadm`, and `dxlsm`) do not support all the operations available through the command line.

Most LSM commands fall into two categories: high-level and low-level.

- The **high-level commands** are generally more powerful than the low-level commands and are the recommended method of performing the majority of LSM operations. The high-level commands are shortcuts; they might pass the specified operands or values to several low-level commands, initiating many operations with just one command. The high-level commands sequence the intermediate steps in the correct order and also perform a certain amount of error checking by evaluating the operands you specify and the intended end result of the operation, alerting you to problems. The high-level commands use algorithms and default values that provide the best LSM configuration for the majority of cases.

This manual focuses chiefly on the high-level LSM commands, except where more specificity than they provide is required.

- The **low-level commands** require detailed knowledge and understanding of your particular environment and what you are trying to achieve with your LSM configuration. The low-level commands operate on specific LSM object types. In many cases you must perform several operations, sometimes in a precise order, to accomplish what the high-level commands can do for you in fewer steps and with less risk of error.

Table 1–1 lists the LSM commands described in this manual, their functions, and, where applicable, indicates whether the command is considered a high-level or low-level command.

Table 1–1: LSM Commands Described in This Manual

Command	Function	Command Level (If Applicable)
Setup and Daemon Commands		
volsetup	Initializes the LSM software by creating the <code>rootdg</code> disk group.	—
volsave	Backs up the LSM configuration database.	—
volrestore	Restores the LSM configuration database.	—
voldctl, vold, voliod	Controls LSM volume configuration and kernel daemon operations.	—
volwatch	Monitors LSM for failure events and performs hot-sparing if enabled. Used typically only during initial LSM setup to enable the hot-sparing feature.	—
Object Creation and Management Commands		
volassist	Creates, mirrors, backs up, and moves volumes automatically.	High-level — The most-often used LSM command for creating and managing LSM volumes.
voldiskadd	Creates LSM disks and disk groups.	High-level — Performs many of the same functions as <code>voldisksetup</code> and <code>voldg</code> , in one interactive session
voldisksetup	Adds one or more disks for use with LSM.	High-level
voldisk	Administers LSM disks.	Low-level
voldg	Administers disk groups.	High-level
volume	Administers volumes.	Low-level
volplex	Administers plexes.	Low-level

Table 1–1: LSM Commands Described in This Manual (cont.)

Command	Function	Command Level (If Applicable)
volstd	Administers subdisks.	Low-level
volmake	Creates LSM objects manually.	Low-level
voledit	Creates, modifies, and removes LSM records.	Low-level
volrecover	Synchronizes plexes and parity data after a crash or disk failure.	High-level
volmend	Mends simple problems in configuration records.	Low-level
volevac	Evacuates all volume data from a disk.	High-level
<hr/> Data Migration and Encapsulation Commands <hr/>		
volencap	Sets up scripts to encapsulate disks or disk partitions to LSM volumes.	—
volreconfig	Performs the encapsulation scripts set up by volencap, and if necessary restarts the system to complete the encapsulation.	—
volrootmir	Mirrors the root and swap volumes. (Not supported in a cluster.)	—
volunroot	Removes the root and swap volumes. (Not supported in a cluster.)	—
volmigrate, volunmigrate	Migrates AdvFS domains to or from LSM volumes.	—
vollogcnvt	Converts volumes with Block Change Logging (pre-Version 5.0) to Dirty Region Logging (Version 5.0 and higher).	—
<hr/> Informational Commands <hr/>		
volprint	Displays LSM configuration information.	—
voldisk	Displays information about LSM disks.	—
volinfo	Displays volume status information.	—
volstat	Displays LSM statistics.	—
volnotify	Displays LSM configuration events.	—

Table 1–1: LSM Commands Described in This Manual (cont.)

Command	Function	Command Level (If Applicable)
Interface Start-Up Commands		
lsmsa	Starts the LSM Storage Administrator GUI.	—

In addition to commands, LSM includes the `volmake(4)` and `vol_pattern(4)` description files.

For more information on a command, see the reference page corresponding to its name. For example, for more information on the `volassist` command, enter:

```
# man volassist
```

For a list of LSM commands and files, see `volintro(8)`.

1.2.2 Storage Administrator Interface

The Storage Administrator provides dialog boxes in which you enter information to create or manage LSM objects. Completing a dialog box can be the equivalent of entering several commands. The Storage Administrator lets you manage local or remote systems on which LSM is running. You need an LSM license to use the Storage Administrator.

For more information, see Appendix A.

Planning LSM Volumes and Disk Groups

To use LSM most effectively, you should plan your LSM configuration before you use LSM volumes for applications or file systems. Your configuration will depend on the following:

- The number, size, and performance characteristics of available disks and their connectivity to the host system or TruCluster Server cluster.
- The number of disk groups you need and which disks you will configure in each disk group
- The number of volumes you need and their availability and performance requirements

This chapter provides information and worksheets to assist you in planning LSM volumes and disk groups. You can make copies of the worksheets for future use.

2.1 Planning LSM Volumes

Planning LSM volumes includes deciding which attributes you want the LSM volumes to have. An LSM volume has the following attributes:

- **Name** (no default).
- **Size** (no default). You can specify the volume size in sectors (the default unit), kilobytes, megabytes, gigabytes, or terabytes.
- **Plex layout type** (concatenated, striped, or RAID 5). Default: concatenated.
- **Number of plexes**. Default: One concatenated data plex, no log plex.
- **Log plex size**. Default: LSM calculates the appropriate log size based on the volume size.
 - For **mirrored volumes**, the default size for a DRL log plex is approximately 65 blocks per gigabyte of volume size for both standalone systems and clusters, to allow for migration of a disk group or a standalone system running LSM to a TruCluster Server environment.

The minimum DRL size is approximately 2 blocks per gigabyte of volume size. (You can use the minimum size if the LSM configuration will not be used in a cluster.)

- For **RAID 5 volumes**, the default log plex size is large enough to store up through 10 full stripes of data and parity.
- **Disk group.** Default: `rootdg` disk group.
A volume can use disks from only one disk group.
- **Disks used in the volume.** Default: LSM chooses the disks.
 - If the volume uses a striped or RAID 5 plex, each column must be the same size and be on different disks, preferably on different buses.
 - If the volume has mirror plexes, each mirror should use disks on different buses, and the DRL plex should be on a disk that is not used for a data plex.
- **Usage type.** Default: `fsgen`.

Table 2–1 lists the benefits and trade-offs of various plex layout types and describes situations where one plex type might provide better performance, or be more cost effective, than a different plex type. For optimal performance you might need to tune your system to the work load. The layout you choose depends on your specific system configuration, data availability and reliability needs, and application requirements.

Table 2–1: Plex Layout Type Benefits and Trade-Offs

Plex Type	Benefits and Applicability	Trade-Offs
Concatenated	<p>Uses space on multiple disks that might otherwise be wasted.</p> <p>Concatenated plex can be mirrored for data redundancy and high availability.</p> <p>Good for volumes containing infrequently used data or data that does not change often or small volumes that can be confined to a single disk.</p>	<p>Possible uneven performance (hot spots, where one disk is in use by multiple applications).</p> <p>When mirrored, requires 2 through 32 times as much disk space.</p>
Striped	<p>Distributes data and therefore I/O load evenly across many disks.</p> <p>Good for:</p> <ul style="list-style-type: none"> • Large volumes that cannot be confined to a single disk. • Applications with large read-request size. • Volumes that contain data that changes often (many writes). <p>Striping is preferred over RAID 5 in this case, because RAID 5 imposes the overhead of calculating and writing parity data along with volume data.</p> <p>Striped plexes can be mirrored for data redundancy and high availability.</p>	<p>When mirrored, requires 2 through 32 times as much disk space.</p>
RAID 5	<p>Provides the I/O distribution benefit of striping.</p> <p>Provides redundancy through parity, using fewer disks than a volume with mirrored striped plexes.</p> <p>Good for volumes with a high read-to-write ratio.</p>	<p>Depending on the I/O stripe size, performance might be slower than a volume with striped plexes due to parity calculation.</p> <p>A RAID 5 plex requires a minimum of three disks, plus a separate disk for the log plex.</p> <p>The RAID 5 plex type is not supported in a cluster.</p>

When you create an LSM volume with the `volassist` command (the recommended and simplest method), LSM performs all the necessary calculations and creates a volume and log plexes of the appropriate sizes.

Section 2.1.1 and Section 2.1.2 discuss the space requirements for Dirty Region Logs and RAID 5 logs. The worksheets in Section 2.1.3, Section 2.1.4,

Section 2.1.5 and Section 2.2 are provided to help you calculate the space needed and ensure the disk group has enough space for the volumes you want.

2.1.1 Planning Dirty-Region Logs

By default, LSM configures a DRL log subdisk of approximately 65 blocks per gigabyte of volume size for both standalone systems and clusters so that the associated mirrored volume can be used in either a cluster or a standalone system. In a cluster, each member that issues I/O to the volume maintains its own private log to keep track of changes to the volume from that member. The volume's DRL log plex is an aggregate of every member's private log and must therefore be larger than on a standalone system.

If you plan to use LSM strictly on a standalone system, you can use the minimum DRL size, which is 2 blocks per gigabyte of volume size. However, if you create mirrored volumes with the smaller DRL size and then make those volumes available to a cluster (by using the standalone system to create a cluster or by moving the storage to an existing cluster), the DRL logs on the volumes will be disabled, but the volumes themselves will be usable.

Table 2–2 lists the approximate default DRL sizes for various volume sizes.

Table 2–2: Default (Approximate) Sizes of DRL Logs

Volume Size (GB)	DRL Size (blocks)
<=1	65
2	130
3	130
4	195
60–61	2015
62–63	2080
1021	33215
1022–1023	33280
1024	33345

We highly recommend using the default log size. The log is a bitmap that represents regions of the volume (each bit maps to a number of sectors), and the granularity of the bitmap changes if you specify a larger or smaller log size.

- **Specifying a larger log size** results in smaller regions, like finer resolution in a photograph. Each bit represents a smaller number of volume sectors; for example, one bit per MB (2048 sectors). This means

more log I/O overhead, but faster resynchronization after a system crash because fewer regions are resynchronized.

- **Specifying a smaller log size** results in larger regions, like a grainy photograph. Each bit represents a larger number of volume sectors; for example, one bit per 100 MB (204800 sectors). This means less log I/O overhead, but slower resynchronization after a system crash, because more regions are resynchronized.
- **Using no log at all** is like using a log that represents the entire volume (a one-to-one bitmap). When one write operation occurs, the entire volume is dirty and will undergo complete plex resynchronization after a system crash.

For best performance, each volume's DRL plex should be on a different disk from the log plexes for any other volumes. Using the same disk for the DRL plex for many mirrored volumes will slow down that disk and degrade volume performance, because the dirty bits are set before the write actually occurs. (If you are using hardware such as an HSG80 RAID Array, the cache manages and reduces much of the disk contention.)

In addition, if that disk fails, the logs for those volumes are lost. If the system crashes before you replace that disk, those volumes will go through complete plex resynchronization when the system restarts. This is a performance risk, but not a data loss risk.

2.1.2 Planning RAID 5 Logs

LSM calculates the default log size for a RAID 5 volume using a different algorithm than for mirrored volumes. The RAID 5 log is large enough to contain a copy of the full stripe width of data and parity for several writes. When you create a RAID 5 volume using the `volassist` command, the RAID 5 log size is calculated for you and is sized appropriately to the volume.

The formula for the default RAID 5 log size:

```
[10 × (number of columns × stripe width)]
```

For example:

- For a RAID 5 volume with 3 columns (the minimum) and a default stripe width of 16K bytes, the RAID 5 log size is [10 × (3 × 16K bytes)], or 480K bytes.
- For a RAID 5 volume with 5 columns and the same stripe width, the RAID 5 log size is [10 × (5 × 16K bytes)], or 800K bytes.

Volumes that use a RAID 5 plex are not supported in a cluster, so the RAID 5 log size applies only to a standalone system.

For best performance, the RAID 5 log plex should be on its own disk. Using the same disk for several log plexes (for different volumes) will slow down that disk and degrade the performance of those volumes, because the log is written before the data plex is written. In addition, if that disk fails, the logs for those volumes are lost. If the system crashes before you replace that disk, those volumes will go through complete parity and data resynchronization when the system restarts. This is both a performance risk and a data loss risk.

2.1.3 Worksheet for LSM Volumes with Concatenated Plexes

Use the following worksheet to plan LSM volumes that use concatenated plexes.

Figure 2–1: Worksheet for LSM Volumes with Concatenated Plexes

Attribute	Default Values	Chosen Values
Volume name	—	_____
Volume size	—	_____
Number of data plexes	1	_____
DRL plex size (for mirrored volumes)	65 blocks per GB of volume size (default) or 2 blocks per GB of volume size (for standalone systems only)	_____
Number of DRL plexes	1	_____
Disk group name	rootdg	_____
Usage type	fsgen	_____
Total space required	(Volume size × number of plexes) + (DRL size × number of logs)	_____

2.1.4 Worksheet for LSM Volumes with Striped Plexes

Use the following worksheet to plan LSM volumes that use one or more striped plexes.

Figure 2–2: Worksheet for LSM Volumes with Striped Plexes

Attribute	Default Values	Chosen Values
Volume name	—	_____
Volume size	—	_____
Data unit size (stripe width)	64K bytes	_____
Number of columns	Minimum of two, based on number of disks in disk group and the volume size	_____
Number of data plexes	1	_____
DRL plex size (for mirrored volumes)	65 blocks per GB of volume size (default) or 2 blocks per GB of volume size (for standalone systems only)	_____
Number of DRL plexes	1	_____
Disk group name	rootdg	_____
Usage type	fsgen	_____
Total space required	(Volume size × number of plexes) + (DRL size × number of logs)	_____

2.1.5 Worksheet for LSM Volumes with a RAID 5 Plex

Use the following worksheet to plan an LSM volume that uses a RAID 5 plex.

Note

RAID 5 volumes are not supported in a cluster.

Figure 2–3: Worksheet for LSM Volumes with a RAID 5 Plex

Attribute	Default Values	Chosen Values
Volume name	—	_____
Volume size	—	_____
Data unit size (stripe width)	16K bytes	_____
Number of columns (NCOL)	Between 3 and 8 based on number of disks in disk group and the volume size	(Minimum of three) _____
Log plex size	10 × (NCOL × data unit size)	_____
Number of logs (NLOGS)	1	_____
Disk group name	rootdg	_____
Usage type	Must be raid5	raid5
Total space required	(Volume size × NCOL ÷ (NCOL–1)) + (log plex size × NLOGS)	_____

2.2 Planning Disk Groups

On every system or cluster running LSM, there will be one mandatory, default disk group called `rootdg`. You can create additional disk groups to help you manage your storage and to support moving storage (LSM volumes) to a different system. The following sections provide recommendations and explain restrictions for `rootdg` and other disk groups.

2.2.1 Guidelines for the Number and Usage of Disks

Each disk group (including `rootdg`) should have two or more `sliced` or `simple` LSM disks to ensure multiple copies of the disk group's configuration database. Only `sliced` and `simple` disks have a private region, which stores a copy of the disk group's configuration database. By default, LSM maintains a minimum of four active copies of the configuration database in each disk group, distributed among several disks (and, if applicable, among buses) to ensure redundancy. All the copies should be on separate disks. A disk group cannot contain only `nopriv` disks.

The `rootdg` disk group stores information about all disk groups in its configuration database. Each time an LSM object is created, modified, or removed, the configuration database for that disk group and for `rootdg` is updated. Each LSM object (disk, subdisk, plex, and volume) requires one record; two records fit in one sector (512 bytes).

To improve performance, keep the number of disks in `rootdg` to ten or fewer. If possible, use `rootdg` only for the following:

- System-related volumes, such as volumes for the boot disk partitions and primary swap space on a standalone system.
- Volumes for the clusterwide file system domains and the swap device volumes for cluster members.
- Hot-spare disks for the preceding volumes that are mirrored.

Create additional disk groups if you have more than ten disks to place under LSM control or might move a disk group to a different system or cluster.

In each disk group, configure one or more hot-spare disks for each redundant (mirrored or RAID 5) volume in the disk group. Each hot-spare disk should be as large as the largest disk serving a volume.

2.2.2 Guidelines for Connectivity and Availability

A disk group should have storage devices on different buses to improve performance and availability.

Ideally, in a cluster, LSM should be used only with fully shared storage. Storage availability in a cluster increases as more cluster members have

direct access to all disks in a disk group. Availability is highest when all disks in a disk group have connectivity with all cluster members.

The `drdmgr` and `hwmgrr` commands can give you information about which cluster members serve which disks. For a graphical display of the cluster hardware configuration, including active members, buses, storage devices, and their connections, use the `sms` command to invoke the graphical interface for the SysMan Station, and then choose Hardware from the Views menu.

In a cluster, the disks in each disk group should be accessible by all cluster members, so that all members have access to the LSM volumes, regardless of the state of any other member. LSM volumes have the same connectivity when either one of the following conditions is true:

- All disks in the LSM disk group are on the same shared SCSI bus.
- Disks in the LSM disk group are on different shared SCSI buses, but all of those buses are connected to the same cluster members.

Private disk groups (a disk group all of whose disks are connected to the private bus of a single cluster member) are supported, but if that member becomes unavailable, then the cluster loses access to the disk group. Private disk groups are suitable only when a cluster member that the disk group is physically connected to is also the only member that needs access to the disk group.

Avoid configuring a disk group with a combination of local and shared disks or a disk group whose disks are distributed among the private buses of multiple members. Such disk groups are not recommended, because no single member has direct access to all storage in the group.

It is especially important that the disks in `rootdg` are shared. However, this might not always be possible. For example, you can encapsulate cluster members' swap devices to LSM volumes; all swap volumes must belong to `rootdg`, but those devices might be private to their respective member. Avoid creating volumes that use a mix of shared and private disks or a mix of private disks that are connected to different members.

Section 2.2.3 contains a blank worksheet you can use to plan your disk groups. You can make copies and fill in the information on the copies rather than in this manual and keep the completed worksheet with each system running LSM, for your reference. Also, because you can change your LSM configuration at any time, you can make a new copy of the blank worksheet to record your changes.

Include in the worksheet any information that will help you keep track of the purpose of that disk group. For example, you might create a disk group named `finance_dg` whose purpose is to contain one or more volumes that

will be used by a financial application. You might create another disk group named `db1`, which will contain a volume used by a database. Choose a name for the disk group carefully, as it is not easy to rename it when it contains active volumes.

Under **Volume, Plex and Hot-Spare Disk Information**, include the names of all volumes in that disk group, their plex type, which disks belong to which plex, and any hot-spare disks. For information on hot-spare disks, see Section 3.5.

In addition to the information you include in the worksheet, you can print and save the output from the `volprint` command, which shows detail such as the subdisks used in each volume and the sizes of all the LSM objects. For more information, see Section 5.4.1 and `volprint(8)`.

Table 2–3 is an example of what your disk group planning worksheets might look like when complete. This example applies only to a standalone system, not a cluster.

Table 2–3: Completed Disk Group Worksheets for a Standalone System**Disk Group Name:** `rootdg` — Root file system and system disks.

Disks in Group	Bus/LUN	Disk Size	Volume, Plex, and Hot-Spare Disk Information
dsk0	1	4 GB (all)	volumes: <code>rootvol</code> , <code>swapvol</code> , <code>vol-dsk0g</code>
dsk1	1		(for the <code>/usr</code> and <code>/var</code> file systems on
dsk6	2		partition <code>dsk0g</code>)
			<code>rootvol-01(dsk0a)</code>
			<code>rootvol-02 (dsk6a)</code>
			<code>swapvol-01 (dsk0b)</code>
			<code>swapvol-02 (dsk6b)</code>
			<code>vol-dsk0g-01 (dsk0g)</code>
			<code>vol-dsk0g-02 (dsk6g)</code>
dsk7	2	4 GB	hot-spare disk

Disk Group Name: `data_dg` — Database, must be redundant. Contains volume with mirrored striped plexes and a DRL.

Disks in Group	Bus/LUN	Disk Size	Volume, Plex, and Hot-Spare Disk Information
dsk2	1	18 GB (all)	volume: <code>db_vol</code>
dsk3	1		plex: <code>db_vol-01 (dsk2, dsk3)</code>
dsk8	2		plex: <code>db_vol-02 (dsk8, dsk9)</code>
dsk9	2		plex: <code>db_vol-03 (DRL plex, dsk12)</code>
dsk12	3		
dsk13	3	18 GB (all)	hot-spare disks (each on different bus)
dsk20	4		

Disk Group Name: `finance_dg` — Financial application, must be highly available. Contains volume with a RAID 5 plex (read-only application).

Disks in Group	Bus/LUN	Disk Size	Volume, Plex, and Hot-Spare Disk Information
dsk4	1	9 GB (all)	volume: <code>fin_vol</code>
dsk10	2		plex: <code>fin_vol-01 (column 1 — dsk4)</code>
dsk14	3		plex: <code>fin_vol-01 (column 2 — dsk10)</code>
dsk18	4		plex: <code>fin_vol-01 (column 3 — dsk14)</code>
dsk24	5		plex: <code>fin_vol-01 (column 4 — dsk18)</code>
			plex: <code>fin_vol-02 (log plex — dsk24)</code>
dsk5	1	9 GB (all)	hot-spare disks (one on each bus)
dsk11	2		
dsk15	3		
dsk19	4		
dsk25	5		

2.2.3 Worksheet for Planning Disk Groups

You can make copies of the following worksheet for each disk group on your system, including the `rootdg` disk group.

Table 2–4: Worksheet for Disk Group

Disk Group Name:			
Disks in Group	Bus/LUN Number	Disk Size	Volume, Plex, and Hot-Spare Disk Information

2.3 Finding Unused Storage Devices

Unused storage devices are unused disks, partitions, and hardware RAID disks that LSM can initialize as LSM disks, as well as unused LSM disks, initialized for LSM use but not assigned to a disk group.

The following sections describe how to find unused disks, partitions, and LSM disks. See your hardware RAID documentation for information on finding unused hardware RAID disks.

To find unused storage devices, you can use:

- The Disk Configuration Graphical User Interface (GUI) (Section 2.3.1).
- The command-line interpreter interface (Section 2.3.2).
- The `voldisk list` command on a system where LSM is running (Section 2.3.3).

2.3.1 Using the Disk Configuration GUI to Find Unused Disks

To find unused disks using the Disk Configuration GUI, start the Disk Configuration interface using either of the following methods:

- From the system prompt, enter:

```
# /usr/sbin/diskconfig
```
- From the SysMan Applications pop-up menu on the CDE Front Panel:
 1. Choose Configuration.
 2. Double click the Disk icon in the SysMan Configuration folder.

A window titled Disk Configuration on *hostname* is displayed. The window lists the following information for each disk:

- The disk name, such as `dsk10`
- The device model, such as `RZ1CB-CA`
- The bus number for the device

For more information about a disk, double click the list item (or click Configure when a disk is highlighted). The Disk Configuration: Configure Partitions: window is displayed.

This window contains:

- A graphical representation of the disk partitions in a horizontal bar-chart format and disk information such as the disk name, the total size of the disk, and usage information.
- A Partition Table button that you can click to display a bar chart of the current partitions in use, their sizes, and the file system in use.

- A Disk Attributes button that you can click to display values for disk attributes.

For more information about the Disk Configuration GUI, see its online help.

2.3.2 Using Operating System Commands to Find Unused Disks

You can use the following operating system commands to find unused disks:

1. List all the disks on the system:

```
# ls /dev/disk/dsk*c
/dev/disk/dsk0c /dev/disk/dsk26c /dev/disk/dsk42c /dev/disk/dsk59c
/dev/disk/dsk10c /dev/disk/dsk27c /dev/disk/dsk43c /dev/disk/dsk5c
/dev/disk/dsk11c /dev/disk/dsk28c /dev/disk/dsk44c /dev/disk/dsk60c
/dev/disk/dsk12c /dev/disk/dsk29c /dev/disk/dsk45c /dev/disk/dsk61c
/dev/disk/dsk13c /dev/disk/dsk2c /dev/disk/dsk46c /dev/disk/dsk62c
/dev/disk/dsk14c /dev/disk/dsk30c /dev/disk/dsk47c /dev/disk/dsk63c
/dev/disk/dsk15c /dev/disk/dsk31c /dev/disk/dsk48c /dev/disk/dsk64c
/dev/disk/dsk16c /dev/disk/dsk32c /dev/disk/dsk49c /dev/disk/dsk65c
/dev/disk/dsk17c /dev/disk/dsk33c /dev/disk/dsk4c /dev/disk/dsk66c
/dev/disk/dsk18c /dev/disk/dsk34c /dev/disk/dsk50c /dev/disk/dsk67c
/dev/disk/dsk19c /dev/disk/dsk35c /dev/disk/dsk51c /dev/disk/dsk68c
/dev/disk/dsk1c /dev/disk/dsk36c /dev/disk/dsk52c /dev/disk/dsk6c
/dev/disk/dsk20c /dev/disk/dsk37c /dev/disk/dsk53c /dev/disk/dsk7c
/dev/disk/dsk21c /dev/disk/dsk38c /dev/disk/dsk54c /dev/disk/dsk8c
/dev/disk/dsk22c /dev/disk/dsk39c /dev/disk/dsk55c /dev/disk/dsk9c
/dev/disk/dsk23c /dev/disk/dsk3c /dev/disk/dsk56c
/dev/disk/dsk24c /dev/disk/dsk40c /dev/disk/dsk57c
/dev/disk/dsk25c /dev/disk/dsk41c /dev/disk/dsk58c
```

2. To verify that a disk or partition is unused, choose a disk from the output of the previous command and enter the `disklabel` command with the name of the disk; for example:

```
# disklabel dsk20c | grep -p '8 part'
8 partitions:
#          size      offset  fstype  fsize  bsize   cpg   # ~Cyl values
a:         131072         0  unused     0     0     #      0 - 25*
b:         262144    131072  unused     0     0     #     25*- 77*
c:        35556389         0  unused     0     0     #      0 - 6999*
d:           0         0  unused     0     0     #      0 - 0
e:           0         0  unused     0     0     #      0 - 0
f:           0         0  unused     0     0     #      0 - 0
g:        17581586    393216  unused     0     0     #     77*- 3538*
h:        17581587    17974802  unused     0     0     #    3538*- 6999*
```

For more information, see `disklabel(8)`.

3. If you are using AdvFS, display the disks in use by all domains:

```
# ls /etc/fdmns/*/*
```

In a cluster, do not use any partitions (even if unused) on a member's boot disk or on the quorum disk.

4. In a cluster, identify the cluster quorum disk:

```
# clu_quorum
```

5. If you are using UFS, display all mounted file sets:

```
# mount
```

The `drdmgr` and `hwmgr` commands can give you information about which systems serve which disks. To get a graphical display of the cluster hardware configuration, including active members, buses, storage devices, and their connections, use the `sms` command to invoke the graphical interface for the SysMan Station, and then choose Hardware from the Views menu.

2.3.3 Using the LSM voldisk Command to Find Unused Disks

When LSM starts, it obtains a list of disk device addresses from the operating system software and examines the disk labels to determine which devices are initialized for LSM use and which are not.

If LSM is running on the system, you can use the `voldisk` command to display a list of all known disks and to display information about a particular disk:

1. To view a list of disks:

```
# voldisk list
```

DEVICE 1	TYPE 2	DISK 3	GROUP 4	STATUS 5
dsk0	sliced	-	-	unknown
dsk1	sliced	-	-	unknown
dsk2	sliced	dsk2	rootdg	online
dsk3	sliced	dsk3	rootdg	online
dsk4	sliced	dsk4	rootdg	online
dsk5	sliced	dsk5	rootdg	online
dsk6	sliced	dsk6	dgl	online
dsk7	sliced	-	-	online
dsk8	sliced	dsk8	dgl	online
dsk9	sliced	-	-	online
dsk10	sliced	-	-	online
dsk11	sliced	-	-	online
dsk12	sliced	-	-	online
dsk13	sliced	-	-	unknown
dsk14	sliced	-	-	unknown

The following list describes the information in the output.

- 1** Specifies the disk access name.
- 2** Specifies the LSM disk type (`sliced`, `simple`, or `nopriv`). A dash (-) means the device is not under LSM control and therefore does not have an LSM disk type.
- 3** Specifies the LSM disk media name. A dash (-) means the device is not assigned to a disk group and therefore does not have an LSM disk media name.
- 4** Specifies the disk group to which the device belongs. A dash (-) means the device is not assigned to a disk group.

- 5 Specifies the status of the disk according to LSM:
- `unknown` — The disk is not under LSM control.
 - `online` — The disk is under LSM control. If both the `DISK` and `GROUP` columns contain dashes, the disk does not belong to a disk group and is an unused LSM disk.
 - `offline` — The disk is not accessible, either because it failed or because it was intentionally placed off line.
 - `online aliased` — The disk is a hardware clone disk.
 - `error` — The disk is detected but has experienced I/O errors.
 - `failed was` — An LSM disk media name exists but the disk is not associated with a `DEVICE`. Displays the last device associated with this name.

2. To display information about a specific LSM disk:

```
# voldisk list disk
```

For example:

```
# voldisk list dsk5
```

```
Device:      dsk5
devicetag:   dsk5
type:        sliced 1
hostid:      hostname
disk:        name=dsk5 id=942260116.1188.hostname 2
group:       name=dg1 id=951155418.1233.hostname 3
flags:       online ready autoimport imported
pubpaths:    block=/dev/disk/dsk5g char=/dev/rdisk/dsk5g
privpaths:   block=/dev/disk/dsk5h char=/dev/rdisk/dsk5h
version:     n.n
iosize:      min=512 (bytes) max=2048 (blocks)
public:      slice=6 offset=16 len=2046748 4
private:     slice=7 offset=0 len=4096
update:      time=952956192 seqno=0.11
headers:     0 248
configs:     count=1 len=2993
logs:        count=1 len=453
Defined regions:
config  priv   17-   247[  231]: copy=01 offset=000000 enabled
config  priv   249-  3010[ 2762]: copy=01 offset=000231 enabled
log     priv   3011- 3463[  453]: copy=01 offset=000000 enabled
```

- 1 Identifies the LSM disk type.
- 2 The `name=` field contains the disk media name.
- 3 Identifies the disk group.
- 4 The `len=` field shows the amount of available space in blocks on the LSM disk. On a Tru64 UNIX system, one block is 512 bytes.

For more information, see `voldisk(8)`.

Installing and Initializing the LSM Software

This chapter describes how to install and initialize LSM on a standalone system or a TruCluster Server cluster and how to use LSM to provide redundancy for the root file systems and domains for both environments. To upgrade a system using LSM from Version 4.0, see Section 7.1.

3.1 Installing the LSM Software

The LSM software comprises three optional subsets. These are located on the CD-ROM containing the base operating system software for the Tru64 UNIX product kit. You can install the LSM subsets either at the same time or after you install the mandatory operating system software.

- During a full operating system installation, you have the option to install the system's root file system and `/usr`, `/var`, and `swap` partitions directly to LSM volumes. If you choose that option, the LSM subsets are installed automatically.

If you plan to create a cluster from the system, skip that option. The standalone system's root volumes are not used in a cluster.

- An upgrade installation automatically upgrades any subsets currently installed on the system. You do not need to specify the LSM subsets during an upgrade if they are currently installed.

Note

If you are upgrading a system with file systems that use LSM volumes, first boot to single-user mode, start LSM and its volumes, and then proceed with the Tru64 UNIX upgrade installation.

- To install LSM on a running system (not as part of installing or upgrading the operating system), see the Tru64 UNIX *Installation Guide*.

To configure a new cluster with LSM:

1. Install the base operating system (Tru64 UNIX) and the LSM subsets on one system, but do not install the base file system to LSM volumes through the Installation GUI.
2. Create the cluster (`clu_create` command).
3. Initialize LSM (`volsetup` command).
4. Add other cluster members (`clu_add_member` command).
5. Optionally, migrate AdvFS domains, including the `cluster_root`, `cluster_usr`, and `cluster_var` domains to LSM volumes (`volmigrate` command).
6. Optionally, encapsulate the swap devices for the cluster members to LSM volumes (`volencap` command).

Table 3–1 lists the LSM subsets; in the subset name, *nnn* indicates the operating system version.

Table 3–1: LSM Software Subsets

Subset	Function
OSFLSMBIN <i>nnn</i>	Provides the kernel modules to build the kernel with LSM drivers. This software subset supports uniprocessor, SMP, and real-time configurations. This subset requires Standard Kernel Modules.
OSFLSMBASE <i>nnn</i>	Contains the LSM administrative commands and tools required to manage LSM. This subset is mandatory if you install LSM during a full Tru64 UNIX installation. This subset requires LSM Kernel Build Modules.
OSFLSMX11 <i>nnn</i>	Contains the LSM Motif-based graphical user interface (GUI) management tool and related utilities. This subset requires the Basic X Environment.

3.2 Installing the LSM License

The base operating system comes with a base LSM license, which lets you create LSM volumes that use a single concatenated plex (simple volumes). All other LSM features, such as the ability to create LSM volumes with striped, mirrored, and RAID 5 plexes, and the ability to use the LSM GUIs, require a separate LSM license.

This separate LSM license is supplied in the form of a product authorization key (PAK) called `LSM-OA`.

To install the LSM license, load the `LSM-OA` PAK into the Tru64 UNIX License Management Facility (LMF).

If you need to order an LSM license, contact your service representative. For more information on the License Management Facility, see `lmf(8)`.

3.3 Initializing LSM

LSM is automatically initialized when you perform a full operating system installation and choose the option to install the root file system and `/usr`, `/var`, and `swap` partitions directly to LSM volumes, or if you perform an upgrade installation on a system or cluster that was previously running LSM. Otherwise, you must initialize LSM manually.

Initializing LSM does the following:

- Creates the `rootdg` disk group.
- Reestablishes an existing LSM configuration, if found.
- Adds entries to the `/etc/inittab` file to automatically start LSM when the system or cluster restarts.
- Creates the `/etc/vol/volboot` file, which contains the host ID.
- Creates LSM files and directories. (For a description of these files and directories, see Section 3.6.)
- Starts the `vold` and `voliod` daemons.

To initialize LSM, you need at least two unused disks or partitions to create the `rootdg` disk group to ensure there are multiple copies of the LSM configuration database. See Chapter 2 if you need help choosing disks or partitions for the `rootdg` disk group. In a cluster, do not use unused partitions on the quorum disk or on any member's private boot disk.

To initialize LSM on standalone systems and clusters:

1. Verify (on any member in a cluster) that the LSM subsets are installed:

```
# setld -i | grep LSM
```

The following information is displayed, where *nnn* indicates the operating system revision:

```
OSFLSMBASEnnn installed Logical Storage Manager (System Administration)
OSFLSMBINnnn installed Logical Storage Manager Kernel Modules (Kernel
Build Environment)
OSFLSMX11nnn installed Logical Storage Manager GUI (System Administration)
```

If the LSM subsets do not show a status of `installed`, use the `setld` command to install them. For more information on installing software subsets, see the *Installation Guide*.

2. Verify LSM drivers are configured into the kernel:

```
# devswmgr -getnum driver=LSM
Device switch reservation list
```

driver name	instance	major
-----	-----	-----
LSM	4	43
LSM	3	42
LSM	2	41*
LSM	1	40*

If LSM driver information is not displayed, you must rebuild the kernel using the `doconfig` command. For more information on rebuilding the kernel, see the *Installation Guide*.

- Initialize LSM, specifying at least two disks or partitions:

```
# volsetup {disk|partition} {disk|partition...}
```

For example, to initialize LSM with disks `dsk4` and `dsk5`:

```
# volsetup dsk4 dsk5
```

If you omit a disk or partition name, the `volsetup` script prompts you for it. If the `volsetup` command displays an error message that the initialization failed, you might need to reinitialize the disk. For more information about reinitializing a disk, see the Disk Configuration GUI online help.

When LSM is initialized on a Tru64 UNIX system, the LSM configuration is propagated to the cluster when the cluster is created with the `clu_create` command and when members are added with the `clu_add_member` command.

- In a cluster, synchronize LSM throughout the cluster by entering the following command on all the other current members (except the one on which you performed step 3):

```
# volsetup -s
```

If you subsequently add a new member to the cluster with the `clu_add_member` command, LSM is automatically synchronized on the new member. Do not run the `volsetup -s` command on the new member.

Normally, you do not need to verify that LSM was initialized. If the initialization fails, the system displays error messages indicating the problem.

To verify that LSM is initialized, do one or more of the following:

- To verify that the `rootdg` disk group exists:

```
# volprint
```

```
Disk group: rootdg
```

```
TY NAME    ASSOC    KSTATE  LENGTH  PLOFFS  STATE    TUTILO  PUTILO
```

```

dg rootdg rootdg - - - - -
dm dsk4 dsk4 - 1854536 - - - -
dm dsk5 dsk5 - 1854536 - - - -

```

In this example, dsk4 and dsk5 are part of the rootdg disk group.

- To verify that the `/etc/inittab` file was modified to include LSM entries:

```

# grep LSM /etc/inittab
lsmr:s:sysinit:/sbin/lsmbootstrap -b /dev/console 2>&1 ##LSM
lsm:23:wait:/sbin/lsmbootstrap -n /dev/console 2>&1 ##LSM
vol:23:wait:/sbin/vol-reconfig -n /dev/console 2>&1 ##LSM

```

- To verify that the `/etc/vol/volboot` file was created:

```

# /sbin/voldctl list
Volboot file
version: 3/1
seqno: 0.4
hostid: hostname
entries:

```

- To verify that the `vold` daemon is enabled:

```

# voldctl mode
mode: enabled

```

- To verify that two or more `voliod` daemons are running:

```

# voliod
2 volume I/O daemons are running

```

By default, LSM initiates one daemon for each CPU in the system or a minimum of two.

3.4 Using LSM for Critical File Systems and Swap Space

After you initialize LSM you can:

- Encapsulate the root file system and primary swap space on a standalone system (Section 3.4.1)
- Migrate the clusterwide root, `/usr`, and `/var` file system domains to LSM volumes (Section 3.4.2)
- Encapsulate cluster members' swap devices to LSM volumes (Section 3.4.3)
- Enable the hot-sparing feature of LSM to provide disk failover (Section 3.5), and configure hot-spare disks for each disk group (Section 3.5.1)

3.4.1 Creating Alternate Boot Disks (Standalone System)

You can use LSM to create an alternate boot disk for a standalone system by encapsulating the boot disk partitions to LSM volumes and mirroring those volumes. This copies the data in the boot disk partitions to another disk, which provides complete redundancy and recovery capability if the boot disk fails. If the primary boot disk fails, the system continues running off the surviving mirror, which is on a different disk. You can also reboot the system using the surviving mirror.

To create an alternate boot disk with LSM:

1. Use the LSM encapsulation procedure to configure each root file system partition and the primary swap space to use LSM volumes.
2. Add a mirror plex to the volumes to create copies of the data in the boot disk partitions.

Note

To facilitate recovery of environments that use LSM, you can use the bootable tape utility. This utility enables you to build a bootable standalone system kernel on magnetic tape. The bootable tape preserves your local configuration and provides a basic set of the LSM commands you will use during restoration. For more information on the SysMan Menu `boot_tape` option, see the *System Administration* manual or the online help, and `btcreate(8)`.

3.4.1.1 Restrictions and Requirements

The following restrictions apply when you encapsulate the system partitions and primary swap space:

- The system cannot be part of a TruCluster cluster.
To create LSM volumes for the clusterwide root, `/usr`, and `/var` file system domains, see Section 3.4.2.
- You must encapsulate the root file system and the primary swap space partition at the same time. They do not have to be on the same disk.
- The LSM volumes are created in the `rootdg` disk group and have the following names:
 - `rootvol` — Assigned to the volume created for the root file system. Do not change this name, move the `rootvol` volume out of the `rootdg` disk group, or change the assigned minor device number of 0.

- `swapvol` — Assigned to the volume created for the swap space partition. Do not change this name, move the `swapvol` volume out of the `rootdg` disk group, or change the assigned minor device number of 1.
- All other partitions are assigned an LSM volume name based on the original partition name; for example, `vol-dsk0g`.

The following disk requirements apply:

- **Original Boot Disk** — The partition table for the boot disk (and primary swap disk, if different) must have at least one unused partition for the LSM private region, which cannot be the `a` or `c` partition.

The unused partition does not have to have any space in it; if necessary, LSM takes the required space (4096 blocks by default) from the swap space and relabels the disk partitions accordingly.

If there is no space (or not enough) on any unused partition for LSM to use, the encapsulation fails.

- **Mirror Disk**
 - If the primary swap space is on the boot disk, you need one separate disk to create the mirrors for the boot partition and swap space volumes. This disk cannot be under LSM control, must have a disk label with all partitions marked `unused`, and must be as large as the total of the root file system and swap partitions on the primary boot disk (the partitions being mirrored), plus the size of the private region (4096 blocks by default).
 - If the primary swap space is on a separate disk, you need two separate disks to create the mirrors for the boot partition and swap space volumes. These disks cannot be under LSM control and all partitions in their disk labels must be marked `unused`. For more information, see `disklabel(8)`.
 - The disk for the mirror for the boot partition volume must be as large as the total of the root file system partitions and the private region (4096 blocks by default).
 - The disk for the mirror for the swap volume must be as large as the total of the swap partition and the private region (4096 blocks by default).

3.4.1.2 Encapsulating System Partitions (Creating System Volumes)

When you encapsulate the system partitions, each partition is converted to an LSM volume with a single concatenated plex. The steps to encapsulate the system partitions are the same whether you are using the UNIX File System (UFS) or the Advanced File System (AdvFS).

Note

The encapsulation procedure requires that you restart the system.

The encapsulation process changes the following files:

- **For AdvFS**, the links in the `/etc/fdmns/*` directory for domains associated with the root disk are changed to use LSM volumes instead of disk partitions.
- **For UFS**, the `/etc/fstab` file is changed to use LSM volumes instead of disk partitions.
- **For swap space**, in the `/etc/sysconfigtab` file the `swapdevice` entry is changed to use the LSM `swapvol` volume and the `lsm_rootdev_is_volume` entry is set to 1.

In addition, LSM creates a private region and stores in it a copy of the configuration database. If the system partitions are on different disks (for example, the boot partitions on `dsk0` and the swap partition on `dsk1`), LSM creates a private region on each disk. Normally, when you encapsulate a disk or partition, LSM creates only an LSM `nopriv` disk for the area being encapsulated. However, because of the need to be able to boot the system even if the rest of the LSM configuration is corrupted or missing, LSM creates these special-case private regions.

To encapsulate the system partitions:

1. Log in as `root`.
2. Identify the boot disk:

```
# consvar -l | grep boot
boot_dev = dsk0
bootdef_dev = dsk0
booted_dev = dsk0
boot_file =
booted_file =
boot_osflags = A
booted_osflags = A
boot_reset = OFF
```

3. Identify the primary swap disk:

```
# swapon -s
Swap partition /dev/disk/dsk0b (default swap):
:
```

4. Verify that there is at least one unused partition on the boot disk other than `a` or `c`:

```
# disklabel dsk0 | grep -p '8 part'
```

```

8 partitions:
#          size      offset   fstype  [fsize bsize  cpg]  # NOTE: values not exact
a:    262144         0   AdvFS          # (Cyl.   0 - 115*)
b:    262144    262144   swap          # (Cyl. 115*- 231*)
c:   8380080         0  unused         0    0          # (Cyl.   0 - 3707)
d:     4096    8375984  unused         0    0          # (Cyl. 3706*- 3707)
e:   2618597    3142885  unused         0    0          # (Cyl. 1390*- 2549*)
f:   2614502    5761482  unused         0    0          # (Cyl. 2549*- 3706*)
g:   1433600     524288  AdvFS          # (Cyl.  231*- 866*)
h:   6418096    1957888  unused         0    0          # (Cyl. 866*- 3706*)

```

5. If the swap partition is on a separate disk, repeat step 4, specifying the swap disk name.
6. Encapsulate the boot disk and the swap disk, specifying both if different; for example:

```

# volencap dsk0 dsk4
Setting up encapsulation for dsk0.
- Creating simple disk dsk0d for config area (privlen=4096).
- Creating nopriv disk dsk0a for rootvol.
- Creating nopriv disk dsk0b.
- Creating nopriv disk dsk0g.

Setting up encapsulation for dsk4.
- Creating simple disk dsk4h for config area (privlen=4096).
- Creating nopriv disk dsk4b for swapvol.

```

The following disks are queued up for encapsulation or use by LSM:
dsk0d dsk0a dsk0b dsk0g dsk4h dsk4b

You must now run `/sbin/volreconfig` to perform actual encapsulations.

7. If appropriate, send a warning to your user community to alert them of the impending shutdown of the system.

When there are no users on the system, proceed with step 7.

8. Complete the encapsulation process. Enter **now** when prompted to shut down the system:

```

# volreconfig
The system will need to be rebooted in order to continue with
LSM volume encapsulation of:
dsk0d dsk0a dsk0b dsk0g
dsk4h dsk4b

```

Would you like to either quit and defer encapsulation until later or commence system shutdown now? Enter either 'quit' or time to be used with the `shutdown(8)` command (e.g., quit, now, 1, 5): [quit] **now**

The system shuts down, completes the encapsulation, and automatically reboots:

```
Encapsulating dsk0d.  
Encapsulating dsk0a.  
Encapsulating dsk0b.  
Encapsulating dsk0g.  
Encapsulating dsk4h.  
Encapsulating dsk4b.  
Shutdown at 14:36 (in 0 minutes) [pid 11708]
```

```
*** FINAL System shutdown message from root@hostname ***
```

```
System going down IMMEDIATELY  
... Place selected disk partitions under LSM control.
```

```
System shutdown time has arrived
```

3.4.1.3 Mirroring System Volumes

After you encapsulate the boot disk partitions and swap space to LSM volumes, mirror the volumes to provide redundancy. This process might take a few minutes, but the root file system and swap space are available during this time.

Mirror all the system volumes at the same time. The `-a` option in the following procedure does this for you; to mirror only the volumes for the root file system and primary swap but not any other volumes on the boot disk (such as the volumes for the `/usr` and `/var` file systems), omit the `-a` option.

- If the swap space is on the boot disk, enter:

```
# volrootmir -a target_disk
```

For example:

```
# volrootmir -a dsk4
```

This creates a mirror on `dsk4` for all the volumes on the boot disk.

- If the swap space is on a separate disk, enter:

```
# volrootmir -a swap=swap_target_disk boot_target_disk
```

For example:

```
# volrootmir -a swap=dsk5 dsk4
```

This creates a mirror on `dsk4` for all the volumes on the boot disk and creates a mirror on `dsk5` for the swap volume.

Note

This procedure does not add a log plex (DRL) to the root and swap volumes; logging is not supported on the root volume, and there is no need for a log on the swap volume. Attaching a log plex

degrades `rootvol` and `swapvol` performance and provides no benefit as the log is not used during system recovery.

3.4.1.4 Displaying Information for System Volumes

If you followed the steps in Section 3.4.1.3 and did not receive error messages, you can assume that the operation was successful. However, you can also display the results using the following commands:

- Display simplified volume information:

```
# volprint -pt
```

PL NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WID	MODE
pl rootvol-01	rootvol	ENABLED	ACTIVE	262144	CONCAT	-	RW
pl rootvol-02	rootvol	ENABLED	ACTIVE	262144	CONCAT	-	RW
pl swapvol-01	swapvol	ENABLED	ACTIVE	333824	CONCAT	-	RW
pl swapvol-02	swapvol	ENABLED	ACTIVE	333824	CONCAT	-	RW
pl vol-dsk0g-01	vol-dsk0g	ENABLED	ACTIVE	1450796	CONCAT	-	RW
pl vol-dsk0g-02	vol-dsk0g	ENABLED	ACTIVE	1450796	CONCAT	-	RW

In this example, three volumes, named `rootvol`, `swapvol`, and `vol-dsk0g`, contain the `/usr` and `/var` file systems. Each volume has two plexes (listed in the column labeled `PL NAME`), indicating that the volumes were successfully mirrored.

- Confirm that the system recognizes both boot devices:

```
# consvar -l
auto_action = HALT
boot_dev = dsk0,dsk4
bootdef_dev = dsk0,dsk4
booted_dev = dsk0
:
```

- If you are using AdvFS, confirm that the domains point to the LSM volumes instead of the original partitions by doing one of the following:

- Display a simplified listing of the domains:

```
# ls -R /etc/fdmns/*
```

The following information is displayed, showing the volume name (`rootdg.volume`) instead of the original disk partition:

```
/etc/fdmns/root_domain:
rootdg.rootvol
```

```
/etc/fdmns/usr_domain:
rootdg.vol-dsk0g
```

- Display the full domain detail:

1. Change to the `fdmns` directory:

```
# cd /etc/fdmns
```

2. Display attributes of all AdvFS domains:

```
# showfdmn *
```

The following information is displayed, showing the volume name for each AdvFS domain:

```
      Id                Date Created  LogPgs  Version  Domain Name
3a5e0785.000b567c  Thu Jan 11 14:20:37 2001    512      4  root_domain

Vol  512-Blks      Free  % Used  Cmode  Rblks  Wblks  Vol Name
1L   524288      339936   35%   on    256   256   /dev/vol/rootdg/rootvol

      Id                Date Created  LogPgs  Version  Domain Name
3a5e078e.000880dd  Thu Jan 11 14:20:46 2001    512      4  usr_domain

Vol  512-Blks      Free  % Used  Cmode  Rblks  Wblks  Vol Name
1L   2879312     1703968   41%   on    256   256   /dev/vol/rootdg/vol-dsk0g

      Id                Date Created  LogPgs  Version  Domain Name
3a5e0790.0005b501  Thu Jan 11 14:20:48 2001    512      4  var_domain

Vol  512-Blks      Free  % Used  Cmode  Rblks  Wblks  Vol Name
1L   2879312     2842160    1%   on    256   256   /dev/vol/rootdg/vol-dsk0h
```

- If you are using UFS, enter:

```
# mount
/dev/vol/rootdg/rootvol on / type ufs (rw)
/dev/vol/rootdg/vol-dsk0g on /usr type ufs (rw)
/proc on /proc type procfs (rw)
```

- Display the swap volume information:

```
# swapon -s
```

The following information is displayed, showing the volume name (/dev/vol/rootdg/volume_name) instead of the original disk partition:

```
Swap partition /dev/vol/rootdg/swapvol (default swap):
  Allocated space:      25600 pages (200MB)
  In-use space:         426 pages ( 1%)
  Free space:           25174 pages ( 98%)
```

```
Total swap allocation:
  Allocated space:      25600 pages (200.00MB)
  Reserved space:       9015 pages ( 35%)
  In-use space:         426 pages ( 1%)
  Available space:      16585 pages ( 64%)
```

3.4.2 Using LSM Volumes for Cluster Domains

In a TruCluster Server environment, you can use LSM volumes for AdvFS domains, including the clusterwide root, /usr, and /var file system domains (`cluster_root`, `cluster_usr`, and `cluster_var`), and for cluster members' swap devices. You cannot use LSM volumes for the quorum disk, or for any partitions on members' private boot disks.

Note

Using an LSM volume for the `cluster_root` domain does not help with cluster recovery after a failure and does not create a disaster-tolerant cluster. The cluster does not boot from the `cluster_root` file system.

LSM provides high availability for data but does not provide overall cluster availability or reduce recovery time after a cluster failure. LSM is not a disaster-tolerant solution.

LSM provides the following methods for placing a cluster's AdvFS domains and swap devices under LSM control:

- The `volmigrate` command creates LSM volumes on the disks that you specify, moves the AdvFS domain to the volumes, and removes the original disk from the domain and leaves it unused.

The advantage to migrating is that the migration occurs while the domain's filesets are mounted, and no reboot is required. The disadvantage is that the migration process temporarily uses additional disk space while the domain data is copied to the LSM volume.

Note

The `volmigrate` command uses the AdvFS `addvol` command. You need the AdvFS Utilities license PAK to use the `addvol` command.

- The `volencap` command creates LSM volumes on the same disks or disk partitions that the AdvFS domain or swap space is currently using.

The advantage to encapsulating is that no extra disk space is required. The disadvantage is that if the domain is mounted and cannot be unmounted, you need to shut down and reboot the cluster or cluster member for the encapsulation to complete.

Table 3-2 lists when you can use each command.

Table 3–2: LSM Commands for Providing Cluster Redundancy

Domain or Partition	volmigrate Command	volencap Command
cluster_root, cluster_usr, cluster_var	Yes (Section 3.4.2.1)	No
Other AdvFS domains (application data)		Yes (Section 4.6.2)
Member's swap partitions	No	Yes (Section 3.4.3)
Member's private boot partitions (rootmemberID_domain#root)	No	No

3.4.2.1 Migrating AdvFS Domains to LSM Volumes

In a cluster, you can use the `volmigrate` command to migrate any AdvFS domain to an LSM volume except individual members' root domains (`rootmemberID_domain#root`). For example, you can migrate the clusterwide root, `/usr`, and `/var` file system domains to LSM volumes, allowing you to mirror the volumes for high availability.

The `volmigrate` command operates on AdvFS domain names. Within this procedure, the clusterwide file systems are referred to by the default AdvFS domain names `cluster_root`, `cluster_usr`, and `cluster_var`.

The `volmigrate` command is a shell script that calls several other commands to:

- Create an LSM volume for the AdvFS domain on the LSM disk or disks that you specify.
You can specify properties for the LSM volume such as striping and mirroring.
- Add the LSM volume to the domain being migrated, with the AdvFS `addvol` command (requires the AdvFS Utilities license).
- Migrate the data from the original disk partition to the LSM volume.
- Remove the original disk partition from the domain with the AdvFS `rmvol` command (requires the AdvFS Utilities license), and set the disk label partition table entry for that partition to `unused`.

3.4.2.1.1 Disk Space Considerations

If you have limited available disks but want the benefits of mirroring, you can mirror the volumes for the AdvFS domains to the original disk or disks after the migration is complete; however, when you place the original disk under LSM control, its usable space is reduced by 4096 blocks (2 MB) for the LSM private metadata. Therefore, depending on the size and usage of the

original disk, you might have to migrate one or more of the AdvFS domains to a volume smaller than the domain:

- If one domain uses the whole disk (the `c` partition), migrate the domain to a volume 2 MB smaller than the domain.
- If several domains are on the same disk, and if the disk is not at least 2 MB larger than the total size of the domains, migrate one domain to a volume 2 MB smaller than the domain.

Decide which domain to reduce based on disk usage and expected growth.

By default, the `volmigrate` command creates an LSM volume the same size as the AdvFS domain. However, you can specify a smaller volume size, within the restrictions described in `volmigrate(8)`. In effect, this shrinks the domain.

The volume size you specify must be at least 10 percent larger than the in-use portion of the domain. Reduce the volume by only the 2 MB necessary. Both LSM and AdvFS allow you to add space to an LSM volume or AdvFS domain later.

3.4.2.1.2 Migrating AdvFS Domains

The general syntax for migrating an AdvFS domain to an LSM volume is as follows:

```
volmigrate [-g disk_group] [-m num_mirrors] [-s num_stripes] \  
domain_name disk_media_name...
```

You can run the `volmigrate` command from any cluster member. Depending on the size of the domain, the migration might take several minutes to complete. Unless you see an error message, the migration is progressing.

If you plan to reuse the original disk or disk partition for the mirror, migrate the domain to an unmirrored volume (optionally specifying the volume size, as discussed in Section 3.4.2.1.1), place the original disk under LSM control, and then mirror the volume in a separate step.

If possible, do not use the same disk to support multiple volumes, because this increases the number of volumes at risk if the disk fails. Mirroring the volumes and configuring hot-spare disks reduces this risk. If you have available disks, consider migrating each AdvFS domain to its own disk, use the original disk to mirror the domain's volume, and use other disks to mirror the other domains' volumes.

As you migrate a domain, you can specify:

- The disk group in which to create the volume. The default is `rootdg`.
The volume for the `cluster_root` domain must belong to the `rootdg` disk group.

- The name of the volume.
The default is the name of the domain with the suffix `vol`.
- Stripe and mirror attributes.
LSM does not add a DRL to the volume for the `cluster_root` domain, if mirrored. This volume does not need to be resynchronized after a system failure, so does not need a DRL. All other domain volumes, if mirrored, will have a DRL enabled by default.
- The size of the volume.
By default, the volume will be the same size as the domain. If necessary, you can create a volume smaller (or larger) than that, within the restrictions described in `volmigrate(8)`.

The following procedure migrates the `cluster_root`, `cluster_usr`, and `cluster_var` domains but can apply to any other eligible AdvFS domain.

To migrate AdvFS domains to LSM volumes:

1. Display the attributes (specifically, sizes and names) of the AdvFS domains:

```
# cd /etc/fdmns
# showfdmn *
```

Information similar to the following is displayed. This example shows only the `cluster_root`, `cluster_usr` and `cluster_var` domains, for simplicity.

```

      Id                Date Created  LogPgs  Version  Domain Name
3c7d2bd9.0001fe4b  Wed Feb 27 13:56:25 2002      512      4  cluster_root

  Vol  512-Blks      Free  % Used  Cmode  Rblks  Wblks  Vol Name
  1L   401408      211072   47%   on    256    256  /dev/disk/dsk3b

      Id                Date Created  LogPgs  Version  Domain Name
3c7d2bdb.0004b3c9  Wed Feb 27 13:56:27 2002      512      4  cluster_usr

  Vol  512-Blks      Free  % Used  Cmode  Rblks  Wblks  Vol Name
  1L   1787904      204480   89%   on    256    256  /dev/disk/dsk3g

      Id                Date Created  LogPgs  Version  Domain Name
3c7d2bdd.0008401f  Wed Feb 27 13:56:29 2002      512      4  cluster_var

  Vol  512-Blks      Free  % Used  Cmode  Rblks  Wblks  Vol Name
  1L   1790096      1678608   6%   on    256    256  /dev/disk/dsk3h
```

The `cluster_root` domain is 196 MB (401408 blocks). Each block on the Tru64 UNIX system is 512 bytes. The total for all three domains is 3979408 blocks (approximately 2 GB). If you plan to migrate all three domains to the same LSM disk, the disk must have that much free space.

2. Display the disks in the `rootdg` disk group to find simple or sliced disks:

```
# voldisk -g rootdg list
```

DEVICE	TYPE	DISK	GROUP	STATUS
dsk2	sliced	dsk2	rootdg	online
dsk8	sliced	dsk8	rootdg	online
dsk9	sliced	dsk9	rootdg	online
dsk10	sliced	dsk10	rootdg	online
dsk11	sliced	dsk11	rootdg	online
dsk12	sliced	dsk12	rootdg	online
dsk13	sliced	dsk13	rootdg	online
dsk14	sliced	dsk14	rootdg	online
dsk20	sliced	dsk20	rootdg	online
dsk21	sliced	dsk21	rootdg	online
dsk22	sliced	dsk22	rootdg	online
dsk23	sliced	dsk23	rootdg	online
dsk24	sliced	dsk24	rootdg	online
dsk25	sliced	dsk25	rootdg	online

3. Display the free space available in the rootdg disk group:

```
# voldg -g rootdg free
```

DISK	DEVICE	TAG	OFFSET	LENGTH	FLAGS
dsk2	dsk2	dsk2	0	4106368	-
dsk8	dsk8	dsk8	0	4106368	-
dsk9	dsk9	dsk9	0	4106368	-
dsk10	dsk10	dsk10	0	4106368	-
dsk11	dsk11	dsk11	0	4106368	-
dsk12	dsk12	dsk12	0	35552277	-
dsk13	dsk13	dsk13	0	35552277	-
dsk14	dsk14	dsk14	0	35552277	-
dsk20	dsk20	dsk20	0	35552277	-
dsk21	dsk21	dsk21	0	35552277	-
dsk22	dsk22	dsk22	0	35552277	-
dsk23	dsk23	dsk23	0	35552277	-
dsk24	dsk24	dsk24	0	35552277	-
dsk25	dsk25	dsk25	0	35552277	-

4. Choose sliced or simple disks with enough free space to create a volume with the characteristics you want (such as mirrored or striped).

If possible, choose disks with an offset of 0. Avoid using disks with an offset, because these are in use by another volume.

5. Confirm that each disk you want to use is accessible by all cluster members:

```
# hwmgr view devices -cluster
```

Information similar to the following is displayed (edited for brevity):

HWID:	Device Name	Mfg	Model	Hostname	Location
:					
:					
68:	/dev/disk/dsk3c	DEC	RZ28D (C)	DEC moe	bus-2-targ-1-lun-0
68:	/dev/disk/dsk3c	DEC	RZ28D (C)	DEC larry	bus-2-targ-1-lun-0
68:	/dev/disk/dsk3c	DEC	RZ28D (C)	DEC curly	bus-2-targ-1-lun-0
:					
:					
73:	/dev/disk/dsk8c	DEC	RZ28D (C)	DEC moe	bus-3-targ-10-lun-0
73:	/dev/disk/dsk8c	DEC	RZ28D (C)	DEC larry	bus-3-targ-10-lun-0
73:	/dev/disk/dsk8c	DEC	RZ28D (C)	DEC curly	bus-3-targ-10-lun-0
74:	/dev/disk/dsk9c	DEC	RZ28D (C)	DEC moe	bus-3-targ-11-lun-0
74:	/dev/disk/dsk9c	DEC	RZ28D (C)	DEC larry	bus-3-targ-11-lun-0
74:	/dev/disk/dsk9c	DEC	RZ28D (C)	DEC curly	bus-3-targ-11-lun-0

```

75: /dev/disk/dsk10c DEC RZ28D (C) DEC moe bus-3-targ-12-lun-0
75: /dev/disk/dsk10c DEC RZ28D (C) DEC larry bus-3-targ-12-lun-0
75: /dev/disk/dsk10c DEC RZ28D (C) DEC curly bus-3-targ-12-lun-0
76: /dev/disk/dsk11c DEC RZ28D (C) DEC moe bus-3-targ-13-lun-0
76: /dev/disk/dsk11c DEC RZ28D (C) DEC larry bus-3-targ-13-lun-0
76: /dev/disk/dsk11c DEC RZ28D (C) DEC curly bus-3-targ-13-lun-0
77: /dev/disk/dsk12c DEC HSG80 moe bus-4-targ-0-lun-1
77: /dev/disk/dsk12c DEC HSG80 larry bus-4-targ-0-lun-1
77: /dev/disk/dsk12c DEC HSG80 curly bus-4-targ-0-lun-1
78: /dev/disk/dsk13c DEC HSG80 moe bus-4-targ-0-lun-2
78: /dev/disk/dsk13c DEC HSG80 larry bus-4-targ-0-lun-2
78: /dev/disk/dsk13c DEC HSG80 curly bus-4-targ-0-lun-2
79: /dev/disk/dsk14c DEC HSG80 moe bus-4-targ-0-lun-3
79: /dev/disk/dsk14c DEC HSG80 larry bus-4-targ-0-lun-3
79: /dev/disk/dsk14c DEC HSG80 curly bus-4-targ-0-lun-3
:
:
85: /dev/disk/dsk20c DEC HSG80 moe bus-4-targ-0-lun-9
85: /dev/disk/dsk20c DEC HSG80 larry bus-4-targ-0-lun-9
85: /dev/disk/dsk20c DEC HSG80 curly bus-4-targ-0-lun-9
86: /dev/disk/dsk21c DEC HSG80 moe bus-4-targ-0-lun-10
86: /dev/disk/dsk21c DEC HSG80 larry bus-4-targ-0-lun-10
86: /dev/disk/dsk21c DEC HSG80 curly bus-4-targ-0-lun-10
87: /dev/disk/dsk22c DEC HSG80 moe bus-4-targ-0-lun-11
87: /dev/disk/dsk22c DEC HSG80 larry bus-4-targ-0-lun-11
87: /dev/disk/dsk22c DEC HSG80 curly bus-4-targ-0-lun-11
88: /dev/disk/dsk23c DEC HSG80 moe bus-4-targ-0-lun-12
88: /dev/disk/dsk23c DEC HSG80 larry bus-4-targ-0-lun-12
88: /dev/disk/dsk23c DEC HSG80 curly bus-4-targ-0-lun-12
89: /dev/disk/dsk24c DEC HSG80 moe bus-4-targ-0-lun-13
89: /dev/disk/dsk24c DEC HSG80 larry bus-4-targ-0-lun-13
89: /dev/disk/dsk24c DEC HSG80 curly bus-4-targ-0-lun-13
90: /dev/disk/dsk25c DEC HSG80 moe bus-4-targ-0-lun-14
90: /dev/disk/dsk25c DEC HSG80 larry bus-4-targ-0-lun-14
90: /dev/disk/dsk25c DEC HSG80 curly bus-4-targ-0-lun-14
:
:

```

In this example:

- There are three cluster members: moe, larry, and curly.
 - Disks dsk3 through dsk25 are shared by all members; each disk appears three times in the output, with a different host name in each line.
 - Of these, disks dsk8 through dsk14 and dsk20 through dsk25 are under LSM control as members of rootdg (as shown in the output from the `voldisk list` command in step 2). All are candidate disks to use for the migration.
6. Migrate the domains, specifying options such as the number of mirrors and the number of stripe columns.

Note

When you create a mirrored volume with the `volmigrate` command, LSM automatically adds a DRL (except for `cluster_rootvol`). However, you cannot specify a separate

disk for the DRL. If there is enough space, LSM puts the DRL on one of the disks you specified. (If there is not enough space for the mirrors and the DRL, the command fails with a message about insufficient space.)

Even if you specify more disks than needed to create the mirrors, LSM does not put the DRL on the extra disk.

To improve this configuration, after you migrate the domain, you can immediately delete the volume's log and then add a new one, as shown in the following procedure.

-
- To migrate a domain to an LSM volume with the default properties (concatenated, no mirror):

```
# volmigrate domain disk...
```

For example:

```
# volmigrate cluster_root dsk10
```

You can optionally mirror the volume to the original disk later.

- To migrate a domain to an LSM volume of a specific size (for example, smaller than the domain by 2 MB):

```
# volmigrate -l sectors domain disk...
```

For example, to migrate the `cluster_var` domain, which is 17581584 sectors (a little over 8 GB), to a volume that is 2 MB smaller (17577488 sectors):

```
# volmigrate -l 17577488 cluster_var dsk11
```

- To migrate a domain to a volume striped over four disks:

```
# volmigrate -s 4 cluster_root disk disk disk disk
```

For example:

```
# volmigrate -s 4 cluster_root dsk10 dsk11 dsk12 dsk13
```

- To migrate a domain to a mirrored volume on two disks:

```
# volmigrate -m 2 domain disk disk
```

For example:

```
# volmigrate -m 2 cluster_usr dsk12 dsk13
```

- To migrate a domain to a striped, mirrored volume on six disks (each mirror will be striped over three disks):

```
# volmigrate -m 2 -s 3 cluster_usr \  
disk disk disk disk disk disk
```

For example:

```
# volmigrate -m 2 -s 3 cluster_usr \  
dsk10 dsk11 dsk12 dsk13 dsk14 dsk15
```

- If you migrated the domain to a mirrored volume and want to reconfigure the volume so that the DRL plex is on a disk not used by one of the data plexes:

- a. Display the volume properties to identify the DRL plex:

```
# volprint volume
```

For example:

```
# volprint cluster_usrvol
```

```
Disk group: rootdg
```

TY	NAME	ASSOC	KSTATE	LENGTH	...
v	cluster_usrvol	fsgen	ENABLED	1787904	...
pl	cluster_usrvol-01	cluster_usrvol	ENABLED	1787904	...
sd	dsk22-01	cluster_usrvol-01	ENABLED	1787904	...
pl	cluster_usrvol-02	cluster_usrvol	ENABLED	1787904	...
sd	dsk23-01	cluster_usrvol-02	ENABLED	1787904	...
pl	cluster_usrvol-03	cluster_usrvol	ENABLED	LOGONLY	...
sd	dsk22-02	cluster_usrvol-03	ENABLED	65	...

- b. Dissociate the DRL plex:

```
# volplex -o rm dis cluster_usrvol-03
```

- c. Add a new DRL plex, specifying a disk not already used in the same volume (in this case, not dsk22 or dsk23):

```
# volassist addlog cluster_usrvol disk
```

3.4.2.1.3 Mirroring Migrated Domain Volumes to the Original Disk (Optional)

The attributes you specify for a volume with the `volmigrate` command (such as the length and whether the volume is striped) are applied to the mirror when you mirror that volume.

To mirror a volume that is striped over several disks, you must specify the same number of additional disks for the mirror. For example, if the volume is striped over four disks, you need four additional disks to create the mirror, one of which can be the original disk.

To mirror a volume to the original disk:

1. Confirm that all partitions on the original disk are unused (in this example, `dsk6`):

```
# disklabel -r dsk6
```

2. Add the disk to LSM:

```
# voldisksetup -i dsk6
```

3. Add the disk to the `rootdg` disk group. You do not need to specify a disk group because the `rootdg` disk group is the default:

```
# voldg adddisk dsk6
```

4. Verify that there is enough space in the public region of the LSM disk to mirror the volume:

```
# voldisk list dsk6 | grep public
public:      slice=6 offset=16 len=8375968
```

5. Display the cluster volume names to ensure you enter the correct names in the next step:

```
# volprint -vt | grep cluster

v cluster_rootvol fsgen      ENABLED ACTIVE  557936 SELECT  -
v cluster_usrvol  fsgen      ENABLED ACTIVE  1684224 SELECT -
v cluster_varvol  fsgen      ENABLED ACTIVE  1667024 SELECT -
```

6. Mirror the volumes. You must mirror each volume separately:

```
# volassist mirror volume disk
```

- To mirror all the volumes to the same LSM disk:

```
# volassist mirror cluster_rootvol dsk6
# volassist mirror cluster_usrvol dsk6
# volassist mirror cluster_varvol dsk6
```

- To mirror each volume to its own LSM disk:

```
# volassist mirror cluster_rootvol dsk4
# volassist mirror cluster_usrvol dsk8
# volassist mirror cluster_varvol dsk11
```

- If the volume is striped over several disks, specify the same number of disks for the mirror plex.

For example, to mirror a volume that is striped over four disks:

```
# volassist mirror cluster_rootvol dsk6 dsk11 dsk12 dsk13
```

7. For all volumes except `cluster_rootvol`, add a DRL log plex to the volume:

```
# volassist addlog volume
```

3.4.3 Encapsulating Cluster Members' Swap Devices

You can use LSM for the swap devices of cluster members whether or not the clusterwide root, `/usr`, and `/var` file system domains also use LSM volumes.

Note

A cluster member must be running for you to encapsulate its swap devices.

In one command, you can encapsulate:

- All the swap devices for one member at once
- All the swap devices for one member plus swap devices for other cluster members
- Only the swap devices you specify for one or more members

You can set up the encapsulation for several members at once with the `volencap` command, but you must complete the encapsulation procedure on each member in turn with the `volreconfig` command.

- To encapsulate all the swap devices for only the current member:

```
# volencap swap
# volreconfig
```

The `swap` operand is a member-specific shortcut for specifying all the swap devices in the cluster member's private `/etc/sysconfigtab` file.

After the cluster member reboots, its swap devices use separate LSM volumes.

- To encapsulate the swap devices for multiple cluster members:

1. Identify the cluster members:

```
# clu_get_info
```

2. Display the swap devices for each cluster member using one of the following methods:

– On each member, enter:

```
# swapon -s
```

– On one member, display the swap devices listed in each cluster member's `sysconfigtab` file:

```
# grep swap \
/cluster/members/membn/boot_partition/etc/sysconfigtab
```

3. Do one of the following:

– To encapsulate all the swap devices for the current cluster member plus any number of other cluster members' swap devices:

```
# volencap swap dsknp dsknp...
```

- To encapsulate only specific swap devices for the current cluster member plus any number of other cluster members' swap devices:

```
# volencap dsknp dsknp...
```

4. Complete the encapsulation on each cluster member separately:

```
# volreconfig
```

For more information on encapsulating swap devices in a cluster, see `volencap(8)`. For information on creating additional LSM swap volumes, see Section 4.3.7.

3.5 Enabling the Automatic Data Relocation (Hot-Sparing) Feature

To provide higher availability for LSM volumes, you can enable the hot-sparing feature and configure one or more hot-spare disks per disk group to enable LSM to automatically relocate data from a failing disk to a hot-spare disk.

Hot-sparing automatically attempts to relocate redundant data (in mirrored or RAID 5 volumes), and performs parity resynchronization for RAID 5 volumes. For a failing RAID 5 log plex, relocation occurs only if the log plex is mirrored. If so, the hot-spare feature also resynchronizes the RAID 5 log plexes.

A hot-sparing operation:

- Sends mail to the root user before and after the operation. See Section 3.5.2 for examples of mail messages.
- Relocates the LSM objects from the failed disk to a hot-spare disk or to free disk space in the disk group, except if redundancy cannot be preserved. For example, LSM will not relocate data to a disk that contains a mirror of the data.
- Initiates parity resynchronization for an affected RAID 5 plex.
- Updates the configuration database with the relocation information.
- Ensures that the failed disk space is not recycled as free disk space.

The hot-sparing feature is part of the `volwatch` daemon. The `volwatch` daemon has two modes:

- Mail-only mode, which is the default. This setting notifies you of a problem but does not perform hot-sparing. You can reset the daemon to this mode with the `-m` option. Without hot-sparing enabled, you must investigate and resolve problems manually. For more information, see Section 6.4.

- Mail-and-spare mode, which you set with the `-s` option.

You can specify mail addresses with either option. Alternatively, use the `rcmgr` command to set the `VOLWATCH_USERS` variable in the `/etc/rc.config.common` file. For more information, see `rcmgr(8)`.

To enable the hot-sparing feature:

```
# volwatch -s [mail-address...]
```

To return the `volwatch` daemon to mail-only mode:

```
# volwatch -m [mail-address...]
```

Note

Only one `volwatch` daemon can run on a system or cluster member at any time. The daemon's setting applies to the entire LSM configuration; you cannot specify some disk groups to use hot-sparing but not others.

In a cluster, entering the `volwatch -s` command on any member enables hot-sparing across the entire cluster, and it remains in effect until you disable it by entering `volwatch -m`. The setting is persistent across cluster or member reboots.

3.5.1 Configuring and Deconfiguring Hot-Spare Disks

Configure at least one hot-spare disk in each disk group, but ideally, configure one hot-spare disk for every disk used in a mirrored or RAID 5 (redundant) volume in the disk group. Each hot-spare disk should be large enough to replace any disk in a redundant volume in the disk group, because there is no way to assign specific hot-spare disks to specific volumes. After a hot-spare operation occurs, you can add more disks to the disk group and configure them as replacement hot-spare disks.

You can configure `sliced` or `simple` disks as hot-spare disks, but `sliced` disks are preferred. If you use `simple` disks, try to use only those that you initialized on the `c` partition of the disk (the entire disk). A `simple` disk that encompasses the entire disk has only one private region. This makes the maximum amount of disk space available to LSM, providing the most flexibility in relocating redundant data off a failing disk in a hot-spare operation.

LSM does not use a hot-spare disk for normal data storage (creating new volumes) unless you specify otherwise.

- To configure a hot-spare disk:

```
# voledit [-g disk_group] set spare=on disk
```

For example, to configure `dsk5` as a hot-spare disk in the `rootdg` disk group:

```
# voledit set spare=on dsk5
```

- To deconfigure a hot-spare disk:

```
# voledit [-g disk_group] set spare=off disk
```

For example, to deconfigure `dsk5` as a hot-spare disk in the `rootdg` disk group:

```
# voledit set spare=off dsk5
```

To display the hot-spare disks in a disk group, use the `voldisk list` command. In the `STATUS` column of the output, all available hot-spare disks show a status of `online spare`.

3.5.2 Examples of Mail Notification for Exception Events

During a hot-sparing operation, LSM sends mail to the `root` account (and other specified accounts) to notify the recipients about the failure and identify the affected LSM objects. Afterward, LSM sends another mail message if any action is taken.

There is a 15-second delay before the event is analyzed and the message is sent. This delay allows a group of related events to be collected and reported in a single mail message.

- Example 3–1 shows a sample mail notification sent when LSM detects an exception event.
- Example 3–2 shows the mail message sent if a disk completely fails.
- Example 3–3 shows the mail message sent if a disk partially fails.
- Example 3–4 shows the mail message sent if data relocation is successful.
- Example 3–5 shows the mail message sent if relocation cannot occur, because there is no hot-spare disk or free disk space.
- Example 3–6 shows the mail message sent if data relocation fails.
- Example 3–7 shows the mail message sent if volumes using mirror plexes are made unusable due to disk failure.
- Example 3–8 shows the mail message sent if volumes using RAID 5 plexes are made unusable due to disk failure.

Example 3–1: Sample Mail Notification

Failures have been detected by the Logical Storage Manager:

```
failed disks:
disk    [1]
:
:

failed plexes:
plex    [2]
:
:

failed log plexes:
plex    [3]
:
:

failing disks:
disk    [4]
:
:

failed subdisks:
subdisk [5]
:
:
```

The Logical Storage Manager will attempt to find spare disks, relocate failed subdisks and then recover the data in the failed plexes.

The following describes the sections of the mail notification:

- [1] The *disk* under *failed disks* specifies a disk that appears to have failed completely.
- [2] The *plex* under *failed plexes* shows a plex that is detached due to I/O failures to subdisks the plex contains.
- [3] The *plex* under *failed log plexes* indicates a RAID 5 or dirty region log (DRL) plex that has experienced failures.
- [4] The *disk* under *failing disks* indicates a partial disk failure or a disk that is in the process of failing. When a disk has failed completely, the same *disk* appears under both *failed disks* and *failing disks*.
- [5] The *subdisk* under *failed subdisks* indicates a subdisk in a RAID 5 volume that has been detached due to I/O errors.

Example 3–2 shows that a disk named *disk02* was failing, then detached by a failure and that plexes named *home-02*, *src-02*, and *mkting-01* were also detached (probably due to the disk failure).

Example 3–2: Complete Disk Failure Mail Notification

```
To: root
Subject: Logical Storage Manager failures on hostname

Failures have been detected by the Logical Storage Manager

failed disks:
  disk02

failed plexes:
  home-02
  src-02
  mktng-01

failing disks:
  disk02
```

Example 3–3: Partial Disk Failure Mail Notification

```
To: root
Subject: Logical Storage Manager failures on hostname

Failures have been detected by the Logical Storage Manager:

failed disks:
  disk02

failed plexes:
  home-02
  src-02
```

Example 3–4: Successful Data Relocation Mail Notification

```
Volume home Subdisk home-02-02 relocated to dsk12-02,
but not yet recovered.
```

If the data recovery is successful, the following message is sent:

```
Recovery complete for volume home in disk group dg03.
```

If the data recovery is unsuccessful, the following message is sent:

```
Failure recovering home in disk group dg03.
```

Example 3–5: No Hot-Spare or Free Disk Space Mail Notification

Relocation was not successful for subdisks on disk dsk3
in volume vol_02 in disk group rootdg.
No replacement was made and the disk is still unusable.

The following volumes have storage on dsk3:

```
vol_02
:
:
```

These volumes are still usable, but the redundancy of those
volumes is reduced. Any RAID5 volumes with storage on the
failed disk may become unusable in the face of further failures.

Example 3–6: Data Relocation Failure Mail Notification

Relocation was not successful for subdisks on disk dsk14 in
volume data02 in disk group data_dg.
No replacement was made and the disk is still unusable.

```
:
```

The actual mail notification includes an error message indicating why the
data relocation failed.

Example 3–7: Unusable Volume Mail Notification

The following volumes:

```
finance
:
:
```

have data on dsk23 but have no other usable mirrors
on other disks. These volumes are now unusable and the data on
them is unavailable. These volumes must have their data restored.

Example 3–8: Unusable RAID 5 Volume Mail Notification

The following RAID5 volumes:

```
vol_query
:
:
```

have storage on dsk43 and have experienced
other failures. These RAID5 volumes are now unusable
and data on them is unavailable. These RAID5 volumes must
have their data restored.

3.6 LSM Files, Directories, Device Drivers, and Daemons

After you install and initialize LSM, several new files, directories, device drivers, and daemons are present on the system. These are described in following sections.

3.6.1 LSM Files

The `/dev` directory contains the device special files (Table 3–3) that LSM uses to communicate with the kernel.

Table 3–3: LSM Device Special Files

Device Special File	Function
<code>/dev/volconfig</code>	Allows the <code>vold</code> daemon to make configuration requests to the kernel
<code>/dev/volevent</code>	Used by the <code>voliotrace</code> command to view and collect events
<code>/dev/volinfo</code>	Used by the <code>volprint</code> command to collect LSM object status information
<code>/dev/voliiod</code>	Provides an interface between the volume extended I/O daemon (<code>voliiod</code>) and the kernel

3.6.2 LSM Directories

The `/etc/vol` directory contains the `volboot` file and the subdirectories (Table 3–4) for LSM use.

Table 3–4: LSM /etc/vol Subdirectories

Directory	Function
<code>reconfig.d</code>	Provides temporary storage during encapsulation of existing file systems. Instructions for the encapsulation process are created here and used during the reconfiguration.
<code>tempdb</code>	Used by the volume configuration daemon (<code>vold</code>) while creating the configuration database during startup and while updating configuration information.
<code>vold_diag</code>	Creates a socket portal for diagnostic commands to communicate with the <code>vold</code> daemon.
<code>vold_request</code>	Provides a socket portal for LSM commands to communicate with the <code>vold</code> daemon.

The `/dev` directory contains the subdirectories (Table 3–5) for volume block and character devices.

Table 3–5: LSM Block and Character Device Subdirectories

Directory	Contains
/dev/vol	Block device interfaces for LSM volumes
/dev/rvol	Character device interfaces for LSM volumes

3.6.3 LSM Device Drivers

There are two LSM device drivers:

- `volspec` — The volume special device driver communicates with the LSM device special files. This is not a loadable driver; it must be present at boot time.
- `voldev` — The volume device driver communicates with LSM volumes and provides an interface between LSM and the physical disks.

3.6.4 LSM Daemons

There are two LSM daemons:

- `vold` — The Volume Configuration Daemon maintains configurations of disks and disk groups. It also:
 - Takes requests from other utilities for configuration changes
 - Communicates change requests to the kernel
 - Modifies configuration information stored on disk
 - Initializes LSM when the system starts
- `voliiod` — The Volume Extended I/O Daemon performs the functions of a utility and a daemon.

As a utility, `voliiod`:

- Returns the number of running volume I/O daemons
- Starts more daemons when necessary
- Removes some daemons from service when they are no longer needed

As a daemon, `voliiod`:

- Schedules I/O requests that must be retried
- Schedules writes that require logging

4

Creating LSM Disks, Disk Groups, and Volumes

This chapter describes how to:

- Create LSM disks (Section 4.1)
- Create a disk group and display free space in a disk group (Section 4.2)
- Create an LSM volume for new data (Section 4.3)
- Create LSM volumes with special properties, such as a mirrored volume with each plex on a different bus (Section 4.4)
- Configure UFS or AdvFS file systems to use an LSM volume (Section 4.5)
- Place existing data in an LSM volume (Section 4.6)

Use the information from the worksheets you filled out in Chapter 2 to create disk groups and LSM volumes.

4.1 Overview of Creating LSM Disks

You create an LSM disk when you initialize or encapsulate a disk or disk partition for LSM use. Specifying a disk name, such as `dsk10`, initializes the entire disk as an LSM `sliced` disk. Specifying a partition name, such as `dsk10g` or `dsk10c`, initializes the partition as an LSM `simple` disk. Encapsulating a disk or disk partition that contains data you want placed under LSM control creates a `nopriv` disk.

Initializing an LSM disk:

- Formats the disk or partition as an LSM disk
- Assigns a disk media name to the LSM disk
- Writes a new disk label
- Overwrites existing data on the disk

Note

If the disk is new to the system and LSM is already running, enter the `hwmgr scan scsi` command, then the `voldctl enable` command, to make LSM recognize the disk.

To create LSM `sliced` or `simple` disks, you can use either the `voldisksetup` command or the `voldiskadd` script. (To encapsulate a disk or partition, see Section 4.6.1.)

- The `voldisksetup` command initializes LSM disks with the default attributes or the attributes you specify for the disk, such as the number of configuration database copies (Section 4.1.1) or a specific offset for the start of the public or private region (Section 4.1.2).

See Section 4.1.3 to create LSM `sliced` disks or Section 4.1.4 to create LSM `simple` disks.

- The `voldiskadd` script lets you interactively create LSM disks (with default attributes only), add them to disk groups, or create new disk groups. (See Section 4.2.1.)

4.1.1 Overview of Configuration Database Copies

By default, LSM configures each `sliced` or `simple` disk with the potential to have one copy of the configuration database for its disk group. For disk groups with fewer than four disks, configure each disk to have two copies of the configuration database to ensure multiple copies in case one or more disks fail. Use the `voldisksetup` command to specify the number of configuration database copies on an LSM disk.

An LSM `sliced` or `simple` disk can have 0, 1, or 2 copies of the configuration database for its disk group. LSM enables the specified number of configuration database copies on each disk only when you add the disk to a disk group and only if necessary to maintain the proper number and distribution of copies for the disk group as a whole. For most system configurations, initialize your LSM disks with the default number of copies and allow LSM to manage them.

To maintain the proper number and distribution of LSM configuration database copies in Fibre Channel environments, see the Best Practice entitled *Ensuring Redundancy of LSM Configuration Databases on a Fibre Channel* at the following URL:

http://www.tru64unix.compaq.com/docs/best_practices

4.1.2 Overview of Disk Offsets

When you initialize a disk for LSM use, by default LSM skips over the first 16 blocks on the disk to preserve the disk header and bootstrap information. The public region of a `sliced` disk and the private region of a `simple` disk start at the first block after this default offset. Using the `voldisksetup` command, you can specify a different offset, if necessary.

For example, you can set up your LSM disks to align I/O requests to the chunk size of an underlying RAID hardware device. For more information on this specific application, see the Best Practice entitled *Aligning LSM Disks and Volumes to Hardware RAID Devices* at the following URL:

http://www.tru64unix.compaq.com/docs/best_practices

4.1.3 Creating LSM Sliced Disks

The following examples show how to create LSM `sliced` disks with various attributes:

- Default offset (16 blocks) with default number of configuration database copies (1):

```
# voldisksetup -i dsk5
```

- Default offset with a different number of configuration database copies (can be either 0 or 2):

```
# voldisksetup -i dsk5 nconfig=2
```

- Specified offset and default number of configuration database copies (1):

```
# voldisksetup -i dsk5 puboffset=128
```

- Specified offset with a different number of configuration database copies (can be either 0 or 2):

```
# voldisksetup -i dsk5 puboffset=128 nconfig=2
```

Optionally (but recommended), make a backup copy of the disk label information for all your LSM disks (Section 4.1.5).

After you initialize a disk or disk partition as an LSM disk, you can add it to a disk group. For information on creating a disk group, see Section 4.2. For information on adding an LSM disk to an existing disk group, see Section 5.2.2.

4.1.4 Creating LSM Simple Disks

The default private region offset for `simple` disks is 16 blocks on the `a` or `c` partitions and 0 blocks on all other partitions. The default number of configuration database copies is 1.

The following examples show how to create LSM `simple` disks with various attributes:

- Default offset with default number of configuration database copies (1):

```
# voldisksetup -i dsk7g
```

- Default offset with a different number configuration database copies (either 0 or 2):

```
# voldisksetup -i dsk7g nconfig=2
```

- Specified offset with default number of configuration database copies (1):

```
# voldisksetup -i dsk7c privoffset=128
```

- Specified offset with a different number configuration database copies (either 0 or 2):

```
# voldisksetup -i dsk7c privoffset=128 nconfig=2
```

Optionally (but recommended), make a backup copy of the disk label information for all your LSM disks (Section 4.1.5).

After you initialize an LSM disk, you can add it to a disk group. For information on creating a disk group, see Section 4.2. For information on adding an LSM disk to an existing disk group, see Section 5.2.2.

4.1.5 Backing Up Disk Label Information

Back up the updated disk label information for each LSM disk. Having this information will simplify the process of replacing a failed disk, by allowing you to copy the failed disk's attributes to a new disk. After a disk fails, you cannot read its disk label and therefore cannot copy that information to a new disk.

You can back up the disk label information before or after adding a disk to a disk group; the information does not change.

To back up the disk label information:

```
# disklabel dskn > file
```

For more information, see `disklabel(8)`.

4.2 Creating Disk Groups

The default `rootdg` disk group is created when you initialize LSM and always exists on a system running LSM. You can create additional disk groups to organize your disks into logical sets. Each disk group that you create must have a unique name and must contain at least one `sliced` or `simple` LSM disk to store the disk group's configuration database. An LSM disk can belong to only one disk group.¹

For large LSM configurations, consider keeping `rootdg` fairly small (ten disks or fewer) and create other disk groups for the rest of your LSM configuration. If possible, use `rootdg` only for volumes relating to the

¹ It is possible to create multiple LSM disks from different partitions of the same physical device and add them to different disk groups. However, because of the complications this can cause in volume recovery and disk group management, we strongly discourage this practice.

system disk (on a standalone system), the clusterwide root, /usr, and /var file system domains, and members' swap devices.

Having separate disk groups gives you the ability to move LSM volumes to different systems or clusters.

You can create an LSM disk group using the following commands:

- The `voldiskadd` interactive script (Section 4.2.1)
- The `voldg` command (Section 4.2.2)

4.2.1 Creating LSM Disks and Disk Groups Using the `voldiskadd` Script

The `voldiskadd` script lets you do all of the following tasks in one interactive session:

- Initialize disks or disk partitions (with default values only) for exclusive use by LSM
- Create a disk group
- Add disks to an existing disk group

Note

If a disk group will have fewer than four disks, see Section 4.1.1.

You can invoke the `voldiskadd` script with or without a disk name. If you invoke the script by itself, it prompts you for the following information:

- A disk or disk partition
If you specify an entire disk, LSM initializes it as an LSM `sliced` disk. If you specify a disk partition, LSM initializes the partition as an LSM `simple` disk. You can specify several disks and disk partitions at once, separated by a space; for example:

```
# voldiskadd dsk3 dsk4a dsk5 dsk6g
```
- A disk group name
If you are creating a disk group, the disk group name must be unique and can contain up to 31 alphanumeric characters that cannot include spaces or the slash (/).
- A disk media name for each disk you configure in the disk group
You can use the default disk media name (which will be the same as the disk access name) or you can assign a disk media name of up to 31 alphanumeric characters that cannot include spaces or the slash (/).
- Whether to configure the disk as a hot-spare disk for the disk group

For more information about hot-spare disks, see Section 3.5. For the best protection, configure at least one hot-spare disk in each disk group that contains redundant volumes.

Example 4–1 uses a disk named `dsk9` to create a disk group named `dg1`:

Example 4–1: Creating an LSM Disk Group with the `voldiskadd` script

```
# voldiskadd dsk9
Add or initialize disks

Menu: VolumeManager/Disk/AddDisks

Here is the disk selected.

dsk9

Continue operation? [y,n,q,?] (default: y) Return

You can choose to add this disk to an existing disk group, a
new disk group, or leave the disk available for use by future
add or replacement operations. To create a new disk group,
select a disk group name that does not yet exist. To leave
the disk available for future use, specify a disk group name
of "none".

Which disk group [<group>,none,list,q,?] (default: rootdg) dg1

There is no active disk group named dg1.

Create a new group named dg1? [y,n,q,?] (default: y) Return

The default disk name that will be assigned is:

dg101

Use this default disk name for the disk? [y,n,q,?] (default: y) Return

Add disk as a spare disk for dg1? [y,n,q,?] (default: n) Return

A new disk group will be created named dg1 and the selected disks
will be added to the disk group with default disk names.
dsk9

Continue with operation? [y,n,q,?] (default: y) Return

The following disk device has a valid disk label, but does
not appear to have been initialized for the Logical Storage
Manager. If there is data on the disk that should NOT be
destroyed you should encapsulate the existing disk partitions
as volumes instead of adding the disk as a new disk.

dsk9

Initialize this device? [y,n,q,?] (default: y) Return

Initializing device dsk9.

Creating a new disk group named dg1 containing the disk
device dsk9 with the name dg101.

Goodbye.
```

4.2.2 Creating Disk Groups Using the `voldg` Command

Use the `voldg` command to create disk groups from disks that are initialized for LSM, including disks configured to have a nondefault number of configuration database copies (Section 4.1).

By default, LSM maintains a minimum of four active copies of the configuration database in each disk group. You can specify a different number of active copies even if you are using disks initialized with the defaults. You can specify a maximum equal to the number of sliced and simple disks in the disk group.

For example, if you create a disk group with ten sliced or simple disks, each of which is configured by default to store one copy, you can set the number of copies for that disk group to ten. If you configured each disk to store two copies, you can set the number of copies for the disk group to 20.

For any disk group, the maximum number of active configuration database copies is derived from the number of sliced or simple disks in the group and the number of copies each of those disks is configured to store.

Note

You can set the number of configuration database copies that a disk group will maintain only when you create the disk group. You cannot change the number of active copies for existing disk groups.

Maintaining more than the default number of copies can affect performance, because every change to the LSM configuration is written to all active copies of the database. If you want to use a nondefault number, choose a number of copies sufficient to meet your environment's needs but small enough to minimize the performance impact.

To create a disk group with default values using the `voldg` command:

```
# voldg init disk_group disk [disk...]
```

For example:

```
# voldg init dg1 dsk5 dsk6 dsk7 dsk9 dsk10 dsk11 dsk12
```

If a disk group will have fewer than four disks, configure each disk to have two copies of the disk group's configuration database (Section 4.1) to ensure that the disk group has multiple copies in case one or more disks fail.

To create a disk group and set the number of configuration copies to 10:

```
# voldg init newdg disks nconfig=10
```

For example:

```
# voldg init newdg dsk100 dsk101 dsk102... dsk110 nconfig=10
```

4.3 Creating LSM Volumes for New Data

To create an LSM volume for a new file system or application, use the `volassist` command. The `volassist` command either finds the necessary space within the disk group and creates all the objects for the volume or uses attributes you supply on the command line, such as specific disk names. You must assign a name and length (size) on the command line.

You can specify values for other LSM volume attributes on the command line or in a text file that you create. If you do not specify a value for an attribute, LSM uses a default value.

By default, LSM uses a stripe width of 64K bytes for striped plexes and 16K bytes for RAID 5 plexes. However, you can use a different stripe width if you have applications that use an I/O transfer size that requires a different value, or if you have created hardware devices with a particular stripe width and you want the LSM volume created from those devices to align its writes (data stripes) to the hardware's stripe width.

For more information, see the Best Practice entitled *Aligning LSM Disks and Volumes to Hardware RAID Devices*, available at the following URL:

http://www.tru64unix.compaq.com/docs/best_practices/sys_bps.html

- To learn more about the default volume attributes file and creating your own volume attributes file, see Section 4.3.1.
- To create an LSM volume with a single concatenated plex, see Section 4.3.2.
- To create an LSM volume with mirrored concatenated plexes, see Section 4.3.3.
- To create an LSM volume with a single striped plex, see Section 4.3.4.
- To create an LSM volume with mirrored striped plexes, see Section 4.3.5.
- To create an LSM volume with a RAID 5 plex, see Section 4.3.6.

4.3.1 Overview of LSM Volume Attributes

The following lists the priority given to assignable attributes:

1. Values on the command line
2. Values in a file that you specify by using the `volassist -d` option
3. Values in the `/etc/default/volassist` file

4. Default values

To display the default values for volume attributes:

```
# volassist help showattrs

#Attributes:
  layout=nomirror,nostripe,span,nocontig,raid5log,noregionlog,nofpalog,diskalign,nostorage
  mirrors=2 columns=0 nlogs=1 regionlogs=1 raid5logs=1 fpalogs=1
  min_columns=2 max_columns=8
  regionloglen=0 raid5loglen=0 logtype=region
  stripe_stripeunitsize=128 raid5_stripeunitsize=32
  usetype=fsgen diskgroup= comment="" fstype=
  user=0 group=0 mode=0600
  probe_granularity=2048
  alloc=
  wantalloc=
  mirror=
```

Some volume attributes have several options to define them. Some options define an attribute globally, where others define an attribute for a specific plex type. For example, you can specify the size of a stripe data unit using the `stripeunit` (or `stwidth`) option for both striped or RAID 5 plexes, the `stripe_stripeunit` (or `stripe_stwid`) option specifically for striped plexes, or the `raid5_stripeunit` (or `raid5_stwid`) option specifically for RAID 5 plexes.

For a complete list of attributes, see `volassist(8)`. Table 4–1 describes some of the common attributes for which you can specify a value.

Table 4–1: Common LSM Volume Attributes

Attribute Description	Attribute Options
Plex type	<code>layout={concatenated striped raid5}</code>
Usage type	<code>-U {fsgen gen raid5}</code>
Number of plexes (mirrors). Default is 2.	<code>mirror={number yes no}</code>
Type of log: <ul style="list-style-type: none">• <code>dr1</code> for mirrored volumes (only)• <code>region</code> for mirrored or RAID 5 volumes	<code>logtype={dr1 region none}</code>
Number of FPA logs (for mirrored volumes only)	<code>nfpalog=number</code>
Size of stripe width for a striped or RAID 5 plex, in blocks, sectors, kilobytes, megabytes, or gigabytes	<code>stripeunit=data_unit_size</code> or <code>stwid=data_unit_size</code>
Number of columns for a striped or RAID 5 plex; typically, the number of disks in each plex	<code>nstripe=number_of_columns</code> or <code>ncolumn=number_of_columns</code>

Creating a text file that specifies many of these attributes is useful if you create many LSM volumes that use the same nondefault values for attributes. Any attribute that you can specify on the command line can be specified on a separate line in the text file. By default, LSM looks for the `/etc/default/volassist` file when you create an LSM volume. If you created an `/etc/default/volassist` file, LSM creates each volume using the attributes that you specify on the command line and in the `/etc/default/volassist` file.

Example 4–2 shows an LSM volume attributes file called `/etc/default/volassist` that creates an LSM volume using a four-column striped plex with two mirrors, a stripe width of 32K bytes, and no log.

Example 4–2: LSM Volume Attribute Defaults File

```
# LSM Vn.n
# volassist defaults file. Use '#' for comments
# number of stripes
nstripe=4
# layout
layout=striped
# mirroring
nmirror=2
# logging
logtype=none
# stripe size
stripeunit=32k
```

For example, to create an LSM volume using the attributes in the `/etc/default/volassist` file:

```
# volassist make volume length
```

If you have created a custom attributes file and want LSM to use the applicable attributes in that file when creating the volume, specify the attributes file as follows:

```
# volassist make volume length -d filename
```

With this option, LSM creates the volume using both the attributes that you specify on the command line (such as name and length) and those in the named file. If you specify an attribute that conflicts with the contents of the file, the command line takes precedence.

To specify a length (size) for the volume, enter a number and the appropriate suffix:

Suffix	Unit
b	Blocks
s	Sectors (default)
k	Kilobytes
m	Megabytes
g	Gigabytes
t	Terabytes

4.3.2 Creating LSM Volumes with a Single Concatenated Plex

A volume with a single concatenated plex is also called a simple volume. It provides no data redundancy; if a disk fails, the data is lost. To avoid this, you can either create the volume initially with mirrored concatenated plexes (Section 4.3.3) or add another data plex to the volume later (Section 5.5.2).

You can either let LSM find space on any available disks in the disk group or specify which disks you want LSM to use in creating the volume.

For volumes that will support a file system, use the default `fsgen` usage type. For volumes that will contain raw data, such as a database, use the `gen` usage type.

- To create an LSM volume with one concatenated plex on any available disks in the disk group:

```
# volassist [-g disk_group] [-U use_type] make volume length
```

For example, to make a 3 GB volume named `data01` with the `gen` usage type in the `dg1` disk group:

```
# volassist -g dg1 -U gen make data01 3g
```

- To create an LSM volume with one concatenated plex on the disk or disks you specify:

```
# volassist [-g disk_group] [-U use_type] make volume \  
length disks
```

For example, to make an 800 MB volume named `acct_files` with the default `fsgen` usage type in the `dg1` disk group using disks `dsk10` and `dsk11`:

```
# volassist -g dg1 make acct_files 800m dsk10 dsk11
```

4.3.3 Creating LSM Volumes with Mirrored, Concatenated Plexes

To provide data redundancy (high availability) you can create an LSM volume with two or more concatenated plexes. To further increase

availability, you can specify disks on a different bus for each data plex and the DRL plex by making the volume in steps, allowing you to control which disks LSM uses to create each plex.

For volumes that will support a file system, use the default `fsgen` usage type. For volumes that will contain raw data, such as a database, use the `gen` usage type. You specify a usage type only when you create the volume, but not when you add a mirror plex or a log plex to an existing volume.

To improve availability of the logs, you can add multiple logs when you create the volume, with the `nlog=count` attribute. You can also add one or more FPA logs to the volume when you create it, with the `nfpalog=count` attribute. An FPA log is not required; it is used only when you create a secondary volume from one plex of a primary volume, (Section 5.4.2.2). Multiple FPA logs ensure that FPA logging remains in effect in case of disk failure.

4.3.3.1 Creating a Mirrored, Concatenated Volume in One Step

- To create an LSM volume with mirrored concatenated plexes on any available disks, optionally with multiple DRL logs or FPA logs (default: one DRL, no FPA logs):

```
# volassist [-g disk_group] [-U use_type] make volume length \  
mirror=2 [nlog=count] [nfpalog=count]
```

For example, to create a 256 MB volume named `mirrVol` with two concatenated plexes and the default `fsgen` usage type on any available disks in the `dg1` disk group:

```
# volassist -g dg1 make mirrVol 256m mirror=2
```

By default, LSM creates a DRL plex for all mirrored volumes.

- To create an LSM volume with mirrored, concatenated plexes on the disks you specify:

```
# volassist [-g disk_group] make volume \  
length mirror=2 disks
```

For example, to create a 256 MB volume named `mirrVol` with two concatenated plexes and the `gen` usage type on disks `dsk21`, `dsk22`, `dsk23`, and `dsk24` in the `dg1` disk group:

```
# volassist -g dg1 -U gen make mirrVol 256m mirror=2 \  
dsk21 dsk22 dsk23 dsk24
```

This creates a mirrored volume with a DRL plex on at least two of the specified disks. If you specify more disks than LSM needs, they are not used.

4.3.3.2 Creating a Mirrored, Concatenated Volume with Plexes on Different Buses

To create a mirrored, concatenated volume with each plex on a different bus:

1. Create the volume with a single concatenated plex, specifying the disks:

```
# volassist [-g disk_group] [-U use_type] make \  
volume length disks
```

For example:

```
# volassist -g dg1 -U gen make vol2 3g dsk2 dsk3 dsk4
```

2. Add another concatenated plex (mirror) to the volume, specifying disks on a different bus:

```
# volassist [-g disk_group] mirror volume disks
```

For example:

```
# volassist -g dg1 mirror vol2 dsk5 dsk6 dsk7
```

When you add a mirror to a volume manually, LSM does not add a DRL plex.

3. Add a DRL plex to the volume, specifying a disk that is not used by one of the data plexes:

```
# volassist -g dg1 addlog volume disk
```

For example:

```
# volassist -g dg1 addlog vol2 dsk8
```

The volume is ready for use.

4.3.4 Creating LSM Volumes with a Single Striped Plex

An LSM volume with a striped plex can offer faster performance than a volume with a concatenated plex. You can specify the number of disks you want LSM to stripe the data over (the number of stripe columns), or you can let LSM stripe the data over as many disks as necessary based on the volume size and the stripe width you specify, if other than the default.

A volume with a single striped plex does not provide data redundancy; if a disk fails, the data is lost. To avoid this, you can either create the volume initially with mirrored striped plexes (Section 4.3.5) or add another data plex to the volume later (Section 5.5.2).

Note

In general, do not use LSM to stripe data if you also use a hardware controller to stripe data. In some specific cases such a configuration can improve performance but only if:

- Most of the volume I/O requests are large (≥ 1 MB).

- The LSM volume is striped over multiple RAID sets on different controllers.
- The LSM stripe size is a multiple of the full hardware RAID stripe size.

The number of LSM columns in each plex in the volume should be equal to the number of hardware RAID controllers. See your hardware RAID documentation for information about how to choose the best number of columns for the hardware RAID set.

This section contains a number of examples of creating volumes with one or more striped plexes. In the examples, the syntax for specifying a different stripe width is shown. To use the default stripe width, omit that option.

By default, the `volassist` command creates columns for a striped plex on disks in alphanumeric order, regardless of their order on the command line. To improve performance, create the columns in each plex using disks on different buses. For more information about specifying the disk order for columns in a striped plex, see Section 4.4.2.

When creating volumes with a striped plex, you must specify the number of stripe columns per plex. Each column must be on a different disk; therefore, this is the number of disks over which to stripe the plex.

For volumes that will support a file system, use the default `fsgen` usage type. For volumes that will contain raw data, such as a database, use the `gen` usage type. You specify a usage type only when you create the volume, but not when you add a mirror plex or a log plex to an existing volume.

The following examples show how to create LSM striped volumes with different properties:

- To create an LSM volume with a single striped plex on any available disks in the disk group:

```
# volassist [-g disk_group] [-U use_type] make volume length \
layout=stripe ncolumn=number_of_columns [stwid=data_unit_size]
```

For example, to create a 128 MB volume named `v_stripe`, with a usage type of `gen` and an 8-column striped plex with a stripe width of 32K bytes, in the `dg2` disk group:

```
# volassist -g dg2 -U gen make v_stripe 128m \
layout=stripe ncolumn=8 stwid=32k
```

The volume looks similar to the following:

```
Disk group: dg2
```

V NAME	USETYPE	KSTATE	STATE	LENGTH	READPOL	PREFPLEX
PL NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WID MODE
SD NAME	PLEX	DISK	DISKOFFS	LENGTH	[COL/]OFF	DEVICE MODE

```

v v_stripe      gen          ENABLED ACTIVE 262144 SELECT v_stripe-01
pl v_stripe-01 v_stripe    ENABLED ACTIVE 262144 STRIPE 8/64 RW
sd dsk12-01    v_stripe-01 dsk12    0      32768 0/0    dsk12    ENA
sd dsk13-01    v_stripe-01 dsk13    0      32768 1/0    dsk13    ENA
sd dsk14-01    v_stripe-01 dsk14    0      32768 2/0    dsk14    ENA
sd dsk15-01    v_stripe-01 dsk15    0      32768 3/0    dsk15    ENA
sd dsk16-01    v_stripe-01 dsk16    0      32768 4/0    dsk16    ENA
sd dsk17-01    v_stripe-01 dsk17    0      32768 5/0    dsk17    ENA
sd dsk18-01    v_stripe-01 dsk18    0      32768 6/0    dsk18    ENA
sd dsk19-01    v_stripe-01 dsk19    0      32768 7/0    dsk19    ENA

```

- To create an LSM volume with a single striped plex on the disk or disks you specify:

```
# volassist [-g disk_group] make volume length layout=stripe \
ncolumn=number_of_columns [stwid=data_unit_size] disks
```

For example, to create a 300 MB LSM volume named `volst` with a 2-column striped plex with the default stripe width (64K bytes) and a usage type of `gen` on disks `dsk20` and `dsk21` in the `dg2` disk group:

```
# volassist -g dg2 -U gen make volst 128m \
layout=stripe ncolumn=2 dsk20 dsk21
```

The volume looks similar to the following:

```

v volst        gen          ENABLED 262144 -        ACTIVE -        -
pl volst-01    volst        ENABLED 262144 -        ACTIVE -        -
sd dsk20-01    volst-01     ENABLED 131072 0        -        -        -
sd dsk21-01    volst-01     ENABLED 131072 0        -        -        -

```

4.3.5 Creating LSM Volumes with Mirrored, Striped Plexes

To provide data redundancy (high availability) and improved performance, you can create an LSM volume with two or more striped plexes.

You can either let LSM find space on any available disks in the disk group or specify which disks you want LSM to use in creating the volume. To further increase performance and availability, you can specify disks on a different bus for each plex and the DRL plex by making the volume in steps, allowing you to control which disks LSM uses to create each plex.

To improve availability of the logs, you can add multiple logs with the `nlog=count` attribute when you create the volume. You can also add one or more FPA logs to the volume with the `nfpalog=count` attribute when you create it. An FPA log is not required; it is used only when you create a secondary volume from one plex of a primary volume (Section 5.4.2.2). Multiple FPA logs ensure that FPA logging remains in effect in case of disk failure.

When creating volumes with a striped plex, you must specify the number of stripe columns per plex. Each column must be on a different disk; therefore, this is the number of disks over which to stripe the plex.

For volumes that will support a file system, use the default `fs`gen usage type. For volumes that will contain raw data, such as a database, use the `gen` usage type. You specify a usage type only when you create the volume, but not when you add a mirror plex or a log plex to an existing volume.

4.3.5.1 Creating a Mirrored, Striped Volume in One Step

- To create an LSM volume with mirrored, striped plexes on any available disks in the disk group:

```
# volassist [-g disk_group] [-U use_type] make volume length \
mirror=2 layout=stripe ncolumn=number_of_columns \
[stwid=data_unit_size]
```

For example, to create a mirrored, striped volume named `mv01` in the `rootdg` disk group with a stripe width of 32K bytes on any available disks in the disk group:

```
# volassist -U gen make mv01 256m \
mirror=2 layout=stripe ncolumn=3 stwid=32k
```

The volume looks similar to the following:

Disk group: `rootdg`

V NAME	USETYPE	KSTATE	STATE	LENGTH	READPOL	PREFPLEX	
PL NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WID	MODE
SD NAME	PLEX	DISK	DISKOFFS	LENGTH	[COL/]OFF	DEVICE	MODE
v mv01	gen	ENABLED	ACTIVE	524288	SELECT	-	
pl mv01-01	mv01	ENABLED	ACTIVE	524352	STRIPE	3/64	RW
sd dsk1-01	mv01-01	dsk1	65	174784	0/0	dsk1	ENA
sd dsk2-01	mv01-01	dsk2	0	174784	1/0	dsk2	ENA
sd dsk3-01	mv01-01	dsk3	0	174784	2/0	dsk3	ENA
pl mv01-02	mv01	ENABLED	ACTIVE	524352	STRIPE	3/64	RW
sd dsk4-01	mv01-02	dsk4	0	174784	0/0	dsk4	ENA
sd dsk5-01	mv01-02	dsk5	0	174784	1/0	dsk5	ENA
sd dsk6-01	mv01-02	dsk6	0	174784	2/0	dsk6	ENA
pl mv01-03	mv01	ENABLED	ACTIVE	LOGONLY	CONCAT	-	RW
sd dsk1-02	mv01-03	dsk1	0	65	LOG	dsk1	ENA

By default, LSM creates a DRL plex for all mirrored volumes.

- To create an LSM volume with mirrored, striped plexes on the disks you specify:

```
# volassist [-g disk_group] [-U use_type] make volume length \
mirror=2 layout=stripe ncolumn=number_of_columns \
[stwid=data_unit_size] disks
```

For example:

```
# volassist -g dg1 make mv01 256m mirror=2 layout=stripe \
ncolumn=2 stwid=32k dsk19 dsk20 dsk21 dsk22 dsk23
```

The volume looks similar to the following:

Disk group: `dg1`

TY	NAME	ASSOC	KSTATE	LENGTH	PLOFFS	STATE	TUTILO	PUTILO
v	mvol	fsgen	ENABLED	524288	-	ACTIVE	-	-
pl	mvol-01	mvol	ENABLED	524288	-	ACTIVE	-	-
sd	dsk19-01	mvol-01	ENABLED	262144	0	-	-	-
sd	dsk20-01	mvol-01	ENABLED	262144	0	-	-	-
pl	mvol-02	mvol	ENABLED	524288	-	ACTIVE	-	-
sd	dsk21-01	mvol-02	ENABLED	262144	0	-	-	-
sd	dsk22-01	mvol-02	ENABLED	262144	0	-	-	-
pl	mvol-03	mvol	ENABLED	LOGONLY	-	ACTIVE	-	-
sd	dsk19-02	mvol-03	ENABLED	65	LOG	-	-	-

By default, LSM creates a DRL plex for all mirrored volumes. Notice that LSM did not use `dsk23`, because there was space for the DRL on one of the disks used in a data plex. You can add a new log plex, specifying a disk not used in the volume (Section 5.5.3) and remove the original log plex (Section 5.5.6).

4.3.5.2 Creating a Mirrored, Striped Volume with Plexes on Different Buses

For improved availability, you can create an LSM volume with each data plex and the log plex on a different bus. If your configuration does not support this, you can still use the following procedure to specify which disks LSM uses to create each data plex and the log plex.

The following procedure shows how to ensure that LSM creates each plex on the disks you specify. The `ncolumn` option forces LSM to stripe the plex across all the named disks. The `ncolumn` value must equal the number of disks you specify.

Note

Each data plex is a complete copy of the volume, and uses as much disk space as the volume size you specify.

1. Create the volume with a single striped plex, specifying disks on one bus:

```
# volassist [-g disk_group] [-U use_type] make volume length \
layout=stripe ncolumn=number_of_columns \
[stwid=data_unit_size] disks
```

For example, to create a 1GB volume named `vstripe` in the `dg1` disk group with the default usage type of `fsgen` and one 3-column striped plex with the default stripe width on disks `dsk10`, `dsk11`, and `dsk12`:

```
# volassist -g dg1 make vstripe 1g layout=stripe \
ncolumn=3 dsk10 dsk11 dsk12
```

The volume looks similar to the following:

Disk group: `dg1`

V	NAME	USETYPE	KSTATE	STATE	LENGTH	READPOL	PREFPLEX
PL	NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WID MODE
SD	NAME	PLEX	DISK	DISKOFFS	LENGTH	[COL/]OFF	DEVICE MODE

```

v vstripe      fsgen          ENABLED ACTIVE 2097152 SELECT vstripe-01
pl vstripe-01  vstripe        ENABLED ACTIVE 2097408 STRIPE 3/128 RW
sd dsk10-01    vstripe-01     dsk10      0      699136 0/0      dsk10    ENA
sd dsk11-01    vstripe-01     dsk11      0      699136 1/0      dsk11    ENA
sd dsk12-01    vstripe-01     dsk12      0      699136 2/0      dsk12    ENA

```

2. Add a mirror plex to the volume, specifying disks on a different bus:

```
# volassist [-g disk_group] mirror volume disks
```

For example:

```
# volassist -g dg1 mirror vol_stripe dsk19 dsk20 dsk21
```

3. Add a DRL plex to the volume, specifying a disk that is not used by one of the data plexes, and if possible, on a different bus:

```
# volassist [-g disk_group] addlog volume disk
```

For example:

```
# volassist -g dg1 addlog vol_stripe dsk26
```

The completed volume is ready for use and looks similar to the following:

Disk group: dg1

V NAME	USETYPE	KSTATE	STATE	LENGTH	READPOL	PREFPLEX		
PL NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WID	MODE	
SD NAME	PLEX	DISK	DISKOFFS	LENGTH	[COL/]OFF	DEVICE	MODE	
v vstripe	fsgen	ENABLED	ACTIVE	2097152	SELECT	-		
pl vstripe-01	vstripe	ENABLED	ACTIVE	2097408	STRIPE	3/128	RW	
sd dsk10-01	vstripe-01	dsk10	0	699136	0/0	dsk10	ENA	
sd dsk11-01	vstripe-01	dsk11	0	699136	1/0	dsk11	ENA	
sd dsk12-01	vstripe-01	dsk12	0	699136	2/0	dsk12	ENA	
pl vstripe-02	vstripe	ENABLED	ACTIVE	2097408	STRIPE	3/128	RW	
sd dsk19-01	vstripe-02	dsk19	0	699136	0/0	dsk19	ENA	
sd dsk20-01	vstripe-02	dsk20	0	699136	1/0	dsk20	ENA	
sd dsk21-01	vstripe-02	dsk21	0	699136	2/0	dsk21	ENA	
pl vstripe-03	vstripe	ENABLED	ACTIVE	LOGONLY	CONCAT	-	RW	
sd dsk26-01	vstripe-03	dsk26	0	65	LOG	dsk26	ENA	

4.3.6 Creating LSM Volumes with a RAID 5 Plex

A volume with a RAID 5 data plex uses distributed parity to provide data redundancy. When you create the volume, you can use the default values for the number of columns in the plex (minimum of three, maximum of eight) and the stripe width (16K bytes), as well as let LSM use any available space in the disk group to create the volume. Or you can specify the number of columns, the stripe width, and the disks to use.

If you specify the disks to use, by default the `volassist` command creates the columns for a RAID 5 plex on disks in alphanumeric order, regardless of their order on the command line, and automatically creates a RAID 5 log plex for the volume on a separate disk. LSM will not create the log plex for

a RAID 5 volume on a disk used by the data plex, as it does for mirrored volumes. You must specify enough disks to create the data plex and log plex.

To improve performance, you can create the columns on disks on different buses. For more information about specifying the disk order for columns in a RAID 5 plex, see Section 4.4.3.

The usage type for all volumes with a RAID 5 plex is `raid5`, regardless of what the volume is used for. When you specify `layout=raid5`, LSM automatically applies the `raid5` usage type.

- To create an LSM volume that uses a RAID 5 plex with the default values on any available disks in the disk group:

```
# volassist [-g disk_group] make volume length \
layout=raid5 [ncolumn=number_of_columns]
```

For example, to create a 250 MB volume named `volr5` with the default number of columns on any available disks in the disk group:

```
# volassist make volr5 250m layout=raid5
```

The volume looks similar to the following:

```
Disk group: rootdg
```

V NAME	USETYPE	KSTATE	STATE	LENGTH	READPOL	PREFPLEX	
PL NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WID	MODE
SD NAME	PLEX	DISK	DISKOFFS	LENGTH	[COL/]OFF	DEVICE	MODE
v volr5	raid5	ENABLED	ACTIVE	512064	RAID	-	
pl volr5-01	volr5	ENABLED	ACTIVE	512064	RAID	8/32	RW
sd dsk1-01	volr5-01	dsk1	0	73152	0/0	dsk1	ENA
sd dsk2-01	volr5-01	dsk2	0	73152	1/0	dsk2	ENA
sd dsk3-01	volr5-01	dsk3	0	73152	2/0	dsk3	ENA
sd dsk4-01	volr5-01	dsk4	0	73152	3/0	dsk4	ENA
sd dsk5-01	volr5-01	dsk5	0	73152	4/0	dsk5	ENA
sd dsk6-01	volr5-01	dsk6	0	73152	5/0	dsk6	ENA
sd dsk7-01	volr5-01	dsk7	0	73152	6/0	dsk7	ENA
sd dsk9-01	volr5-01	dsk9	0	73152	7/0	dsk9	ENA
pl volr5-02	volr5	ENABLED	LOG	2560	CONCAT	-	RW
sd dsk10-01	volr5-02	dsk10	0	2560	0	dsk10	ENA

By default, LSM displays stripe width in blocks; 16K bytes is 32 blocks.

- To create an LSM volume with a specific number of columns and stripe width:

```
# volassist [-g disk_group] make volume length \
layout=raid5 ncolumn=number_of_columns stwid=stripe_width
```

For example, to create a 250 MB volume named `5way` with five columns and a stripe width of 32K bytes on any available disks in the `rootdg` disk group:

```
# volassist make 5way 250m layout=raid5 ncolumn=5 stwid=32k
```

The volume looks similar to the following:

Disk group: rootdg

V NAME	USETYPE	KSTATE	STATE	LENGTH	READPOL	PREFPLEX		
PL NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WID	MODE	
SD NAME	PLEX	DISK	DISKOFFS	LENGTH	[COL/]OFF	DEVICE	MODE	
v 5way	raid5	ENABLED	ACTIVE	512000	RAID	-		
pl 5way-01	5way	ENABLED	ACTIVE	512000	RAID	5/64	RW	
sd dsk9-02	5way-01	dsk9	32768	128000	0/0	dsk9	ENA	
sd dsk10-02	5way-01	dsk10	32768	128000	1/0	dsk10	ENA	
sd dsk11-02	5way-01	dsk11	32768	128000	2/0	dsk11	ENA	
sd dsk8-03	5way-01	dsk8	35008	128000	3/0	dsk8	ENA	
sd dsk15-02	5way-01	dsk15	85344	128000	4/0	dsk15	ENA	
pl 5way-02	5way	ENABLED	LOG	3200	CONCAT	-	RW	
sd dsk16-02	5way-02	dsk16	85344	3200	0	dsk16	ENA	

By default, LSM displays stripe width in blocks; 32K bytes is 64 blocks.

- To specify the disks for LSM to use:

```
# volassist [-g disk_group] make volume length \  
layout=raid5 [ncolumn=number_of_columns] \  
[stwid=stripe_width] disks
```

For example, to create a volume called 4way with a 4-column plex using the default stripe width on disks dsk12, dsk13, dsk14, dsk18, and dsk19 (for the log plex) in the rootdg disk group:

```
# volassist make 4way 250m layout=raid5 ncolumn=4 \  
dsk12 dsk13 dsk14 dsk18 dsk19
```

The volume looks similar to the following:

Disk group: rootdg

V NAME	USETYPE	KSTATE	STATE	LENGTH	READPOL	PREFPLEX		
PL NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WID	MODE	
SD NAME	PLEX	DISK	DISKOFFS	LENGTH	[COL/]OFF	DEVICE	MODE	
v 4way	raid5	ENABLED	ACTIVE	512064	RAID	-		
pl 4way-01	4way	ENABLED	ACTIVE	512064	RAID	4/32	RW	
sd dsk12-01	4way-01	dsk12	0	170688	0/0	dsk12	ENA	
sd dsk13-01	4way-01	dsk13	0	170688	1/0	dsk13	ENA	
sd dsk14-01	4way-01	dsk14	0	170688	2/0	dsk14	ENA	
sd dsk18-01	4way-01	dsk18	0	170688	3/0	dsk18	ENA	
pl 4way-02	4way	ENABLED	LOG	1280	CONCAT	-	RW	
sd dsk19-01	4way-02	dsk19	0	1280	0	dsk19	ENA	

4.3.7 Creating LSM Volumes for Swap Space

To protect against system or cluster member crashes due to swap disk errors, you can create LSM mirrored volumes for swap space. For recommendations on the amount of swap space to configure, see the *System Administration* manual or the *Cluster Administration* manual.

HP recommends that you use multiple disks for secondary swap devices and add the devices as several individual volumes, instead of striping or concatenating them into a single large volume. This makes the swapping algorithm more efficient.

The way you use LSM volumes for swap space differs depending on the environment:

- **On a standalone system**, to use LSM volumes for secondary swap, you must also use LSM for the root partition and primary swap space by encapsulating the boot disk (Section 3.4.1).
- **In a cluster**, there is no clusterwide swap; each member has its own private swap devices. You can use LSM volumes for cluster members' swap space independent of whether or not the clusterwide root, /usr, and /var file system domains are under LSM control.

You can encapsulate the swap devices for one or more members to LSM volumes (Section 3.4.3) and create additional swap volumes for one or more cluster members (Section 4.3.7.1).

All swap volumes for both standalone systems and cluster members must belong to the `rootdg` disk group. If there is not enough free disk space to create the volumes, add more disks to `rootdg` (see Section 5.2.2).

4.3.7.1 Creating Swap Volumes

Swap volumes, if mirrored, should not use dirty region logging (DRL). After you create a swap volume, you should also modify the volume's recovery policy, so that LSM will not resynchronize the plexes after a system failure. On a cluster member, choose disks that are local to the member, if possible. The disks must belong to the `rootdg` disk group.

To create an LSM volume for secondary swap space on a standalone system or additional swap space on a cluster member:

1. Create the volume with no DRL plex, if mirrored. Assign the volume one of the following usage types:
 - On a standalone system, use the `gen` usage type for secondary swap volumes.
 - In a cluster, use the `swap` usage type for all members' swap volumes.

```
# volassist -U use_type make volume length [nmirror=count] \  
[layout=nolog] [disks]
```

For example:

- On a standalone system, to create a mirrored secondary swap volume in the `rootdg` disk group on any available disks in the disk group:

```
# volassist -U gen make swapvol_2 128m nmirror=2 \  
layout=nolog
```

- On a cluster member, to create a mirrored swap volume in the `rootdg` disk group using disks `dsk4` and `dsk5` (which are local to the member and belong to `rootdg`):

```
# volassist -U swap make member1_swap 128m nmirror=2 \  
layout=nolog dsk4 dsk5
```

2. If the volume is mirrored, change the volume's recovery policy to prevent plex resynchronization after a system crash; for example:

```
# volume set start_opts=norecov swapvol_2
```

3. Make the LSM volume available as swap space using the `swapon` command; for example:

```
# swapon /dev/vol/rootdg/swapvol_2
```

4. Edit the `sysconfigtab` file to add the volume's device special file to the `swapdevice` kernel attribute value within the `vm:` section.

For example:

```
vm:  
swapdevice = /dev/vol/rootdg/swapvol, /dev/vol/rootdg/swapvol_2
```

In a cluster, be sure to modify the appropriate member's file, which is a context-dependent symbolic link (CDSL) in the form `/cluster/members/member{n}/boot_partition/etc/sysconfigtab`.

4.3.7.2 Mirroring Swap Volumes

The following procedure applies only to secondary swap volumes on a standalone system and to any unmirrored swap volume on a cluster member.

Note

To mirror the primary swap volume on a standalone system, use the `volrootmir` command. See Section 3.4.2.1.3.

In a cluster, choose an LSM disk that belongs to the `rootdg` disk group and is local to the member whose swap volume you want to mirror. You can run the command from any cluster member.

To mirror a secondary swap volume on a standalone system or any unmirrored swap volume on a cluster member:

1. Mirror the volume. In a cluster, specify a disk that is local to the appropriate cluster member:

```
# volassist mirror volume [dskN]
```

For example:

- To mirror a secondary swap volume (`swapvol_2`) on a standalone system using any available disk in `rootdg`:

```
# volassist mirror swapvol_2
```

- To mirror a swap volume (`joey_swap`) for a cluster member using disk `dsk5` (which is local to that member):

```
# volassist mirror joey_swap dsk5
```

2. Change the volume's recovery policy to prevent plex resynchronization after a system crash; for example:

```
# volume set start_opts=norecov swapvol_2
```

For more information on configuring additional swap space, see `swapon(8)`, and `sysconfig(8)`.

4.4 Creating LSM Volumes with Nondefault Properties

This section describes how to create LSM volumes with attributes or properties that you cannot always specify with the high-level commands. Only users who have a thorough understanding of LSM, and the need to create such volumes, are advised to use these procedures.

4.4.1 Creating a Striped Plex with Subdisks of Different Sizes

If you have LSM disks of different sizes and want to use space on particular disks to maximize your use of storage, you can use the low-level commands to manually create the subdisks and plex columns, then create the plex, and then create and start the volume.

The makeup of each column can differ, but each column should total the same number of sectors. For example, each column can contain a different number of subdisks, of different sizes.

Each subdisk should be a multiple of the stripe width (and, therefore, so will each column) so that writes align evenly to subdisk boundaries.

The following table shows how three columns that each total 256000 sectors (125 MB) can comprise different numbers and sizes of subdisks. The three columns make up one plex of 768000 sectors (375 MB). These subdisk names and sizes are used as examples in the following procedure.

Striped Plex: plex-01		
Column 0	Column 1	Column 2
Subdisk dsk3-01, 128000 Sectors	Subdisk dsk6-01, 64000 Sectors	Subdisk dsk13-01, 51200 Sectors
Subdisk dsk4-01, 128000 Sectors	Subdisk dsk9-01, 64000 Sectors	Subdisk dsk14-01, 51200 Sectors
	Subdisk dsk11-01, 64000 Sectors	Subdisk dsk15-01, 51200 Sectors
	Subdisk dsk12-01, 64000 Sectors	Subdisk dsk16-01, 51200 Sectors
		Subdisk dsk17-01, 51200 Sectors
Total: 256000 Sectors (125 MB)	Total: 256000 Sectors (125 MB)	Total: 256000 Sectors (125 MB)

The following procedure shows how to make subdisks of different sizes, create a plex with columns of equal size, and create a volume using that plex. The plex uses the default stripe width of 64K bytes and has three columns, with the subdisk sizes shown in the previous table.

Each subdisk is on a different disk and starts at offset 0 in the public region; therefore you need specify only a length for the subdisk. (To create a subdisk on a disk that already has other subdisks, you must specify the offset — the starting point — of the new subdisk as well as its length.)

To create a volume with different sized subdisks:

1. Decide the size of the volume and the number of columns you want in the plex (or each plex, if mirrored); for the examples that follow, the volume is 375 MB, so each column is one-third of 375, or 125 MB.
2. Divide the column size (125 MB or 256000 sectors) by the number of intended subdisks per column (in the following examples, 2, 4, and 5) to determine the size of each subdisk. In the first column (two subdisks), each subdisk will be 256000/2, or 128000 sectors, and so on.

Alternatively, you can work backwards from the amount of space available on various disks, determine how large a subdisk you can create in that space that is a multiple of the stripe width, and calculate how many subdisks you need for each column. The space availability might influence the number of columns.

3. Create the subdisks:

```
volmake sd subdisk_name disk len=length
```

For example:

```
# volmake sd dsk3-01 dsk3 len=128000
# volmake sd dsk4-01 dsk4 len=128000

# volmake sd dsk6-01 dsk6 len=64000
# volmake sd dsk9-01 dsk9 len=64000
# volmake sd dsk11-01 dsk11 len=64000
# volmake sd dsk12-01 dsk12 len=64000

# volmake sd dsk13-01 dsk13 len=51200
# volmake sd dsk14-01 dsk14 len=51200
# volmake sd dsk15-01 dsk15 len=51200
# volmake sd dsk16-01 dsk16 len=51200
# volmake sd dsk17-01 dsk17 len=51200
```

4. Create a plex with the desired number of columns and associate the appropriate subdisks to each column.

If there are many subdisks involved, you can create an “empty” plex first, then associate the subdisk groups in separate steps, one step for each column.

You do not need to specify a size for the empty plex; as you associate subdisk columns, the plex size is updated to map to the end of the longest column. When you create a plex manually, you must set a stripe width.

For example:

```
# volmake plex plex-01 layout=stripe ncolumn=3 stwidth=64k

# volsd assoc plex-01 dsk3-01:0 dsk4-01:0
# volsd assoc plex-01 dsk6-01:1 dsk9-01:1 dsk11-01:1 dsk12-01:1
# volsd assoc plex-01 dsk13-01:2 dsk14-01:2 dsk15-01:2 \
dsk16-01:2 dsk17-01:2
```

The plex looks similar to the following:

```
# volprint -pht plex-01
```

Disk group: rootdg

PL NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WID	MODE
SD NAME	PLEX	DISK	DISKOFFS	LENGTH	[COL/]OFF	DEVICE	MODE
pl plex-01	-	DISABLED	-	768000	STRIPE	3/128	RW
sd dsk3-01	plex-01	dsk3	0	128000	0/0	dsk3	ENA
sd dsk4-01	plex-01	dsk4	0	128000	0/128000	dsk4	ENA
sd dsk6-01	plex-01	dsk6	0	64000	1/0	dsk6	ENA
sd dsk9-01	plex-01	dsk9	0	64000	1/64000	dsk9	ENA
sd dsk11-01	plex-01	dsk11	0	64000	1/128000	dsk11	ENA
sd dsk12-01	plex-01	dsk12	0	64000	1/192000	dsk12	ENA
sd dsk13-01	plex-01	dsk13	0	51200	2/0	dsk13	ENA

```
sd dsk14-01    plex-01    dsk14    0        51200    2/51200    dsk14    ENA
sd dsk15-01    plex-01    dsk15    0        51200    2/102400   dsk15    ENA
sd dsk16-01    plex-01    dsk16    0        51200    2/153600   dsk16    ENA
sd dsk17-01    plex-01    dsk17    0        51200    2/204800   dsk17    ENA
```

5. Create the volume using the plex:

```
# volmake -U use_type vol volume_name plex=plex_name
```

For example:

```
# volmake -U fsgen vol vol-01 plex=plex-01
```

6. Start the volume:

```
# volume start vol-01
```

The volume is started and ready for use, and looks similar to the following:

```
# volprint -vht vol-01
```

Disk group: rootdg

V NAME	USETYPE	KSTATE	STATE	LENGTH	READPOL	PREFPLEX		
PL NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WID	MODE	
SD NAME	PLEX	DISK	DISKOFFS	LENGTH	[COL/]OFF	DEVICE	MODE	
v vol-01	fsgen	ENABLED	ACTIVE	768000	ROUND	-		
pl plex-01	vol-01	ENABLED	ACTIVE	768000	STRIPE	3/128	RW	
sd dsk3-01	plex-01	dsk3	0	128000	0/0	dsk3	ENA	
sd dsk4-01	plex-01	dsk4	0	128000	0/128000	dsk4	ENA	
sd dsk6-01	plex-01	dsk6	0	64000	1/0	dsk6	ENA	
sd dsk9-01	plex-01	dsk9	0	64000	1/64000	dsk9	ENA	
sd dsk11-01	plex-01	dsk11	0	64000	1/128000	dsk11	ENA	
sd dsk12-01	plex-01	dsk12	0	64000	1/192000	dsk12	ENA	
sd dsk13-01	plex-01	dsk13	0	51200	2/0	dsk13	ENA	
sd dsk14-01	plex-01	dsk14	0	51200	2/51200	dsk14	ENA	
sd dsk15-01	plex-01	dsk15	0	51200	2/102400	dsk15	ENA	
sd dsk16-01	plex-01	dsk16	0	51200	2/153600	dsk16	ENA	
sd dsk17-01	plex-01	dsk17	0	51200	2/204800	dsk17	ENA	

4.4.2 Creating a Striped Plex with Disks on Different Buses

You can improve performance for a volume with striped plexes by striping each plex over disks on different buses. If you have enough buses, you can mirror the volume on different buses from those supporting the first plex. For example, if you had 12 buses, you could stripe one plex over the first six buses, and stripe a second plex over the other six buses.

Caution

For mirrored volumes, do not create all the data plexes on the same bus; this eliminates the volume's availability. If the bus fails, you lose the entire volume.

- If you have limited buses and want to create mirrored, striped volumes, you should create one striped plex using disks on

the same bus and create the second striped plex using disks on another bus. If one bus fails, the volume still has a plex on another bus.

- If the volume will have only one striped plex, you can stripe the plex over all the available buses to improve performance.

Before you begin, decide which LSM disks you want to use, identify which bus each disk is on, and plan how you want to create the volume based on how you want LSM to stripe and mirror the data over the buses.

To create a volume with plexes that are striped down buses and mirrored across buses, you must use low-level commands to create each subdisk, create each plex from those subdisks, create the volume from the plexes, add a log plex, and start the volume.

The following procedure creates a volume with two plexes and a DRL plex using these disks and buses in the `rootdg` disk group:

Plex plex-01		Plex plex-02	
Bus 1	Bus 2	Bus 3	Bus 4
dsk0	dsk7	dsk14	dsk21
dsk1	dsk8	dsk15	dsk22
dsk2	dsk9	dsk16	dsk23
dsk3	dsk10	dsk17	dsk24
dsk4 (for the DRL plex)			

The first plex will stripe alternately over buses 1 and 2. The second plex will stripe alternately over buses 3 and 4. The log plex will be on bus 1, the same as the first plex, but on a different disk. Although recommended, it is not always possible to place the log plex on a different bus from the data plexes.

To create an LSM mirrored, striped volume with each plex on different buses:

1. Create the subdisks on the disks on bus 1 and 2; for example:

```
# volmake sd dsk0-01 dsk0 len=16m
# volmake sd dsk1-01 dsk1 len=16m
# volmake sd dsk2-01 dsk2 len=16m
# volmake sd dsk3-01 dsk3 len=16m

# volmake sd dsk7-01 dsk7 len=16m
# volmake sd dsk8-01 dsk8 len=16m
# volmake sd dsk9-01 dsk9 len=16m
# volmake sd dsk10-01 dsk10 len=16m
```


2. Create the striped plex, specifying the subdisks in alternating bus order; for example:

```
# volmake plex plex-01 layout=stripe stwidth=64k \  
sd=dsk0-01,dsk7-01,dsk1-01,dsk8-01,dsk2-01,dsk9-01, \  
dsk3-01,dsk10-01
```

This creates an 8-column striped plex that alternates between buses 1 and 2. It is more efficient to stripe over all the disks in a plex.

The plex looks similar to the following:

```
# volprint -p plex-01
```

Disk group: rootdg

PL NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WID	MODE
SD NAME	PLEX	DISK	DISKOFFS	LENGTH	[COL/]OFF	DEVICE	MODE
pl plex-01	-	DISABLED	-	262144	STRIPE	8/128	RW
sd dsk0-01	plex-01	dsk0	0	32768	0/0	dsk0	ENA
sd dsk7-01	plex-01	dsk7	0	32768	1/0	dsk7	ENA
sd dsk1-01	plex-01	dsk1	0	32768	2/0	dsk1	ENA
sd dsk8-01	plex-01	dsk8	0	32768	3/0	dsk8	ENA
sd dsk2-01	plex-01	dsk2	0	32768	4/0	dsk2	ENA
sd dsk9-01	plex-01	dsk9	0	32768	5/0	dsk9	ENA
sd dsk3-01	plex-01	dsk3	0	32768	6/0	dsk3	ENA
sd dsk10-01	plex-01	dsk10	0	32768	7/0	dsk10	ENA

3. Create the subdisks for the second data plex on the disks on bus 3 and 4; for example:

```
# volmake sd dsk14-01 dsk14 len=16m  
# volmake sd dsk15-01 dsk15 len=16m  
# volmake sd dsk16-01 dsk16 len=16m  
# volmake sd dsk17-01 dsk17 len=16m  
  
# volmake sd dsk21-01 dsk21 len=16m  
# volmake sd dsk22-01 dsk22 len=16m  
# volmake sd dsk23-01 dsk23 len=16m  
# volmake sd dsk24-01 dsk24 len=16m
```

4. Create the second data plex, specifying the subdisks in alternating bus order; for example:

```
# volmake plex plex-02 layout=stripe stwidth=64k \  
sd=dsk14-01,dsk21-01,dsk15-01,dsk22-01,dsk16-01,dsk23-01, \  
dsk17-01,dsk24-01
```

This creates an 8-column striped plex that alternates between buses 3 and 4.

The plex looks similar to the following:

```
# volprint -pht plex-02
```

Disk group: rootdg

PL NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WID	MODE
SD NAME	PLEX	DISK	DISKOFFS	LENGTH	[COL/]OFF	DEVICE	MODE

```

pl plex-02      -          DISABLED -          262144  STRIPE    8/128    RW
sd dsk14-01    plex-02    dsk14    0          32768    0/0      dsk14    ENA
sd dsk21-01    plex-02    dsk21    0          32768    1/0      dsk21    ENA
sd dsk15-01    plex-02    dsk15    0          32768    2/0      dsk15    ENA
sd dsk22-01    plex-02    dsk22    0          32768    3/0      dsk22    ENA
sd dsk16-01    plex-02    dsk16    0          32768    4/0      dsk16    ENA
sd dsk23-01    plex-02    dsk23    0          32768    5/0      dsk23    ENA
sd dsk17-01    plex-02    dsk17    0          32768    6/0      dsk17    ENA
sd dsk24-01    plex-02    dsk24    0          32768    7/0      dsk24    ENA

```

5. Create the LSM volume using both data plexes; for example:

```
# volmake vol vol_mirr plex=plex-01,plex-02
```

The volume looks similar to the following:

```
# volprint -vht vol_mirr

Disk group: rootdg

V NAME      USETYPE      KSTATE  STATE  LENGTH  READPOL  PREFPLEX
PL NAME     VOLUME       KSTATE  STATE  LENGTH  LAYOUT   NCOL/WID  MODE
SD NAME     PLEX         DISK    DISKOFFS  LENGTH  [COL/]OFF  DEVICE    MODE

v vol_mirr  fsgen        DISABLED  EMPTY  262144  ROUND    -
pl plex-01  vol_mirr     DISABLED  EMPTY  262144  STRIPE   8/128    RW
sd dsk0-01  plex-01     dsk0     0      32768    0/0      dsk0     ENA
sd dsk7-01  plex-01     dsk7     0      32768    1/0      dsk7     ENA
sd dsk1-01  plex-01     dsk1     0      32768    2/0      dsk1     ENA
sd dsk8-01  plex-01     dsk8     0      32768    3/0      dsk8     ENA
sd dsk2-01  plex-01     dsk2     0      32768    4/0      dsk2     ENA
sd dsk9-01  plex-01     dsk9     0      32768    5/0      dsk9     ENA
sd dsk3-01  plex-01     dsk3     0      32768    6/0      dsk3     ENA
sd dsk10-01 plex-01     dsk10    0      32768    7/0      dsk10    ENA
pl plex-02  vol_mirr     DISABLED  EMPTY  262144  STRIPE   8/128    RW
sd dsk14-01 plex-02     dsk14    0      32768    0/0      dsk14    ENA
sd dsk21-01 plex-02     dsk21    0      32768    1/0      dsk21    ENA
sd dsk15-01 plex-02     dsk15    0      32768    2/0      dsk15    ENA
sd dsk22-01 plex-02     dsk22    0      32768    3/0      dsk22    ENA
sd dsk16-01 plex-02     dsk16    0      32768    4/0      dsk16    ENA
sd dsk23-01 plex-02     dsk23    0      32768    5/0      dsk23    ENA
sd dsk17-01 plex-02     dsk17    0      32768    6/0      dsk17    ENA
sd dsk24-01 plex-02     dsk24    0      32768    7/0      dsk24    ENA

```

6. Add a DRL plex to the volume, if possible, specifying a disk that is not used by one of the data plexes; for example:

```
# volassist addlog vol_mirr dsk4
```

The volume looks similar to the following:

```
Disk group: rootdg

V NAME      USETYPE      KSTATE  STATE  LENGTH  READPOL  PREFPLEX
PL NAME     VOLUME       KSTATE  STATE  LENGTH  LAYOUT   NCOL/WID  MODE
SD NAME     PLEX         DISK    DISKOFFS  LENGTH  [COL/]OFF  DEVICE    MODE

v vol_mirr  fsgen        DISABLED  EMPTY  262144  ROUND    -
pl plex-01  vol_mirr     DISABLED  EMPTY  262144  STRIPE   8/128    RW
sd dsk0-01  plex-01     dsk0     0      32768    0/0      dsk0     ENA
sd dsk7-01  plex-01     dsk7     0      32768    1/0      dsk7     ENA
sd dsk1-01  plex-01     dsk1     0      32768    2/0      dsk1     ENA
sd dsk8-01  plex-01     dsk8     0      32768    3/0      dsk8     ENA
sd dsk2-01  plex-01     dsk2     0      32768    4/0      dsk2     ENA

```

```

sd dsk9-01      plex-01      dsk9      0      32768      5/0      dsk9      ENA
sd dsk3-01      plex-01      dsk3      0      32768      6/0      dsk3      ENA
sd dsk10-01     plex-01      dsk10     0      32768      7/0      dsk10     ENA
pl plex-02      vol_mirr     DISABLED  EMPTY  262144    STRIPE   8/128    RW
sd dsk14-01     plex-02      dsk14     0      32768      0/0      dsk14     ENA
sd dsk21-01     plex-02      dsk21     0      32768      1/0      dsk21     ENA
sd dsk15-01     plex-02      dsk15     0      32768      2/0      dsk15     ENA
sd dsk22-01     plex-02      dsk22     0      32768      3/0      dsk22     ENA
sd dsk16-01     plex-02      dsk16     0      32768      4/0      dsk16     ENA
sd dsk23-01     plex-02      dsk23     0      32768      5/0      dsk23     ENA
sd dsk17-01     plex-02      dsk17     0      32768      6/0      dsk17     ENA
sd dsk24-01     plex-02      dsk24     0      32768      7/0      dsk24     ENA
pl vol_mirr-01  vol_mirr     DISABLED  EMPTY  LOGONLY   CONCAT  -        RW
sd dsk4-01      vol_mirr-01 dsk4      0      65        LOG      dsk4      ENA

```

7. Start the LSM volume; for example:

```
# volume start vol_mirr
The volume is ready for use.
```

4.4.3 Creating a RAID 5 Plex with Disks on Different Buses

You can improve performance for a volume with a RAID 5 plex by striping the plex over disks on different buses.

- To create the volume with a plex striped over only one disk on each bus, you can use the `volassist` command and specify the disks. See Section 4.3.6.
- To create the volume with a plex striped over more than one disk per bus, you must use low-level commands to create the subdisks for each column, create the plex, and create and start the volume, because the `volassist` command might not use the disks in the order you specify on the command line.

Before you begin, decide which LSM disks you want to use, identify which bus each disk is on, and plan how you want to create the volume based on how you want LSM to stripe the RAID 5 data plex over the buses.

Each column of subdisks should be the same size and be a multiple of the data unit size. For example, a data unit size (stripe width) of 16K bytes for a RAID 5 plex corresponds to 32 blocks (sectors), so the total of the subdisks in each column should be a multiple of 32.

If each column consists of one subdisk (the typical configuration), then the subdisk size should be a multiple of 32. If a column consists of two subdisks, each subdisk can be a different size as long as the total is a multiple of 32.

You can optionally identify a separate disk, ideally on a different bus, for the RAID 5 log plex, or allow LSM to use any available disk.

Caution

If both the log plex and one column of the data plex are on the same bus and the bus fails, you lose the entire volume. If possible, put the log plex on a bus that does not support a disk used in the data plex.

The following procedure creates a RAID 5 volume using these disks and buses in the `rootdg` disk group:

Bus 1	Bus 2	Bus 3	Bus 4	Bus 5
dsk1	dsk4	dsk8	dsk11	dsk22 (for the RAID 5 log plex)
dsk2	dsk5	dsk9	dsk12	

The finished volume will stripe data and parity alternately over buses 1 through 4. The log plex will be on bus 5.

To create a RAID 5 plex with disks on different buses:

1. Create the subdisks; for example:

```
# volmake sd dsk1-01 dsk1 len=1m
# volmake sd dsk2-01 dsk2 len=1m
# volmake sd dsk4-01 dsk4 len=1m
# volmake sd dsk5-01 dsk5 len=1m
# volmake sd dsk8-01 dsk8 len=1m
# volmake sd dsk9-01 dsk9 len=1m
# volmake sd dsk11-01 dsk11 len=1m
# volmake sd dsk12-01 dsk12 len=1m
```

2. Create the RAID 5 data plex, specifying the subdisks in an order that alternates between the buses in the order you want; for example:

```
# volmake plex plex_r5 layout=raid5 stwidth=16k \
sd=dsk1-01,dsk4-01,dsk8-01,dsk11-01,dsk2-01,dsk5-01, \
dsk9-01,dsk12-01
```

This creates an eight-column RAID 5 data plex named `plex_r5`, where the columns proceed from bus 1 through bus 4, then repeat.

The plex looks similar to the following:

Disk group: `rootdg`

PL NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WID	MODE
SD NAME	PLEX	DISK	DISKOFFS	LENGTH	[COL/]OFF	DEVICE	MODE
pl plex_r5	-	DISABLED	-	14336	RAID	8/32	RW
sd dsk1-01	plex_r5	dsk1	0	2048	0/0	dsk1	ENA
sd dsk4-01	plex_r5	dsk4	0	2048	1/0	dsk4	ENA
sd dsk8-01	plex_r5	dsk8	0	2048	2/0	dsk8	ENA
sd dsk11-01	plex_r5	dsk11	0	2048	3/0	dsk11	ENA
sd dsk2-01	plex_r5	dsk2	0	2048	4/0	dsk2	ENA

```
sd dsk5-01      plex_r5      dsk5      0          2048      5/0       dsk5      ENA
sd dsk9-01      plex_r5      dsk9      0          2048      6/0       dsk9      ENA
sd dsk12-01     plex_r5      dsk12     0          2048      7/0       dsk12     ENA
```

3. Create the LSM volume using the data plex; for example:

```
# volmake -U raid5 vol1 vol15 plex=plex_r5
```

This creates an LSM volume named `volr5` from plex `plex_r5`, with a usage type of `raid5` (required for all volumes with a RAID 5 plex).

4. Add a RAID 5 log plex to the volume, optionally specifying the disk (by default, LSM uses a disk not already used in the volume, if available); for example:

```
# volassist addlog volr5 dsk22
```

5. Start the LSM volume; for example:

```
# volume start volr5
```

The volume is ready for use and looks similar to the following:

```
Disk group: rootdg
```

V NAME	USETYPE	KSTATE	STATE	LENGTH	READPOL	PREFPLEX		
PL NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WID	MODE	
SD NAME	PLEX	DISK	DISKOFFS	LENGTH	[COL/]OFF	DEVICE	MODE	
v volr5	raid5	ENABLED	ACTIVE	14336	RAID	-		
pl plex_r5	volr5	ENABLED	ACTIVE	14336	RAID	8/32	RW	
sd dsk1-01	plex_r5	dsk1	0	2048	0/0	dsk1	ENA	
sd dsk4-01	plex_r5	dsk4	0	2048	1/0	dsk4	ENA	
sd dsk8-01	plex_r5	dsk8	0	2048	2/0	dsk8	ENA	
sd dsk11-01	plex_r5	dsk11	0	2048	3/0	dsk11	ENA	
sd dsk2-01	plex_r5	dsk2	0	2048	4/0	dsk2	ENA	
sd dsk5-01	plex_r5	dsk5	0	2048	5/0	dsk5	ENA	
sd dsk9-01	plex_r5	dsk9	0	2048	6/0	dsk9	ENA	
sd dsk12-01	plex_r5	dsk12	0	2048	7/0	dsk12	ENA	
pl volr5-01	volr5	ENABLED	LOG	2560	CONCAT	-	RW	
sd dsk22-01	volr5-01	dsk22	0	2560	0	dsk22	ENA	

4.5 Configuring File Systems to Use LSM Volumes

After you create an LSM volume, you can use it the same way you can use a disk partition. Because LSM uses the same interfaces as disk device drivers, you can specify an LSM volume in any operation where you can specify a disk or disk partition.

The following sections describe how to configure AdvFS and UFS to use an LSM volume.

4.5.1 Configuring AdvFS Domains to Use LSM Volumes

AdvFS treats LSM volumes as it does any other storage device. (For more information on creating an AdvFS domain, see the *AdvFS Administration* manual.)

Note

If an AdvFS domain needs more storage, you can either create a new LSM volume and add it to the domain using the AdvFS `addvol` command or grow an LSM volume that is already part of the domain (Section 5.4.9).

AdvFS domains can use a combination of LSM volumes and physical storage. For more information, see the *AdvFS Administration* manual.

4.5.2 Configuring UFS File Systems to Use LSM Volumes

To configure a UFS file system to use an LSM volume:

1. Create a file system using the LSM disk group and volume name:

```
# newfs [options] /dev/rvol/disk_group/volume
```

The following example creates a UFS file system on an LSM volume named `vol_ufs` in the `dg1` disk group:

```
# newfs /dev/rvol/dg1/vol_ufs
```

You do not need to specify the name of the disk group for LSM volumes in the `rootdg` disk group.

For more information, see `newfs(8)`.

2. Use the LSM block special device name to mount the file system:

```
# mount /dev/vol/disk_group/volume /mount_point
```

For example:

```
# mount /dev/vol/dg1/vol_ufs /mnt2
```

4.6 Creating LSM Volumes for Existing Data

You can use LSM to manage existing data by encapsulating the disk or disk partition containing the data. LSM converts the disk or partition to an LSM `nopriv` disk and creates an LSM volume from the encapsulated disk or disk partition.

You can encapsulate:

- Disks or disk partitions, including UFS file systems (Section 4.6.1)
- AdvFS domains (Section 4.6.2)
- The boot disk and primary swap space on a standalone system (Section 3.4.1)
- One or more cluster members' swap devices (Section 3.4.3)

4.6.1 Encapsulating Disks or Disk Partitions

Encapsulating existing data is a two-part process: you create the encapsulation scripts with the `volencap` command, then run the `volreconfig` command to execute those scripts.

The encapsulation procedure configures the named disks and disk partitions as LSM `nopriv` disks, using information in the disk label and in the `/etc/fstab` file, and creates a separate LSM volume from each `nopriv` disk. By default, the `nopriv` disk and volume will belong to the `rootdg` disk group, unless you specify a different disk group.

If you encapsulate an entire disk (by not specifying a partition letter), such as `dsk3`, LSM creates a `nopriv` disk and a volume for each in-use partition.

Encapsulation provides a way to put existing data under LSM control, which lets you use LSM to mirror the data in the volume and thereby provide redundancy and high availability. However, recovery of a failed LSM `nopriv` disk can be complex, and there are other cases where `nopriv` disks complicate matters. After you encapsulate a domain, immediately move the volumes to LSM `sliced` or `simple` disks in the disk group if possible, before mirroring or performing other operations on the volumes.

After the encapsulation, entries in the `/etc/fstab` file or in the `/etc/sysconfigtab` file are changed to use the LSM volume name instead of the block device name of the disk or disk partition.

To encapsulate a disk or disk partition:

1. Back up the data on the disk or disk partition to be encapsulated.
2. Unmount the disk or partition or take the data off line. If you cannot unmount the disk or partition or take the data off line, LSM must restart the system to complete the encapsulation procedure.
3. Create the LSM encapsulation script:

```
# volencap [-g disk_group] {disk|partition}
```

The following example creates an encapsulation script for a disk named `dsk3`:

```
# volencap dsk3
```

Note

Although you can encapsulate several disks or disk partitions at the same time, we recommend that you encapsulate each disk or disk partition separately.

4. Complete the encapsulation process:

```
# volreconfig
```

If the encapsulated disk or disk partition is in use, the `volreconfig` command queries you whether to restart the system now or later.

5. Optionally (but recommended), move the volume to `sliced` or `simple` disks in the same disk group before mirroring the volume or performing other operations. See Section 5.1.5.

4.6.2 Creating LSM Volumes for AdvFS Domains

You can place the storage for an existing AdvFS domain under LSM control by either encapsulating the domain or migrating the domain to an LSM volume.

- **Encapsulating an AdvFS domain**, or more precisely, encapsulating the storage in use by the domain, creates an LSM volume for each disk or partition in the domain. If you cannot unmount the filesets before performing the encapsulation, LSM must restart the system to complete the process.

Note

If an AdvFS domain consists of one disk or partition, you can encapsulate the disk or partition (Section 4.6.1).

When you encapsulate an AdvFS domain, LSM changes the links in the `/etc/fdmns` directory to point to the LSM volumes.

- **Migrating an AdvFS domain** creates an LSM volume on disks that you specify, moves the domain data to the new volume, and removes the original disks from the domain. The disks are no longer in use by the domain after the migration completes.

Migrating a domain does not require you to unmount filesets or restart the system, but it temporarily uses additional disk space until the migration is complete.

No mount point changes are necessary during encapsulation or migration, because the mounted filesets are abstractions to the domain. The domain can be activated normally after the encapsulation or migration process completes. After the domain is activated, the filesets remain unchanged and the result of encapsulation or migration is transparent to AdvFS domain users.

4.6.2.1 Encapsulating an AdvFS Domain

Encapsulation provides a way to put existing data under LSM control, which lets you use LSM to mirror the data in the volume to provide redundancy

and high availability. However, encapsulation creates a `nopriv` disk, and recovery of a failed `nopriv` disk can be complex, and there are other cases where `nopriv` disks complicate matters. After you encapsulate a domain, immediately move the volumes to LSM `sliced` or `simple` disks in the disk group if possible, before mirroring or performing other operations on the volumes.

To encapsulate an AdvFS domain:

1. Back up the data in the AdvFS domain with the `vdump` utility.
2. Unmount all filesets.

If the domain is in use (you cannot unmount the filesets), you can create the encapsulation script (step 3) and run `volreconfig` (step 4) when convenient to complete the encapsulation procedure.

3. Create the LSM encapsulation script:

```
# volencap domain
```

The following example creates an encapsulation script for an AdvFS domain named `dom1`:

```
# volencap dom1
```

4. Complete the encapsulation procedure:

```
# volreconfig
```

If the AdvFS domain is mounted, the `volreconfig` command prompts you to restart the system.

The `/etc/fdmns` directory is updated on successful creation of LSM volumes.

5. Optionally (but recommended), move the volume to `sliced` or `simple` disks in the same disk group before mirroring the volume or performing other operations. See Section 5.1.5.

4.6.2.2 Migrating an AdvFS Domain

The `volmigrate` command lets you migrate any AdvFS domain (except for the `root_domain` on a standalone system) to an LSM volume. This operation uses a different disk than the disk the domain uses originally and therefore does not require a restart.

The `volmigrate` command creates a volume with the properties you specify, including:

- The disk group in which to create the volume (based on the disks you specify).
- The name of the volume. (The default is the name of the domain with the suffix `vol`.)

- The number of stripe columns and mirrors in the volume.
Striping improves read performance, and mirroring ensures data availability in the event of a disk failure.

There must be sufficient LSM disks in the same disk group, and the disks must be large enough to contain the domain. For more information on disk requirements and the options for striping and mirroring, see `volmigrate(8)`.

To migrate a domain to an LSM volume:

```
# volmigrate [-g disk_group] [-m num_mirrors] [-s num_stripes] \  
domain disk_media_name...
```

The `volmigrate` command creates a volume with the specified characteristics, moves the data from the domain to the volume, removes the original disk or disks from the domain, and leaves those disks unused. The volume is started and ready for use, and no restart is required.

Managing LSM Objects

This chapter describes how to manage LSM objects using LSM commands. For more information on an LSM command, see the reference page corresponding to its name. For example, for more information on the `volassist` command, enter:

```
# man volassist
```

5.1 Managing LSM Disks

The following sections describe how to use LSM commands to manage LSM disks.

5.1.1 Displaying LSM Disk Information

To display detailed information for an LSM disk:

```
# voldisk list disk
```

The following example contains information for an LSM disk named `dsk12`:

```
Device:      dsk12
devicetag:   dsk12
type:        sliced
hostid:      hostname.com
disk:        name=dsk12 id=1012859934.2400.potamus.zk3.dec.com
group:       name=dg2 id=1012859945.2405.potamus.zk3.dec.com
flags:       online ready autoimport imported
pubpaths:    block=/dev/disk/dsk12g char=/dev/rdisk/dsk12g
privpaths:   block=/dev/disk/dsk12h char=/dev/rdisk/dsk12h
version:     2.1
iosize:      min=512 (bytes) max=2048 (blocks)
public:      slice=6 offset=16 len=2046748
private:     slice=7 offset=0 len=4096
update:      time=1012859947 seqno=0.1
headers:     0 248
configs:     count=1 len=2993
logs:        count=1 len=453
Defined regions:
config  priv   17-   247[   231]: copy=01 offset=000000 enabled
config  priv   249-  3010[  2762]: copy=01 offset=000231 enabled
log     priv   3011-  3463[   453]: copy=01 offset=000000 enabled
```

5.1.2 Renaming LSM Disks

When you initialize an LSM disk, you can assign it a disk media name or use the default disk media name, which is the same as the disk access name.

Caution

Each disk in a disk group must have a unique name. To avoid confusion, you might want to ensure that no two disk groups contain disks with the same name. For example, both the `rootdg` disk group and another disk group might contain disks with a disk media name of `disk03`. Because most LSM commands operate on the `rootdg` disk group unless you specify otherwise, you can inadvertently perform operations on the wrong disk if multiple disk groups contain identically named disks.

The `voldisk list` command displays a list of all the LSM disks in all disk groups on the system.

To rename an LSM disk:

```
# voledit rename old_disk_media_name new_disk_media_name
```

For example, to rename an LSM disk from `disk03` to `disk01`:

```
# voledit rename disk03 disk01
```

5.1.3 Placing LSM Disks Off Line

You can place an LSM disk off line to:

- Prevent LSM from accessing it
- Enable you to move the disk to a different physical location and have the disk retain its LSM identity

Placing a disk off line closes its device file. If a disk is in use, you cannot place it off line.

To place an LSM disk off line:

1. Remove the LSM disk from its disk group:

```
# voldg [-g disk_group] rmdisk disk
```

2. Place the LSM disk off line:

```
# voldisk offline disk
```

5.1.4 Placing LSM Disks On Line

To restore access to an LSM disk that you placed off line, place it on line. The LSM disk is placed in the free disk pool and is accessible to LSM again. After placing an LSM disk on line, you must add it to a disk group before an LSM volume can use it. If the disk belonged to a disk group previously, you can add it to the same disk group.

To place an LSM disk on line:

```
# voldisk online disk
```

For information on adding an LSM disk to a disk group, see Section 5.2.2.

5.1.5 Moving Data Off an LSM Disk

You can move (evacuate) LSM volume data to other LSM disks in the same disk group if there is sufficient free space. If you do not specify a target LSM disk, LSM uses any available LSM disk in the disk group that has sufficient free space.

You might want to move data off an LSM disk in the following circumstances:

- To move data off a `nopriv` disk (created when you encapsulate a disk, disk partition, or AdvFS domain) to `sliced` or `simple` disks in the disk group.

This is recommended, because `nopriv` disks are more complex to manage and are intended to be used only temporarily, as a way to place existing data under LSM control. The exception: `nopriv` disks created by encapsulating the boot disk partitions on a standalone system.

For information on verifying there is sufficient free space in the disk group to support the move, see Section 5.2.1.

- To move LSM objects relocated by a hot-sparing operation. See Section 5.1.6 for specific information.
- To redistribute one or more volumes that use space on the same disk, if this is causing contention and slow performance. You can determine this by using the `volstat` command (Section 6.1.2).

To move data off a disk, use one of the following commands:

- The `volassist move` command, which moves a specific volume off a disk, but leaves objects from other volumes on the disk.
- The `volevac` command, which moves all LSM objects off a particular disk (for example, multiple LSM volumes or logs).

Note

Do not move the contents of an LSM disk to another LSM disk that contains data from the same volume. The resulting layout might not preserve redundancy for volumes that use mirror plexes or a RAID 5 plex.

To move all data (for example, multiple LSM volumes) off an LSM disk:

```
# volevac [-g disk_group] source_disk target_disk [target_disk...]
```

For example:

- To move all the data from LSM disk `dsk8` to `dsk9` (in the `rootdg` disk group):

```
# volevac dsk8 dsk9
```

- To move data from a `nopriv` disk to any available sliced or simple disk:

```
# volevac dsk24c [disk...]
```

If you specify the target disks, make sure they have sufficient free space.

To move a volume off an LSM disk:

```
# volassist [-g disk_group] move volume [\!]source_disk \  
[target_disk...]
```

For example, suppose three volumes in `rootdg` use space on `dsk1`:

```
# volprint -s | grep dsk1
```

```
sd dsk1-04    rootvol-04  ENABLED  65      FPA      -      -      -  
sd dsk1-01    vol_1-03    ENABLED  65      LOG      -      -      -  
sd dsk1-02    vol_2-03    ENABLED  65      LOG      -      -      -  
:  
:
```

The following commands move volumes `vol_1` and `vol_2` off `dsk1` (using the quoting convention for the C shell to correctly interpret the `!` symbol):

```
# volassist move vol_1 \!dsk1  
# volassist move vol_2 \!dsk1
```

The following command confirms that the only volume using `dsk1` after the operation is `rootvol`:

```
# volprint | grep dsk1-
```

```
sd dsk1-04    rootvol-04  ENABLED  65      FPA      -      -      -
```

The following command displays the disks now used in volumes `vol_1` and `vol_2`:

```
# volprint vol_1 vol_2
```

Disk group: `rootdg`

TY	NAME	ASSOC	KSTATE	LENGTH	PLOFFS	STATE	TUTIL0	PUTIL0
v	vol_1	fsgen	ENABLED	262144	-	ACTIVE	-	-
pl	vol_1-01	vol_1	ENABLED	262144	-	ACTIVE	-	-
sd	dsk0-01	vol_1-01	ENABLED	262144	0	-	-	-
pl	vol_1-03	vol_1	ENABLED	LOGONLY	-	ACTIVE	-	-
sd	dsk4-01	vol_1-03	DETACHED	65	LOG	RECOVER	-	-
v	vol_2	fsgen	ENABLED	262144	-	ACTIVE	-	-
pl	vol_2-01	vol_2	ENABLED	262144	-	ACTIVE	-	-
sd	dsk3-01	vol_2-01	ENABLED	262144	0	-	-	-
pl	vol_2-03	vol_2	ENABLED	LOGONLY	-	ACTIVE	-	-
sd	dsk6-01	vol_2-03	ENABLED	65	LOG	-	-	-

5.1.6 Moving LSM Objects Relocated by Hot-Sparing

When LSM objects are moved from a failing disk to a hot-spare disk by the hot-sparing feature, their new locations might not provide the same performance or have the same data layout that existed before. After hot-sparing occurs, you might want to move the relocated LSM objects to a different disk to improve performance, to keep the hot-spare disk space free for future hot-sparing needs, or to restore the LSM configuration to its previous state.

Note

The following procedure assumes you have initialized a new disk to replace the hot-spare disk. For more information on adding disks for LSM use, see Section 4.1. For more information on replacing a failed disk, see Section 6.4.5.

In the following procedure, the hot-spare disk to which LSM moved the data is called the relocation disk. The disk you choose to move the data to is called the target disk. Use the disk media name of both the relocation disk and the target disk.

To move LSM objects that were relocated by hot-sparing:

1. Note the characteristics of the LSM objects before they were relocated.

This information is available from the mail notification about the failure that the `volwatch` daemon sent to the root account. For example, look for a message similar to the following:

```
To: root
Subject: Logical Storage Manager failures on host teal

Attempting to relocate subdisk disk02-03 from plex home-02.
Dev_offset 0 length 1164 dm_name disk02 da_name dsk2.
The available plex home-01 will be used to recover the data.
```

2. Note the new location of the relocated LSM object (the relocation disk).

This information is available from the mail notification about the failure that the `volwatch` daemon sent to the root account. For example, look for a message similar to the following:

```
To: root
Subject: Attempting LSM relocation on host teal

Volume home Subdisk disk02-03 relocated to disk05-01,
but not yet recovered.
```

3. Find a suitable target disk in the same disk group, and ensure that the target disk is not already in use by the same volume.

For example, as shown in the sample mail message in the previous step, do not attempt to relocate the data to a disk in use by the volume home. For more information on finding unused space in a disk group, see Section 5.2.1.

4. Move the objects from the relocation disk to the target disk:

```
# volevac [-g disk_group] dm_relocation_disk dm_target_disk
```

5.1.7 Removing Disks from LSM Control

You can remove a disk from LSM control if you no longer need it.

For information on removing an LSM disk from a disk group, see Section 5.2.3. For information on deporting a disk group, see Section 5.2.4.

To remove an LSM disk from LSM control:

1. Display the disk media name and disk access name of the disk:

```
# voldisk list
```

Information similar to the following is displayed (edited for brevity):

DEVICE	TYPE	DISK	GROUP	STATUS
:				
dsk25	sliced	-	-	unknown
dsk26	sliced	newdisk	rootdg	online
dsk27	sliced	dsk27	rootdg	online

2. Remove the disk media name from the disk group:

```
# voldg [-g disk_group] rmdisk disk
```

For example, to remove dsk26, which has a disk media name of newdisk, from the rootdg disk group:

```
# voldg rmdisk newdisk
```

When you remove an LSM disk from a disk group, it no longer has a disk media name. The disk keeps its disk access name until you remove the disk from LSM control. The `voldisk list` command shows that dsk26 does not belong to any disk group, and it has a status of online, indicating it is under LSM control:

DEVICE	TYPE	DISK	GROUP	STATUS
:				
dsk25	sliced	-	-	unknown
dsk26	sliced	-	-	online
dsk27	sliced	dsk27	rootdg	online

3. Remove the disk access name from LSM control:

```
# voldisk rm disk
```

For example, to remove disk dsk26:


```
# voldisk rm dsk26
```

All the disk partition labels are changed to unused. To use the disk for other purposes, reinitialize it using the `disklabel` command. For more information, see `disklabel(8)`.

5.2 Managing Disk Groups

The following sections describe how to use LSM commands to manage disk groups.

5.2.1 Displaying Disk Group Information

- To display a list of all LSM disks and the disk group to which each belongs, and disks that LSM recognizes but which are not under LSM control:

```
# voldisk list
```

DEVICE	TYPE	DISK	GROUP	STATUS
dsk0	sliced	-	-	unknown
dsk1	sliced	-	-	unknown
dsk2	sliced	dsk2	rootdg	online
dsk3	sliced	dsk3	rootdg	online
dsk4	sliced	dsk4	rootdg	online
dsk5	sliced	dsk5	rootdg	online
dsk6	sliced	dsk6	dg1	online
dsk7	sliced	dsk7	dg1	online
dsk8	sliced	dsk8	dg1	online
dsk9	sliced	dsk9	dg2	online
dsk10	sliced	dsk10	dg2	online
dsk11	sliced	dsk11	dg2	online
dsk12	sliced	-	-	unknown
dsk13	sliced	-	-	unknown

- To display the free space in one or all disk groups:

```
# voldg [-g disk_group] free
```

GROUP	DISK	DEVICE	TAG	OFFSET	LENGTH	FLAGS
rootdg	dsk2	dsk2	dsk2	2097217	2009151	-
rootdg	dsk3	dsk3	dsk3	2097152	2009216	-
rootdg	dsk4	dsk4	dsk4	0	4106368	-
rootdg	dsk5	dsk5	dsk5	0	4106368	-
dg1	dsk6	dsk6	dsk6	0	2046748	-
dg1	dsk8	dsk8	dsk8	0	2046748	-

The value in the `LENGTH` column indicates the amount of free disk space in 512-byte blocks. (2048 blocks equal 1 MB.)

- To display the the largest volume you can create in the disk group, use the `volassist maxsize` command.

The value returned will vary depending on the properties of the volume you want to create, such as striped or mirrored.

For example, to display the largest volume you can create that is striped over four disks, with a stripe width of 512K bytes:

```
# volassist [-g disk_group] maxsize stwidth=512k ncolumn=4
Maximum volume size: 160051200 (78150Mb)
```

You can then create a volume with those properties, of any size up through that value; for example:

```
# volassist -g dg1 make megavol 78150m stwidth=512k ncolumn=4
```

5.2.2 Adding LSM Disks to Disk Groups

You can add unassigned LSM disks to any disk group. To display a list of unassigned disks, enter the `voldisk list` command. Unassigned LSM disks are those with a status of `online` and dashes (-) in the `DISK` and `GROUP` columns of the output.

To add one or more LSM disks to an existing disk group:

```
# voldg -g disk_group adddisk disk [disk...]
```

For example, to add the disk `dsk10` to the disk group `dg1`:

```
# voldg -g dg1 adddisk dsk10
```

To initialize a disk for LSM use and either add it to a disk group or use it to initialize a new disk group in one step, use the `voldiskadd` script (Section 4.2.1).

5.2.3 Removing LSM Disks from Disk Groups

You can remove an LSM disk from a disk group; however, you cannot remove:

- The last disk in a disk group unless the disk group is deported. For information on deporting a disk group, see Section 5.2.4.
- Any disk that is in use (for example, disks that contain active LSM volume data). If you attempt to remove a disk that is in use, LSM displays an error message and does not remove the disk.

For information on moving data off an LSM disk, see Section 5.1.5. For information on removing LSM volumes, see Section 5.4.6.

To remove an LSM disk from a disk group:

1. Verify that the LSM disk is not in use by listing all subdisks:

```
# volprint -st
Disk group: rootdg

SD NAME          PLEX          DISK          DISKOFFS LENGTH  [COL/]OFF DEVICE  MODE
sd dsk1-01       klavol-01     dsk1          0        1408   0/0    dsk1    ENA
sd dsk2-02       klavol-03     dsk2          0         65    LOG    dsk2    ENA
sd dsk2-01       klavol-01     dsk2          65       1408   1/0    dsk2    ENA
sd dsk3-01       klavol-01     dsk3          0        1408   2/0    dsk3    ENA
sd dsk4-01       klavol-02     dsk4          0        1408   0/0    dsk4    ENA
```

```
sd dsk5-01      klavol-02    dsk5    0      1408    1/0      dsk5    ENA
sd dsk6-01      klavol-02    dsk6    0      1408    2/0      dsk6    ENA
```

The disks in the `DISK` column are currently in use by an LSM volume, and therefore you cannot remove those disks from a disk group.

2. Remove the LSM disk from the disk group:

```
# voldg -g disk_group rmdisk disk
```

For example, to remove the LSM disk `dsk8` from the `rootdg` disk group:

```
# voldg rmdisk dsk8
```

The disk remains under LSM control. You can:

- Add the LSM disk to a different disk group (Section 5.2.2).
- Use the disk to create a new disk group (Section 4.2).
- Remove the disk from LSM control (Section 5.1.7).

5.2.4 Deporting Disk Groups

Deporting a disk group makes its volumes inaccessible. You can deport a disk group to:

- Rename the disk group.
- Reuse the disks for other purposes.
- Move the disk group to another system (Section 7.3.2).

You cannot deport the `rootdg` disk group.

Caution

The `voldisk list` command displays the disks in a deported disk group as available (with a status of `online`). However, removing or reusing the disks in a deported disk group can result in data loss.

To deport a disk group:

1. If applicable, stop the volumes:

```
# volume -g disk_group stopall
```

2. Deport the disk group:

- To deport the disk group with no changes:

```
# voldg deport disk_group
```

- To deport the disk group and assign it a new name:

```
# voldg [-n newname] deport disk_group
```

For more information on assigning a new name to a disk group, see `voldg(8)`.

You must import a disk group (Section 5.2.5) before you can use it.

If you no longer need the disk group, you can:

- Add the disks to different disk groups (Section 5.2.2).
- Use the disks to create new disk groups (Section 4.2).
- Remove the disks from LSM control (Section 5.1.7).

5.2.5 Importing Disk Groups

Importing a disk group makes the disk group and its volumes accessible. You cannot import a disk group if you used any of its associated disks while it was deported.

To import a disk group and restart its volumes:

1. Import the disk group:

```
# voldg import disk_group
```

2. Start all volumes within the disk group:

```
# volume -g disk_group startall
```

5.2.6 Renaming Disk Groups

Renaming a disk group involves deporting and then importing the disk group. You cannot rename a disk group while it is in use. All activity on all volumes in the disk group must stop, and the volumes in the disk group are inaccessible while the disk group is deported.

Because renaming a disk group involves an interruption of service to the volumes, you should perform this task during a planned shutdown or maintenance period. Choose the new disk group name carefully, and ensure that the new name is easy to remember and use. Renaming a disk group updates the `/etc/fstab` file.

Note

You cannot rename the `rootdg` disk group.

To rename a disk group:

1. Deport the disk group, assigning it a new name (Section 5.2.4).
2. Import the disk group using its new name (Section 5.2.5).

5.2.7 Creating a Clone Disk Group

The `volclonedg` command lets you create a copy of a disk group using disks that are hardware clones. This command is available on both standalone systems and clusters.

Note

You must create hardware clones of all the disks in a disk group before cloning the disk group. The disk group cannot contain any `nopriv` disks.

The `volclonedg` command creates a new disk group containing the same configuration of LSM objects as the parent disk group, using the `volsave` and `volrestore` commands to save the configuration of the parent disk group and create the same configuration in the clone disk group. LSM starts all possible volumes in the clone disk group, performing recovery of mirrored volumes if necessary. For more information, see `volclonedg(8)`.

To clone a disk group:

1. Display the disks in the disk group to be cloned:

```
# voldisk -g disk_group list
```

DEVICE	TYPE	DISK	GROUP	STATUS
dsk10	sliced	dsk10	dg1	online
dsk11	sliced	dsk11	dg1	online
dsk12	sliced	dsk12	dg1	online
dsk13	sliced	dsk13	dg1	online

Note

If the disk group contains any `nopriv` disks, add new `sliced` or `simple` disks to the disk group, move the data from the `nopriv` disks to the new disks (using the `volevac` command, for example), and remove the `nopriv` disks from the disk group before you clone the disk group.

2. Create hardware clones of the disks. See your hardware documentation for more information.
3. Run the `hwmgr` command to update the system or cluster with the new disk information. For more information, see `hwmgr(8)`.
4. Verify that LSM can access and display the cloned disks:

```
# voldisk list
```

Cloned disks show a status of `online aliased`.

In the following output, the original disks are dsk10, dsk11, dsk12, and dsk13. The hardware disk clones are dsk14, dsk15, dsk16, and dsk17.

DEVICE	TYPE	DISK	GROUP	STATUS
dsk0	sliced	dsk0	rootdg	online
dsk1	sliced	dsk1	rootdg	online
dsk2	sliced	dsk2	rootdg	online spare
dsk3	sliced	dsk3	rootdg	online spare
dsk4	sliced	dsk4	rootdg	online
dsk5	sliced	dsk5	rootdg	online spare
dsk6	sliced	dsk6	rootdg	online
dsk7	sliced	dsk7	rootdg	online
dsk8	sliced	dsk8	rootdg	online
dsk9	sliced	dsk9	rootdg	online
dsk10	sliced	dsk10	dg1	online
dsk11	sliced	dsk11	dg1	online
dsk12	sliced	dsk12	dg1	online
dsk13	sliced	dsk13	dg1	online
dsk14	sliced	-	-	online aliased
dsk15	sliced	-	-	online aliased
dsk16	sliced	-	-	online aliased
dsk17	sliced	-	-	online aliased

- Use the names of the disk clones to clone the disk group dg1, optionally assigning a name other than the default (in this case, dg1_clone):

```
# volclonedg -g dg1 [-N name] dsk14 dsk15 dsk16 dsk17
```

LSM creates the clone disk group and starts its volumes.

For more information, including an example of creating a clone disk group on a different system from the parent disk group, see volclonedg(8).

5.3 Managing the LSM Configuration Database

This section describes how to manage the LSM configuration database, including:

- Backing up the configuration database (Section 5.3.1)
- Restoring the configuration database from backup (Section 5.3.2)
- Changing the size and number of configuration database copies (Section 5.3.3)

5.3.1 Backing Up the LSM Configuration Database

Use the volsave utility to periodically create a copy of the LSM configuration. You can then use the volrestore command to recreate the LSM configuration if you lose a disk group configuration.

The saved configuration database (also called a description set) is a record of the objects in the LSM configuration (the LSM disks, subdisks, plexes, and volumes) and the disk group to which each object belongs.

Whenever you make a change to the LSM configuration, the backup copy becomes obsolete. Like any backup, the content is useful only as long as it accurately represents the current information. Any time the number, nature, or name of LSM objects change, consider making a backup of the LSM configuration database.

The following list describes some of the changes that will invalidate a configuration database backup:

- Creating disk groups
- Adding or removing disks from disk groups or from LSM control
- Creating or removing volumes
- Changing the properties of volumes, such as the plex layout or number of logs

Note

Backing up the configuration database does not save the data in the volumes. For information on backing up volume data, see Section 5.4.2.

The `volsave` command does not save information relating to volumes used for the root, `/usr`, or `/var` file systems or for swap space.

Depending on the nature of a boot disk failure, you might need to restore the system partitions from backups or installation media to return to a state where the system partitions are not under LSM control. From there, you can redo the procedures to encapsulate the boot disk partitions to LSM volumes and add mirror plexes to those volumes.

For more information about recovering from a boot disk failure under LSM control, see Section 6.4.6.

By default, LSM saves the entire configuration database to a time-stamped directory called `/usr/var/lsm/db/LSM.date.hostname`. You can specify a different directory for the backup, but the directory must not already exist.

The backup directory contains the following files and directories:

- A copy of the `volboot` file.
- A file called `header`, which contains host ID and checksum information, and a list of the other files in this directory.

- A file called `voldisk.list`, which contains a list of all LSM disks, their type (`sliced`, `simple`, `nopriv`), the size of their private and public regions, their disk group, and other information.
- A subdirectory called `rootdg.d`, which contains the `allvol.DF` file. The `allvol.DF` file contains detailed descriptions of every LSM subdisk, plex, and volume, describing all their properties and attributes.

To back up the LSM configuration database:

1. Enter the following command, optionally specifying a directory location other than the default to store the LSM configuration database:

```
# volsave [-d directory]
```

2. Save the backup to tape or other removable media.

The `volsave` command saves multiple versions of the configuration database; each new backup is saved in the `/usr/var/lsm/db` directory with its own date and time stamp, as shown in the following example:

```
dr-xr-x---  3 root      system      8192 May  5 09:36 LSM.20000505093612.hostname
dr-xr-x---  3 root      system      8192 May 10 10:53 LSM.20000510105256.hostname
```

5.3.2 Restoring the LSM Configuration Database from Backup

The `volrestore` command restores an LSM configuration database, provided you saved it with the `volsave` command (Section 5.3.1).

You can restore the configuration database of a specific disk group or volume or the entire configuration (all disk groups and volumes except those associated with the boot disk). If you have multiple backups of the configuration database (a new one is created each time you run the `volsave` command), you can choose a specific one to restore. Otherwise, LSM restores the most recent version.

Note

Restoring the configuration database does not restore data in the LSM volumes. For information on restoring volumes, see Section 5.4.3.

The `volrestore` command does not restore volumes associated with the root (`/`), `/usr`, and `/var` file systems and the primary swap area on a standalone system. If volumes for these partitions are corrupted or destroyed, you must reencapsulate the system partitions to use LSM volumes.

To restore a backed-up LSM configuration database:

1. Optionally, display a list of all available database backups:

```
# ls /usr/var/lsm/db
```

If you saved the configuration database to a different directory, specify that directory.

2. Restore the chosen configuration database:

- To restore the entire configuration database:

```
# volrestore [-d directory]
```

- To restore a specific disk group configuration database:

```
# volrestore [-d directory] -g disk_group
```

- To restore a specific volume configuration:

```
# volrestore [-d directory] -v volume
```

- To restore a configuration database interactively, enabling you to select or skip specific objects:

```
# volrestore [-d directory] -i
```

3. Start the restored LSM volumes:

```
# volume -g disk_group startall
```

If the volumes will not start, you might need to manually edit the plex state. See Section 6.5.2.2.

4. If necessary, restore the volume data from backup. For more information, see Section 5.4.3.

5.3.3 Changing the Size and Number of Configuration Database Copies

LSM maintains copies of the configuration database on separate physical disks within each disk group. When the disk group runs out of space in the configuration database, LSM displays the following message:

```
volmake: No more space in disk group configuration
```

This can happen if:

- One or more disks in the disk group contain two copies of the configuration database. Whenever a configuration change occurs, all active copies are updated. If one disk's copies cannot be updated because they have grown too large, then none of the copies for the whole disk group can be updated.
- The disk group came from a system running Tru64 UNIX Version 4.0, which uses a smaller default private region size.

If the configuration database runs out of space and you find that some disks have two copies of the configuration database, you can remove one copy from each disk that has two (Section 5.3.3.1). However, make sure that the disk group still has sufficient copies of the configuration database available for redundancy. For example, if the disk group has a total of four copies and two are on the same disk, remove one copy from that disk and enable a copy on another disk that does not have one.

If all copies of the configuration database are the same size and no disk has more than one copy, this might indicate that the private regions of the disks are too small. For example, the disks were initialized on a system running an earlier version of LSM, with a smaller default private region. To resolve this problem, add new disks to LSM, which will have the larger default private region size, add the new disks to the disk group, and delete the copies of the configuration database on the other disks (Section 5.3.3.2).

5.3.3.1 Reducing the Number of Configuration Database Copies on an LSM Disk

To reduce the number of configuration database copies on an LSM disk:

1. Display information about the disk group's configuration database:

```
# voldg list disk_group

Group:   rootdg
dgid:    783105689.1025.lsm
import-id: 0.1
flags:
config:  seqno=0.1112 permlen=173 free=166 templen=6 loglen=26
config disk dsk13 copy 1 len=173 state=clean online
config disk dsk13 copy 2 len=173 state=clean online
config disk dsk11g copy 1 len=347 state=clean online
config disk dsk10g copy 1 len=347 state=clean online
log disk dsk11g copy 1 len=52
log disk dsk13 copy 1 len=26
log disk dsk13 copy 2 len=26
log disk dsk10g copy 1 len=52
```

2. Identify a disk that has multiple copies of the configuration database.
For example, disk `dsk13` has two copies of the configuration database. This halves the total configuration space available in memory for the disk group and is therefore the limiting factor.
3. Reduce the number of copies on a disk that has two copies:

```
# voldisk modddb disk nconfig=n
```

For example, to reduce the number of configuration copies on `dsk13` from two to one:

```
# voldisk modddb dsk13 nconfig=1
```

4. Display the new configuration:

```
# voldg list disk_group
```

5. If necessary, add a copy to another disk to maintain the appropriate number of copies for the disk group:

- a. Display a list of all disks in the disk group:

```
# voldisk -g disk_group list
```

- b. Compare the disks listed in the output of the `voldisk list` command to those listed in the output of the `voldg list` command to identify a disk in the disk group that does not have a copy of the configuration database.
- c. Enable a configuration database copy on a disk that does not have one, using the disk access name:

```
# voldisk moddb disk_access_name nconfig=1
```

5.3.3.2 Removing Configuration Database Copies on LSM Disks with Small Private Regions

To remove configuration database copies on an LSM disk with a small private region:

1. Display information about the disk group's configuration database:

```
# voldg list disk_group

Group:      rootdg
dgid:      921610896.1026.hostname
import-id: 0.1  flags:
copies:    nconfig=default nlog=default
config:    seqno=0.1081 permlen=347 free=341 templen=3 loglen=52
config disk dsk7 copy 1 len=347 state=clean online
config disk dsk8 copy 1 len=2993 state=clean online
config disk dsk9 copy 1 len=2993 state=clean online
config disk dsk10 copy 1 len=2993 state=clean online
log disk dsk7 copy 1 len=52
log disk dsk8 copy 1 len=453
log disk dsk9 copy 1 len=453
log disk dsk10 copy 1 len=453
```

Disk `dsk7` has a smaller private region than the other disks, as shown by the `len=` information in the lines beginning with `config disk` and `log disk`, and therefore has less space to store copies of the configuration database and log. This restricts the disk group's ability to store additional records, because the smallest private region sets the limit for the group.

2. Remove all configuration database copies from the disk with the smallest private region:

```
# voldisk moddb disk nconfig=0
```

For example, to remove the copies on the disk with the smallest private region, `dsk7`:

```
# voldisk moddb dsk7 nconfig=0
```

3. Display the new configuration:

```
# voldg list disk_group
```

4. If necessary, add a copy to another disk to maintain the appropriate number of copies for the disk group:
 - a. Display a list of all disks in the disk group:

```
# voldisk -g disk_group list
```

- b. Compare the disks listed in the output of the `voldisk list` command to those listed in the output of the `voldg list` command to identify a disk in the disk group that does not have a copy of the configuration database.

- c. Enable a copy on a disk that does not have one, using the disk access name:

```
# voldisk moddb disk_access_name nconfig=1
```

5.4 Managing LSM Volumes

The following sections describe how to use LSM commands to manage LSM volumes. For information on creating LSM volumes, see Chapter 4.

5.4.1 Displaying LSM Volume Information

The `volprint` command displays information about LSM objects, including LSM disks, subdisks, plexes, and volumes.

To display the complete hierarchy of objects in an LSM volume:

```
# volprint [-g disk_group] -ht volume
```

```
Disk group: rootdg 1
```

V NAME	USETYPE	KSTATE	STATE	LENGTH	READPOL	PREFPLEX		
PL NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WID	MODE	
SD NAME	PLEX	DISK	DISKOFFS	LENGTH	[COL/]OFF	DEVICE	MODE	
v data01	fsgen	ENABLED	ACTIVE	512000	SELECT	-	2	
pl data01-01	data01	ENABLED	ACTIVE	512256	STRIPE	3/128	RW	3
sd dsk2-01	data01-01	dsk2	0	170752	0/0	dsk2	ENA	4
sd dsk5-01	data01-01	dsk5	0	170752	1/0	dsk5	ENA	
sd dsk6-01	data01-01	dsk6	0	170752	2/0	dsk6	ENA	
pl data01-02	data01	ENABLED	ACTIVE	512256	STRIPE	3/128	RW	
sd dsk7-01	data01-02	dsk7	0	170752	0/0	dsk7	ENA	
sd dsk8-01	data01-02	dsk8	65	170752	1/0	dsk8	ENA	
sd dsk9-01	data01-02	dsk9	0	170752	2/0	dsk9	ENA	
pl data01-03	data01	ENABLED	ACTIVE	LOGONLY	CONCAT	-	RW	
sd dsk8-02	data01-03	dsk8	0	65	LOG	dsk8	ENA	

The preceding example shows output for a volume with mirrored, three-column, striped plexes:

1 Disk group name.

- 2 Volume information: Name (data01), usage type (fsgen), state (ENABLED ACTIVE), and size (51200 blocks, or 250 MB).
- 3 Plex information: Two data plexes (data01-01 and data01-02) and a DRL plex (data01-03).
- 4 Subdisk information for each plex.

To display a listing of LSM volumes:

```
# volprint [-g disk_group] -vt
Disk group: rootdg
V  NAME           USETYPE    KSTATE    STATE    LENGTH    READPOL    PREFPLEX
v  rootvol        root       ENABLED   ACTIVE   524288    ROUND     -
v  swapvol        swap       ENABLED   ACTIVE   520192    ROUND     -
v  vol-dsk24c     fsgen      ENABLED   ACTIVE   17773524  SELECT    -
v  vol-dsk25g     fsgen      ENABLED   ACTIVE   2296428   SELECT    -
v  vol-dsk25h     fsgen      ENABLED   ACTIVE   765476    SELECT    -
v  vol-01         fsgen      ENABLED   ACTIVE   768000    ROUND     -
v  vstripe        fsgen      ENABLED   ACTIVE   256000    SELECT    vstripe-01
```

5.4.2 Backing Up LSM Volumes

One of the more common tasks of a system administrator is helping users recover lost or corrupted files. To perform that task effectively, establish procedures for backing up LSM volumes and the LSM configuration database at frequent and regular intervals. You will need the saved configuration database as well as the backed-up data if you need to restore a volume after a major failure. (For example, if multiple disks in the same volume failed, or all the disks containing the active configuration records for a disk group failed.)

For information on backing up the LSM configuration database, see Section 5.3.1.

For a thorough discussion of the backup and restore options available on a Tru64 UNIX system, see the *System Administration* manual. If you are using AdvFS, also see the *AdvFS Administration* manual.

LSM commands do not actually create the backup of the volume data, but they provide several ways for you to make volume data available to back up. In many cases, you do not have to unmount file systems or bring the system to single-user mode, which prevents possible corruption of the backup data if users write to a file system before the backup is complete.

To create a backup of LSM volume data, you can use the following commands:

- `volassist snapfast` and `volassist snapback`

These commands use the Fast Plex Attach feature to create a secondary volume from an existing eligible plex in a mirrored volume. They create a

log that keeps track of changes to the original (primary) volume from the time a plex is detached with `snapfast` to the time the plex is reattached to the original volume with `snapback`.

Advantages — These commands save time by using an existing eligible plex and reduce the time required for the returning plex to resynchronize to the original volume.

Disadvantages — These commands remove one plex to create the temporary volume; they might leave the primary volume unmirrored until the `snapback` operation occurs.

For a detailed description of the Fast Plex Attach feature and information on how to use it, see Section 5.4.2.1 and Section 5.4.2.2.

- `volassist snapstart` and `volassist snapshot`

These commands attach a new plex to a volume for the express purpose of creating a temporary volume for backups.

The `snapstart` keyword uses available space in the disk group to create and attach a new plex to the volume and performs a full synchronization of the new plex to the volume's contents.

The `snapshot` keyword uses the new plex to create a temporary volume with the name you specify. The temporary volume is the one you use to perform your backups; this leaves the original volume running and available for use.

After performing the backup, you can optionally detach the plex from the temporary volume and reattach it to the original volume. This performs a full resynchronization.

Advantages — These commands do not reduce the number of mirrors in a volume; they use a new plex created specifically for the backup. You can use this method to mirror a nonmirrored volume (except RAID 5), if sufficient disk space is available.

Disadvantages — These commands use additional disk space (equal to the size of the volume), and take time for the new plex to fully synchronize to the original volume and (optionally) for the reattached plex to resynchronize to the original volume.

For more information, see Section 5.4.2.3.

- `volplex det`, `volmake`, `volume start`, `voledit`, and `volplex att` (low-level commands)

These commands use an existing eligible plex from a mirrored volume to create the backup volume.

Advantages — These commands give you control over which plex to use and save time by using an existing eligible plex.

Disadvantages — These commands are considered low-level commands, introducing more chance of error. They remove one plex to create the temporary volume, which might leave the original volume unmirrored. If you reattach the plex to the original volume, it takes time for the plex to resynchronize.

For more information, see Section 5.4.2.4.

To back up a RAID 5 volume or a volume that use a single concatenated or striped plex that you cannot or choose not to mirror, see Section 5.4.2.5.

5.4.2.1 Overview of the Fast Plex Attach Feature

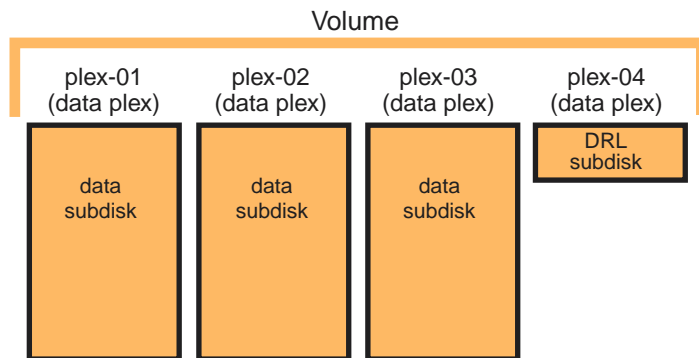
For mirrored LSM volumes, you can use the Fast Plex Attach feature to make a temporary copy of the volume data available for backup. You use the temporary volume to perform your backups, leaving the original volume running and available for use.

You can use the Fast Plex Attach feature on any mirrored volume on a standalone system or a cluster, including the `rootvol`, `cluster_rootvol`, and other volumes for encapsulated standalone system partitions and clusterwide file system domains. The Fast Plex Attach feature cannot be used on mirrored volumes used as swap space.

The `volassist` command provides two keywords (`snapfast` and `snapback`) to create and remove a backup volume using the Fast Plex Attach feature.

Figure 5–1 shows a three-way mirrored volume before a Fast Plex Attach operation.

Figure 5–1: Volume Before Fast Plex Attach



ZK-1875U-AI

A complete Fast Plex Attach operation goes through the following phases:

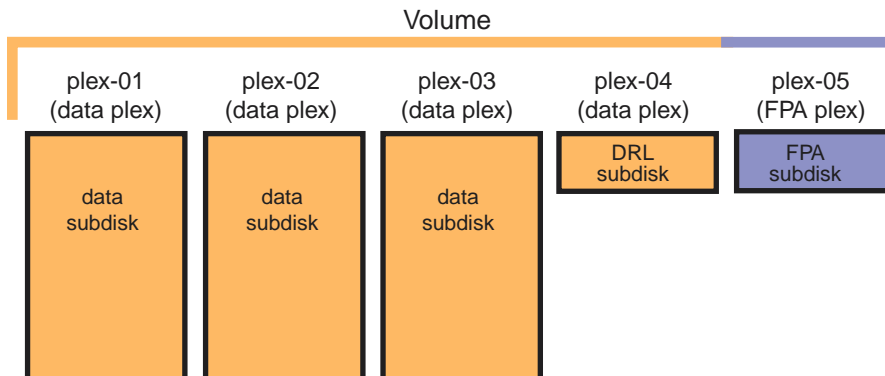
1. When you run the `volassist snapfast` command, LSM examines the volume (known as the primary volume) to ensure it has at least

two complete, read-write plexes and arbitrarily selects one plex as a candidate for Fast Plex Attach (FPA) support. This plex becomes the migrant plex. If any plex has a state of SNAPDONE (the result of a prior snapstart operation), LSM uses it as the migrant plex.

If the volume has only one plex, the command fails. If the volume has only two plexes, the command exits and displays a message stating that you must use the force (-f) option, because the operation will leave the volume with only one plex and therefore unmirrored.

LSM creates an FPA subdisk and attaches it to a separate FPA plex (plex-05) for the primary volume (Figure 5–2).

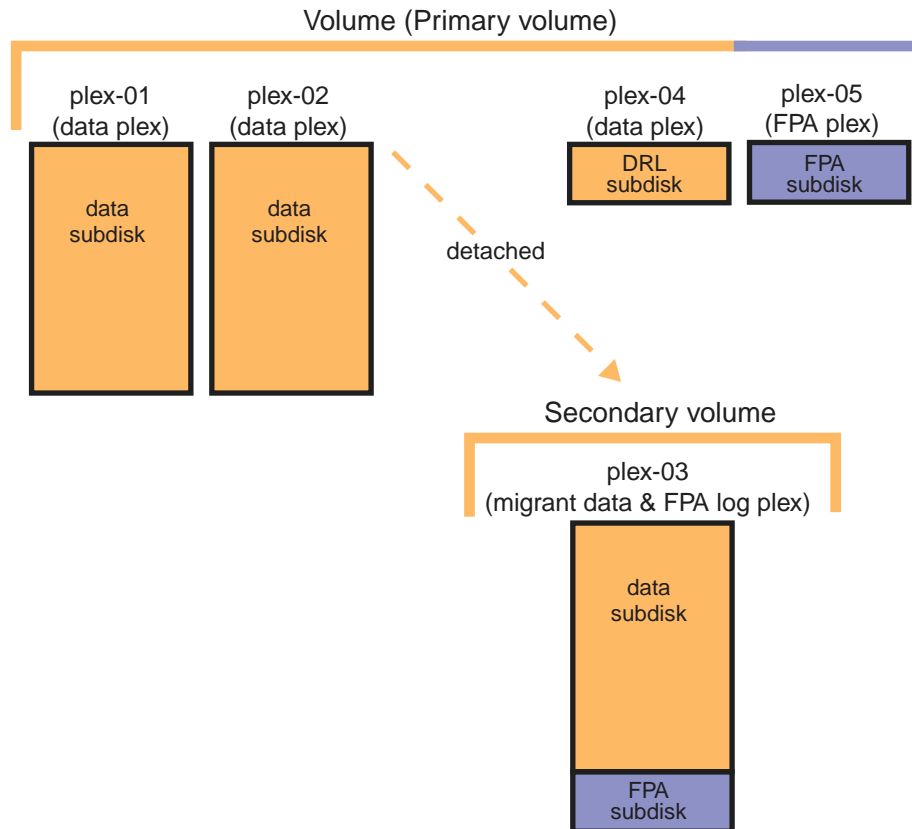
Figure 5–2: Process of the volassist snapfast Command: Phase 1



ZK-1876U-AI

2. LSM creates an FPA subdisk for the migrant plex and attaches this to the migrant plex as its FPA log. LSM then detaches the migrant plex from the primary volume and creates a secondary volume (Figure 5–3).

Figure 5–3: Process of the volassist snapfast Command: Phase 2

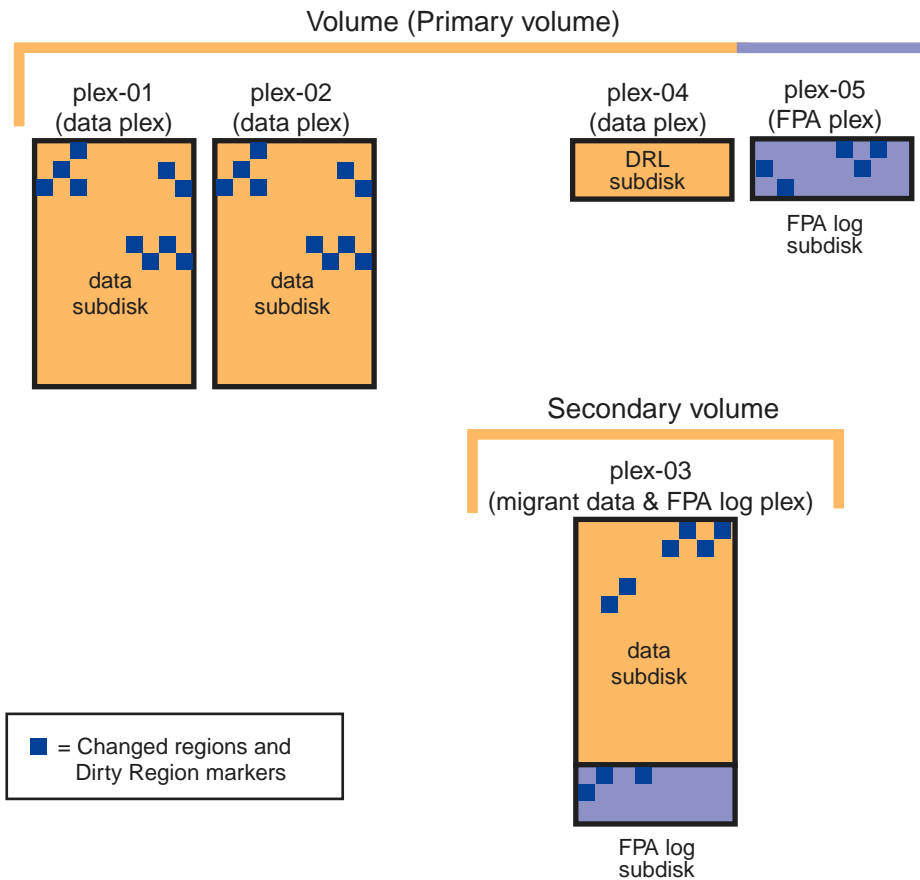


ZK-1877U-AI

LSM uses available disk space in the disk group for both FPA logs, using disks marked as hot-spares only if no other suitable space is available. LSM will not use space on disks marked as reserved or volatile. (For a description of these attributes, see `voledit(8)` and `voldisk(8)`.)

As writes occur to both volumes, their respective FPA logs keep track of the regions that changed (Figure 5–4). The FPA log subdisk attached to the migrant plex keeps track of changes to the secondary volume (such as the I/O that occurs when you mount the secondary volume so you can back it up).

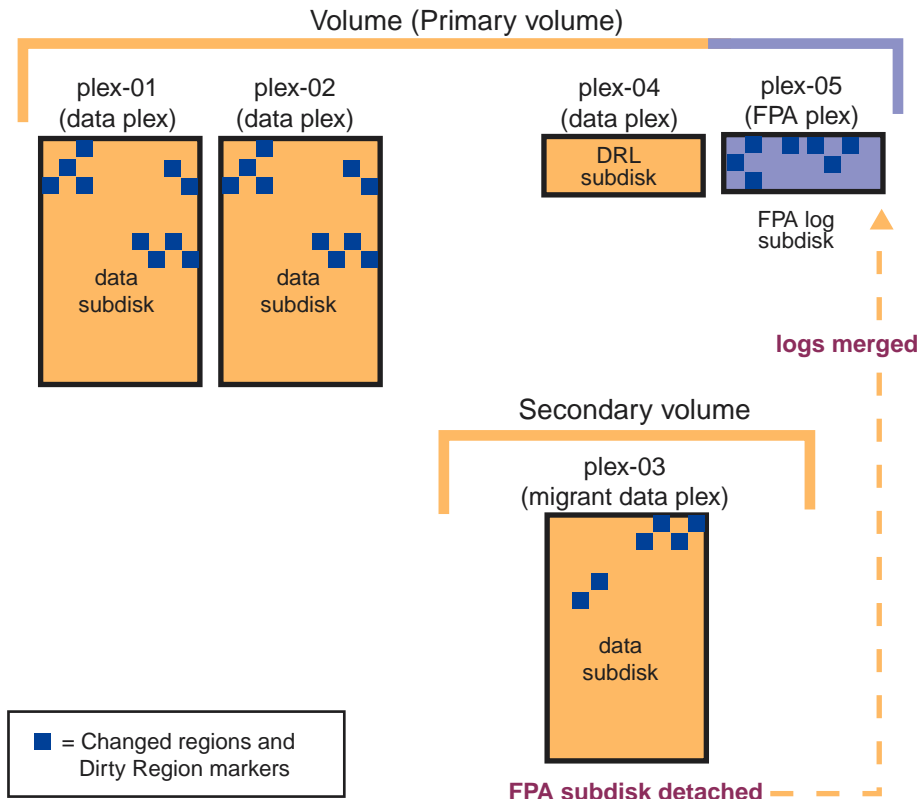
Figure 5–4: Writes Occurring to Primary and Secondary LSM Volumes



ZK-1878U-AI

3. Back up the secondary volume.
4. When the backup is complete, reattach the migrant plex to the primary volume with the `volassist snapback` command.
5. LSM removes the FPA log subdisk from the secondary volume and merges it with the FPA log subdisk on the primary volume (Figure 5–5).

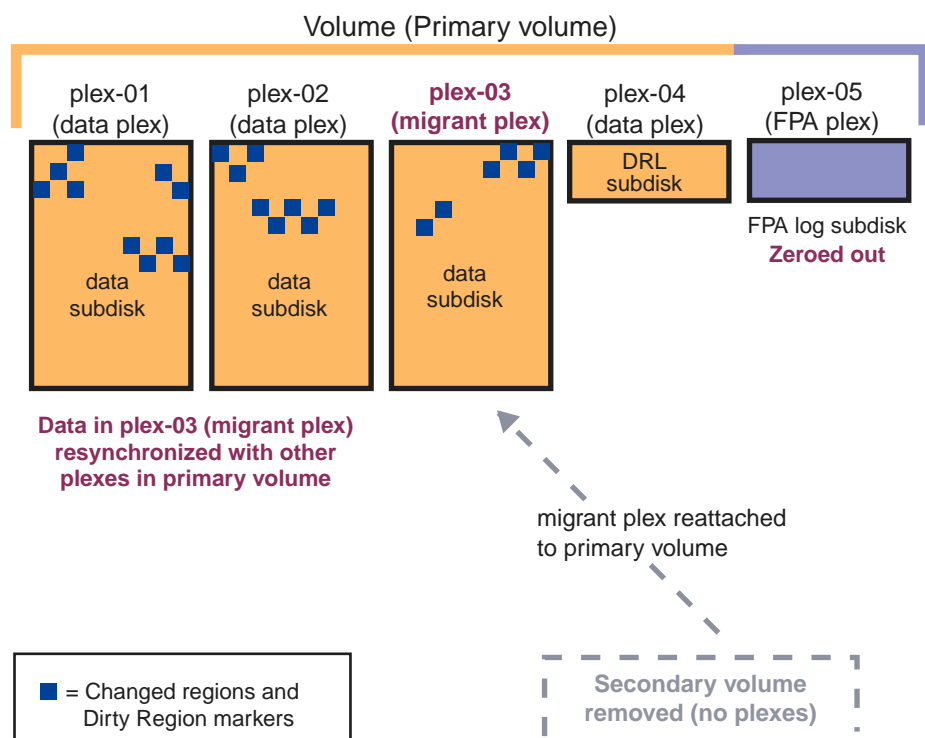
Figure 5–5: Process of volassist snapback Command: Phase 1



ZK-1879U-AI

- LSM reattaches the migrant plex to the primary volume (and removes the secondary volume if it has no other data plexes) and starts the process of resynchronizing the migrant plex to the primary volume according to the regions marked in the merged FPA log (Figure 5–6).

Figure 5–6: Process of volassist snapback Command: Phase 2



ZK-1880U-AI

Any writes that occurred to the secondary volume are ignored; the corresponding regions of the primary volume are written to the returning migrant plex. Also, only the regions of the primary volume that changed in the interim are written to the returning migrant plex, instead of resynchronizing the entire volume. This can greatly reduce the time it takes for a volume to resynchronize and thereby reduce the performance impact.

Depending on how long the migrant plex was away and how active the primary volume was during its absence, resynchronizing only the regions marked in each volume's logs is likely to be faster than resynchronizing the entire volume.

5.4.2.2 Creating a Backup Volume Using the Fast Plex Attach Feature (volassist snapfast and snapback)

You can use the Fast Plex Attach feature to create a temporary backup volume from one plex of a mirrored volume. This is sometimes called splitting a mirror, but is more accurately described as detaching a mirror.

The volume must have at least two complete, enabled plexes before you begin. If the volume has only two plexes, the volume will not be mirrored during the time one plex is detached. You must use the force (-f) option to detach a plex in this case.

Note

When you use the Fast Plex Attach feature to create a backup volume for `rootvol`, which has the special usage type of `root`, LSM creates the backup volume with the usage type of `gen`. Only the `rootvol` volume can have a usage type of `root`. In output from the `volprint` command, you might notice the different usage type for the backup volume for `rootvol`. The usage type does not affect backup operations.

The Fast Plex Attach feature does not let you specify a particular plex to use as the migrant plex. To control which plex is used, you can:

- Run the `volassist snapstart` command. When complete, the volume will have a new plex with a state of `SNAPDONE`. You can then run the `volassist snapfast` command and LSM will use the plex marked `SNAPDONE` as the migrant plex.

There must be available space in the disk group equal to the size of the volume to use the `volassist snapstart` command.

- Use the following commands to detach a specific plex from the primary volume, enable Fast Plex Attach logging on both the primary volume and the migrant plex, and create the secondary volume from that plex:

```
# volplex det -o fpa plex
# volplex att plex secondary_volume
```

Back up the secondary volume as described in the following procedure. When finished, you can use the `volassist snapback` command as shown in step 4.

To back up a volume using the Fast Plex Attach feature:

1. Verify that the volume you want to back up has more than two complete, enabled plexes available; for example:

```
# volprint -vht 3wayvol

Disk group: rootdg

V  NAME          USETYPE    KSTATE    STATE    LENGTH    READPOL    PREFPLEX
PL NAME         VOLUME     KSTATE    STATE    LENGTH    LAYOUT     NCOL/WID  MODE
SD NAME        PLEX       DISK      DISKOFFS LENGTH    [COL/]OFF  DEVICE    MODE

v  3wayvol       fsgen      ENABLED   ACTIVE   1024      SELECT     -         -
pl 3wayvol-01   3wayvol    ENABLED   ACTIVE   1024      CONCAT     -         RW
sd dsk0-01      3wayvol-01 dsk0      0        1024      0          dsk0     ENA
```

```

pl 3wayvol-02 3wayvol      ENABLED ACTIVE 1024  CONCAT -      RW
sd dsk2-01   3wayvol-02 dsk2    0      1024  0      dsk2  ENA
pl 3wayvol-03 3wayvol      ENABLED ACTIVE 1024  CONCAT -      RW
sd dsk7-02   3wayvol-03 dsk7    128   1024  0      dsk7  ENA
pl 3wayvol-04 3wayvol      ENABLED ACTIVE LOGONLY CONCAT -      RW
sd dsk1-01   3wayvol-04 dsk1    0      65    LOG    dsk1  ENA

```

2. Enter the following command to enable Fast Plex Attach logging on the primary volume, detach one plex using the force (-f) option if necessary, and create the secondary volume with the name of your choice:

```
# volassist [-f] snapfast primary_vol secondary_vol
```

For example:

```
# volassist -f snapfast 3wayvol 3wayvol_bk
```

LSM creates a secondary volume from one plex of the primary volume. The volumes look like the following:

```

v 3wayvol      fsgen      ENABLED ACTIVE 1024  SELECT -      RW
pl 3wayvol-01 3wayvol    ENABLED ACTIVE 1024  CONCAT -      RW
sd dsk0-01   3wayvol-01 dsk0     0      1024  0      dsk0  ENA
pl 3wayvol-02 3wayvol    ENABLED ACTIVE 1024  CONCAT -      RW
sd dsk2-01   3wayvol-02 dsk2     0      1024  0      dsk2  ENA
pl 3wayvol-04 3wayvol    ENABLED ACTIVE LOGONLY CONCAT -      RW
sd dsk1-01   3wayvol-04 dsk1     0      65    LOG    dsk1  ENA
pl 3wayvol-05 3wayvol    ENABLED ACTIVE FPAONLY CONCAT -      RW
sd dsk4-05   3wayvol-05 dsk4     524546 65    FPA    dsk4  ENA

v 3wayvol_bk   fsgen      ENABLED ACTIVE 1024  ROUND  -      RW
pl 3wayvol-03 3wayvol_bk ENABLED ACTIVE 1024  CONCAT -      RW
sd dsk0-03   3wayvol-03 dsk0     1152   65    FPA    dsk0  ENA
sd dsk7-02   3wayvol-03 dsk7     128   1024  0      dsk7  ENA

```

3. Back up the secondary volume using your preferred backup method. For more information, see Section 5.4.2.6.
4. When the backup is complete, reattach the migrant plex to the primary volume; for example:

```
# volassist snapback 3wayvol_bk 3wayvol
```

If the secondary volume has no other plexes, it is removed completely.

The FPA log plex remains attached to the primary volume to support future snapfast operations, as shown:

```
# volprint
```

```

v 3wayvol      fsgen      ENABLED ACTIVE 1024  SELECT -      RW
pl 3wayvol-01 3wayvol    ENABLED ACTIVE 1024  CONCAT -      RW
sd dsk0-01   3wayvol-01 dsk0     0      1024  0      dsk0  ENA
pl 3wayvol-02 3wayvol    ENABLED ACTIVE 1024  CONCAT -      RW
sd dsk2-01   3wayvol-02 dsk2     0      1024  0      dsk2  ENA
pl 3wayvol-03 3wayvol    ENABLED ACTIVE 1024  CONCAT -      RW
sd dsk7-02   3wayvol-03 dsk7     128   1024  0      dsk7  ENA
pl 3wayvol-04 3wayvol    ENABLED ACTIVE LOGONLY CONCAT -      RW
sd dsk1-01   3wayvol-04 dsk1     0      65    LOG    dsk1  ENA
pl 3wayvol-05 3wayvol    ENABLED ACTIVE FPAONLY CONCAT -      RW
sd dsk4-05   3wayvol-05 dsk4     524546 65    FPA    dsk4  ENA

```

5.4.2.3 Creating a Backup Volume by Attaching a New Plex (volassist snapstart and snapshot)

The following procedure describes how to add a plex to a volume (mirror the volume) and then detach the new plex to create a separate volume for backups. Detaching a plex is sometimes called splitting a mirror; in this case, you create the new mirror to detach.

This backup method uses available disk space in the disk group for the new plex, equal to the size of the volume, and requires extra time for the new plex to be fully synchronized to the volume before you can detach it to make the backup volume. However, this backup method eliminates the need to detach a plex from a volume and therefore preserves redundancy for a mirrored volume.

If you do not have sufficient disk space or do not want to mirror the volume, see Section 5.4.2.5.

To back up an LSM volume by adding and then detaching a new plex:

1. Display the size of the LSM volume and the disks it uses:

```
# volprint -v [-g disk_group] volume
```
2. Ensure there is enough free space in the disk group to add a plex to the LSM volume:

```
# voldg [-g disk_group] free
```

The amount of free space must be at least equal to the size of the volume and must be on disks that are not used in the volume you want to back up.
3. Add a new plex to the volume, optionally specifying the disks to use:

```
# volassist snapstart volume [disk...]
```

This step initiates a full synchronization, which might take several minutes or longer, depending on the size of the volume.
4. Create a temporary volume from the new plex. (The snapshot keyword uses the plex created in step 4 to create the new volume.)

```
# volassist snapshot volume temp_volume
```

The following example creates a temporary volume named vol3_backup for a volume named vol3:

```
# volassist snapshot vol3 vol3_backup
```
5. Start the temporary volume:

```
# volume start temp_volume
```
6. Back up the temporary volume using your preferred backup method. For more information, see Section 5.4.2.6.

7. When the backup is complete, do one of the following:
 - If you no longer need the backup volume, stop and remove the volume:


```
# volume stop temp_volume
# voledit -r rm temp_volume
```
 - If you want to reattach the plex to the original volume:
 - a. Stop the backup volume:


```
# volume stop temp_volume
```
 - b. Dissociate the plex from the backup volume (requires the force [-f] option):


```
# voledit -f dis plex
```
 - c. Remove the now empty backup volume:


```
# voledit -o rm temp_volume
```
 - d. Attach the backup plex to the original volume:


```
# volplex att plex original_volume
```

This initiates a full resynchronization of the volume's contents to the plex.

5.4.2.4 Creating a Backup Volume by Detaching an Existing Plex

This procedure is recommended only for administrators who have experience with using the low-level LSM commands and who have particular reasons for using this method instead of those described in the previous sections.

If used on a volume with only two enabled plexes, this procedure leaves the original volume unmirrored while one of the plexes is detached.

To back up an LSM volume from one of its existing plexes using the low-level commands:

1. Dissociate one of the volume's complete, enabled plexes, which leaves the plex with an image of the LSM volume at the time of dissociation:

```
# volplex dis plex
```

For example:

```
# volplex dis data-02
```

2. Create a temporary LSM volume using the dissociated plex:

```
# volmake -U fsgen vol temp_volume plex=plex
```

For example:

```
# volmake -U fsgen vol data_temp plex=data-02
```


3. Start the temporary volume:


```
# volume start temp_volume
```

 For example:


```
# volume start data_temp
```
4. Back up the temporary volume using your preferred backup method. For more information, see Section 5.4.2.6.
5. When the backup is complete, stop and remove the temporary volume:


```
# volume stop temp_volume
```

```
# voledit -r rm temp_volume
```
6. Reattach the dissociated plex to the original volume.


```
# volplex att volume plex
```

LSM automatically resynchronizes the plexes when you reattach the dissociated plex. This operation might take a long time, depending on the size of the volume. Running this process in the background returns control of the system to you immediately instead of waiting until the resynchronization is complete.

5.4.2.5 Backing Up a Nonredundant or RAID 5 Volume

If the volume uses a RAID 5 plex layout or if you cannot add a mirror to a volume with a single striped or concatenated data plex, you must either stop all applications from using the volume while the backup is in process or allow the backup to occur while the volume is in use.

If the volume remains in use during the backup, the volume data might change before the backup completes, and therefore the backup data will not be an exact copy of the volume's contents. The following procedure stops the volume to eliminate the risk of data corruption in the backup.

To back up a nonredundant or RAID 5 volume:

1. If applicable, select a convenient time and inform users to save files and refrain from using the volume, the application, or file system that uses the volume while you back it up.
2. Stop the volume:


```
# volume stop volume
```

 For example:


```
# volume stop r5_vol
```
3. Back up the volume using your preferred backup method. For more information, see Section 5.4.2.6.
4. When the backup is complete, restart the volume:

```
# volume start volume
```

For example:

```
# volume start r5_vol
```

5. If applicable, inform users that the volume is available.

5.4.2.6 Performing a Backup

You can use any of the following methods to back up the data in an LSM volume:

- If the volume contains a UFS file system, you can use the `dump` command; for example:

```
# dump -0u /dev/rvol/rootdg/r5_vol
```

For more information, see `dump(8)`.

- If the volume contains raw data such as for a database application, use the application's built-in backup utility. Be careful to select the temporary volume name as the object to back up.
- If the application has no backup utility, you can use the `dd` command to make an image copy of the volume's contents; for example:

```
# dd if=/dev/rvol/rootdg/3wayvol_bk of=/dev/tape/tape0_d0
```

For more information, , including options for specifying input and output block sizes, see `dd(8)`.

5.4.2.7 Special Case: Backing Up LSM Volumes in an AdvFS Domain

Because AdvFS domains can use storage on several devices, including several LSM volumes, it is critical for backups to capture the state of the domain, filesets, and the AdvFS metadata at the same point in time.

Caution

Before you begin, read the information on backing up and restoring data in the *AdvFS Administration* manual.

The following procedure involves temporarily freezing the domain to ensure the metadata is in a consistent state until after you create the backup LSM volumes. To minimize the time the domain must remain frozen, review Section 5.4.2.2, Section 5.4.2.3, Section 5.4.2.4, and Section 5.4.2.5, as appropriate, to determine what you must do to create each backup volume. You can perform some tasks ahead of time, such as finding free space in a disk group and adding a mirror to a volume.

To back up an AdvFS domain that uses several LSM mirrored volumes:

1. Freeze the domain. (The default freeze period is 60 seconds.) Freezing any mount point in the domain freezes the entire domain.

```
# /usr/sbin/freezefs /mount_point
```

2. For each LSM volume in the domain, create a backup volume using the appropriate procedure.

3. If the freeze time has not already elapsed, thaw the domain:

```
# /usr/sbin/thawfs /mount_point
```

4. Create a new domain directory and link the newly created LSM backup volumes to the temporary domain; for example:

```
# mkdir /etc/fdmns/my_dom_BK
# ln -s /dev/vol/rootdg/vol_1_backup /etc/fdmns/my_dom_BK
# ln -s /dev/vol/rootdg/vol_2_backup /etc/fdmns/my_dom_BK
```

Do not use the `mkfdmn` command to create the domain on the LSM volumes; this initializes the backup LSM volumes and destroys the existing data in the volumes.

5. Create a temporary backup directory at the root directory level:

```
# mkdir /backup
```

6. Display the filesets in the domain you are backing up:

```
# showfsets /etc/fdmns/my_dom
```

```
my_files
  Id      : 3caa0e34.000ef531.1.8001
  Files   :      0,  SLim=      0,  HLim=      0
  Blocks (512) :      0,  SLim=      0,  HLim=      0
  Quota Status : user=off group=off
  Object Safety: off
  Fragging   : on
  DMAPi     : off
```

7. Dual-mount the filesets in the new domain to the temporary backup mount point:

```
# mount -o dual temp_domain#parent_fileset /backup
```

For example:

```
# mount -o dual my_dom_BK#my_files /backup
```

8. Perform the backup on the temporary domain:

```
# vdump [options] [backup_device] /backup
```

9. When the backup is complete, unmount the temporary domain, and stop and remove the LSM volumes:

```
# umount /backup
# volume stop backup_volume...
```

```
# voledit -fr rm backup_volume...
```

The following example shows the entire process:

- The original domain is called `data_dmn` with a fileset called `data_files`, mounted on `/data`.
- The domain uses two mirrored volumes called `vol_1` and `vol_2`.
- The backup LSM volumes are called `vol_1_backup` and `vol_2_backup`.
- The backup domain is called `data_dmn_bk`, mounted on `/backup`.

```
# volprint -vht vol_1 vol_2
Disk group: rootdg
```

V NAME	USETYPE	KSTATE	STATE	LENGTH	READPOL	PREFPLEX		
PL NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WID	MODE	
SD NAME	PLEX	DISK	DISKOFFS	LENGTH	[COL/]OFF	DEVICE	MODE	
v vol_1	fsgen	ENABLED	ACTIVE	262144	SELECT	-		
pl vol_1-01	vol_1	ENABLED	ACTIVE	262144	CONCAT	-	RW	
sd dsk0-01	vol_1-01	dsk0	0	262144	0	dsk0	ENA	
pl vol_1-02	vol_1	ENABLED	ACTIVE	262144	CONCAT	-	RW	
sd dsk2-01	vol_1-02	dsk2	0	262144	0	dsk2	ENA	
pl vol_1-03	vol_1	ENABLED	ACTIVE	LOGONLY	CONCAT	-	RW	
sd dsk1-01	vol_1-03	dsk1	0	65	LOG	dsk1	ENA	
v vol_2	fsgen	ENABLED	ACTIVE	262144	SELECT	-		
pl vol_2-01	vol_2	ENABLED	ACTIVE	262144	CONCAT	-	RW	
sd dsk3-01	vol_2-01	dsk3	0	262144	0	dsk3	ENA	
pl vol_2-02	vol_2	ENABLED	ACTIVE	262144	CONCAT	-	RW	
sd dsk5-01	vol_2-02	dsk5	0	262144	0	dsk5	ENA	
pl vol_2-03	vol_2	ENABLED	ACTIVE	LOGONLY	CONCAT	-	RW	
sd dsk1-02	vol_2-03	dsk1	65	65	LOG	dsk1	ENA	

```
# /usr/sbin/freezefs /data_1
# volassist -f snapfast vol_1 vol_1_backup
# volassist -f snapfast vol_2 vol_2_backup
# /usr/sbin/thawfs /ka
# cd /etc/fdmns
# mkdir data_dmn_bk
# ln -s /dev/vol/rootdg/vol_1_backup /etc/fdmns/data_dmn_bk
# ln -s /dev/vol/rootdg/vol_2_backup /etc/fdmns/data_dmn_bk
# mkdir /backup
# showfssets ka_dom
data_files1
  Id          : 3cadbee2.000aca16.1.8001
  Files       :      5,  SLim=      0,  HLim=      0
  Blocks (512) : 29340,  SLim=      0,  HLim=      0
  Quota Status : user=off group=off
  Object Safety: off
  Fragging    : on
  DMAPi      : off

# mount -o dual data_dmn_bk#data_files1 /backup
# vdump [options] [backup_device] /backup
```

5.4.3 Restoring LSM Volumes from Backup

The way you restore an LSM volume depends on what the volume is used for, whether the volume is configured and active, and the method you used to back up volume data.

- If you used the `vdump` command to back up the volume (used by UFS file systems or AdvFS domains), use the `vrestore` command to restore the data.
- If you used the `rvdump` command, use the `rvrestore` command to restore it.
- If the volume is used for an application such as a database, see that application's documentation for the recommended method for restoring backed-up data.

To restore a volume for a UFS file system from a backup that you created with the `dump` command:

```
# restore -Yf backup_volume
```

Note

Both the original volume and the backup volume must be mounted.

You can restore a volume and its contents even if the volume no longer exists, if you have a backup copy of the configuration database. (See Section 5.3.1.)

To restore a volume that no longer exists:

1. Recreate the volume:

```
# volrestore [-g disk_group] -v volume
```

This recreates the structure of the volume on the same disks it was using at the time you saved the configuration with the `volsave` command. You cannot assume the data on the disks is valid at this point.

2. Recreate the file system:

```
# newfs /dev/rvol/disk_group/volume
```

3. Mount the file system:

```
# mount /dev/vol/disk_group/volume /mount_point
```

4. Change directory to the recreated file system:

```
# cd /mount_point
```

5. Restore the volume data:

```
# restore -Yrf backup_volume
```

For more information, see `restore(8)`, `rrestore(8)`, and `vrestore(8)`.

5.4.4 Starting LSM Volumes

LSM automatically starts all startable volumes when the system boots. You can manually start an LSM volume that:

- You manually stopped
- Belongs to a disk group that you manually imported
- Stopped because of a disk failure or other problem that you have since resolved

To start an LSM volume:

```
# volume [-g disk_group] start volume
```

To start all volumes in a disk group (for example, after importing the disk group):

```
# volume -g disk_group startall
```

5.4.5 Stopping LSM Volumes

LSM automatically stops LSM volumes when the system shuts down. When you no longer need an LSM volume, you can stop it then remove it. You cannot stop an LSM volume if a file system is using it.

To stop an LSM volume:

1. If applicable, stop a file system from using the LSM volume.

- For AdvFS, dissociate the volume from the domain:

```
# rmvol LSM_volume domain
```

Data on the volume is automatically migrated to other volumes in the domain, if available. For more information, see the *AdvFS Administration* manual and `rmvol(8)`.

- For UFS, unmount the file system:

```
# umount /dev/rvol/volume
```

2. Stop the LSM volume:

```
# volume [-g disk_group] stop volume
```

For example, to stop an LSM volume named `vol101` in the `dg1` disk group:

```
# volume -g dg1 stop vol101
```

To stop all volumes in a disk group:

```
# volume -g disk_group stopall
```

5.4.6 Removing LSM Volumes

Removing an LSM volume destroys the data in that volume. Remove an LSM volume only if you are sure that you do not need the data in the LSM volume or the data is backed up elsewhere. When an LSM volume is removed, the space it occupied is returned to the free space pool.

Note

- To remove a volume that was created by encapsulating an AdvFS domain, see Section 5.4.6.1.
 - To remove a swap volume for a cluster member, see Section 7.6.
-

The following procedure also unencapsulates UFS file systems.

To remove an LSM volume:

1. If applicable, stop a file system from using the LSM volume.
 - If the volume is part of an AdvFS domain, dissociate the volume from the domain:

```
# rmvol LSM_volume domain
```

Data on the volume is automatically migrated to other volumes in the domain. For more information, see the *AdvFS Administration* manual and `rmvol(8)`.
 - If the volume is used by a UFS file system, unmount the file system:

```
# umount /dev/rvol/volume
```
2. Edit the necessary system files as follows:
 - If the volume was configured as secondary swap (for a standalone system), remove references to the LSM volume from the `vm:swapdevice` entry in the `sysconfigtab` file.
 - If the swap space was configured using the `/etc/fstab` file, update this file to change the swap entries back to disk partitions instead of LSM volumes.

These changes are effective the next time the system restarts.

For more information, see the *System Administration* manual and `swapon(8)`.

3. Stop the LSM volume:

```
# volume [-g disk_group] stop volume
```
4. Remove the LSM volume:

```
# voledit -r rm volume
```

This step removes the plexes and subdisks and the volume itself.

5. If the volume contained an encapsulated UFS file system, edit the `/etc/fstab` file to change the volume name to the disk name. For example, change `/dev/vol/rootdg/vol-dsk4g` to `/dev/dsk4g`.

5.4.6.1 Unencapsulating AdvFS Domains

You can stop using LSM volumes for an AdvFS domain and revert to using the physical disks or disk partitions directly. This is called unencapsulating a domain.

To unencapsulate the storage in an AdvFS domain, you can use the script that LSM created when you originally encapsulated the domain. The script contains both LSM commands and general operating system commands and performs all the steps necessary to remove the LSM volumes, remove the disks from LSM control, and update the links in the `/etc/fdmns` directory.

One script for each disk or storage device in the domain is saved in a subdirectory named after the disk access name in the following format:

```
/etc/vol/reconfig.d/disk.d/dsknp.encapdone/recover.sh
```

Make sure you run the correct script to unencapsulate the storage in the appropriate domain.

Note

Unencapsulating storage in an AdvFS domain requires that you unmount the filesets.

To unencapsulate an AdvFS domain:

1. Identify the name of the LSM volume for the domain (which is typically derived from the disk name):

```
# showfdmn domain
```

For example:

```
          Id                Date Created  LogPgs  Version  Domain Name
3a65b2a9.0004cb3f  Wed Jan 17 09:56:41 2001      512      4    dom_1

Vol   512-Blks      Free  % Used  Cmode  Rblks  Wblks  Vol Name
1L   8380080      8371248    0%    on    256    256  /dev/vol/tempdg/vol-dsk26c
```

2. Display and unmount all the filesets in the domain. For example, to unmount the filesets in the `dom_1` domain:

```
# mount
root_domain#root on / type advfs (rw)
/proc on /proc type procfs (rw)
usr_domain#usr on /usr type advfs (rw)
```



```
var_domain#var on /var type advfs (rw)
mhs:/work on /work type nfs (v3, rw, udp, hard, intr)
dom_1#junk on /junk type advfs (rw)
dom_1#stuff on /stuff type advfs (rw)
# umount /junk /stuff
```

3. Stop the LSM volume:

```
# volume stop volume
```

4. Find the name of the appropriate unencapsulation script, if one exists:

```
# ls /etc/vol/reconfig.d/disk.d/
dsk23c.encapdone dsk24c.encapdone dsk26c.encapdone
dsk27g.encapdone
```

5. Run the appropriate unencapsulation script.

For example, to run the script for the dom_1 domain on disk dsk26c:

```
# sh /etc/vol/reconfig.d/disk.d/dsk26c.encapdone/recover.sh
```

6. If the script is not available, do the following:

- a. Change directory to the domain directory:

```
# cd /etc/fdmns/domain
```

- b. Remove the link to the volume:

```
# rm disk_group.volume
```

- c. Replace the link to the disk device file:

```
# ln -s /dev/disk/dsknp
```

- d. Remove the LSM volume:

```
# voledit [g disk_group] -r rm volume
```

- e. Remove the disk media name from the disk group:

```
# voldg -g disk_group rmdisk dm_name
```

- f. Remove the disk access name from LSM:

```
# voldisk rm da_name
```

7. Remount the filesets to the domain:

```
# mount dom_1#junk /junk
# mount dom_1#stuff /stuff
```

The domain is available for use. I/O to the domain goes through the disk device path instead of the LSM volume. You can confirm this by running the showfdmn command again:

```
# showfdmn dom_1
```

Id	Date Created	LogPgs	Version	Domain Name
3a65b2a9.0004cb3f	Wed Jan 17 09:56:41 2001	512	4	dom_1

Vol	512-Blks	Free	% Used	Cmode	Rblks	Wblks	Vol Name
1L	8380080	8371248	0%	on	256	256	/dev/disk/dsk26c

5.4.7 Disabling the Fast Plex Attach Feature on LSM Volumes

If you no longer want to use FPA logging on a volume, you can turn the feature off for that volume. Turning off FPA logging does not remove the FPA log from the primary volume. You can turn the FPA feature back on later if you choose.

If the volume is actively logging to the FPA log plex (if a secondary volume exists), turning off the FPA feature stops logging. If you then return the migrant plex to the primary volume (with the `volassist snapback` command), the plex undergoes complete resynchronization.

To turn off the FPA feature for a volume:

```
# volume set fpa=off volume
```

To remove an FPA log plex, see Section 5.5.6.

5.4.8 Renaming LSM Volumes

You can rename an LSM volume. The new LSM volume name must be unique within the disk group. If the LSM volume supports a file system or is part of an AdvFS domain, you must also update the `/etc/fstab` file and the `/etc/fdmns` directory entry.

To rename an LSM volume:

```
# voledit rename old_volume new_volume
```

Note

If you do not update the relevant files in the `/etc` directory before the system is restarted, subsequent commands using a volume's previous name will fail.

5.4.9 Growing LSM Volumes

You can increase the size of a volume (grow a volume) by specifying either an amount to grow by or a size to grow to. For example, you can increase the size of the primary swap space volume. The size of any log plexes remains unchanged.

Note

After you grow a volume that is used by an AdvFS file system, use the `mount -o extend` command to update the domain to include the additional space in the LSM volume. For more information on increasing the size of a domain, see the *AdvFS Administration* manual and `mount(8)`.

If the volume is used by a file system other than AdvFS, you must perform additional steps specific to the file system type for the file system to take advantage of increased space. For more information, see the *System Administration* manual, `extendfs(8)`, and `mount(8)`.

If an application other than a file system uses the volume, make any necessary application modifications after the grow operation is complete.

5.4.9.1 Growing LSM Volumes by a Specific Amount

Use the `-f` option to grow a primary or secondary volume that is actively logging to an FPA log plex.

Note

Growing a primary or secondary volume with an active FPA log disables the FPA log. When you reattach a migrant plex to a primary volume in this case, regardless of which volume grew, a full resynchronization will occur as though no FPA log ever existed.

You can use the `-b` option to perform the operation in the background. This is helpful if the *growby* length specified is substantial and if the volume uses mirror plexes or a RAID 5 plex, because it will undergo resynchronization as a result of the grow operation.

To grow a volume by a specific amount:

```
# volassist [-g disk_group] [-f] [-b] growby volume length_change
```

For example, to grow a volume by 100K bytes:

```
# volassist -g dg1 growby myVol 100k
```

The volume looks similar to the following, before and after the `growby` operation:

```
# volprint -g dg1 -vht dataVol
```

V NAME	USETYPE	KSTATE	STATE	LENGTH	READPOL	PREFPLEX
PL NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WID MODE

SD NAME	PLEX	DISK	DISKOFFS	LENGTH	[COL/]OFF	DEVICE	MODE
v dataVol	fsgen	ENABLED	ACTIVE	400	SELECT	dataVol-01	
pl dataVol-01	dataVol	ENABLED	ACTIVE	512	STRIPE	2/128	RW
sd dsk4-01	dataVol-01	dsk4	0	256	0/0	dsk4	ENA
sd dsk5-01	dataVol-01	dsk5	0	256	1/0	dsk5	ENA

```
# volassist -g dg1 growby dataVol 100k
# volprint -vht -g dg1 dataVol
```

V NAME	USETYPE	KSTATE	STATE	LENGTH	READPOL	PREFPLEX	
PL NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WID	MODE
SD NAME	PLEX	DISK	DISKOFFS	LENGTH	[COL/]OFF	DEVICE	MODE
v dataVol	fsgen	ENABLED	ACTIVE	600	SELECT	dataVol-01	
pl dataVol-01	dataVol	ENABLED	ACTIVE	768	STRIPE	2/128	RW
sd dsk4-01	dataVol-01	dsk4	0	384	0/0	dsk4	ENA
sd dsk5-01	dataVol-01	dsk5	0	384	1/0	dsk5	ENA

In this case, LSM was able to grow the subdisks using contiguous space on the same disks. If the subdisks in a volume map to the end of the public region on the disks, LSM uses available space in the disk group to create and associate new subdisks to the volume's plex or plexes.

5.4.9.2 Growing LSM Volumes to a Specific Size

Use the `-f` option to grow a primary or secondary volume that is actively logging to an FPA log plex.

Note

Growing a primary or secondary volume with an active FPA log disables the FPA log. When you reattach a migrant plex to a primary volume in this case, regardless of which volume grew, a full resynchronization will occur as though no FPA log ever existed.

You can use the `-b` option to perform the operation in the background. This is helpful if the `growto` length specified is substantial and if the volume uses mirror plexes or a RAID 5 plex, because it will undergo resynchronization as a result of the grow operation.

To grow a volume to a specific size:

```
# volassist [-g disk_group] [-f] [-b] growto volume new_length
```

For example, to grow a volume from 1MB to 2MB:

```
# volassist -g dg1 growto vol_A 2m
```

The volume looks similar to the following, before and after the `growto` operation:

```
# volprint -vht -g dg1 vol_A
```

V NAME	USETYPE	KSTATE	STATE	LENGTH	READPOL	PREFPLEX
PL NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WID MODE
SD NAME	PLEX	DISK	DISKOFFS	LENGTH	[COL/]OFF	DEVICE MODE
v vol_A	fsgen	ENABLED	ACTIVE	2048	SELECT	vol_A-01
pl vol_A-01	vol_A	ENABLED	ACTIVE	2048	STRIPE	2/128 RW
sd dsk6-01	vol_A-01	dsk6	0	1024	0/0	dsk6 ENA
sd dsk7-01	vol_A-01	dsk7	0	1024	1/0	dsk7 ENA

```
# volassist -g dg1 growto vol_A 2m
# volprint -vht -g dg1 vol_A
```

V NAME	USETYPE	KSTATE	STATE	LENGTH	READPOL	PREFPLEX
PL NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WID MODE
SD NAME	PLEX	DISK	DISKOFFS	LENGTH	[COL/]OFF	DEVICE MODE
v vol_A	fsgen	ENABLED	ACTIVE	4096	SELECT	vol_A-01
pl vol_A-01	vol_A	ENABLED	ACTIVE	4096	STRIPE	2/128 RW
sd dsk6-01	vol_A-01	dsk6	0	2048	0/0	dsk6 ENA
sd dsk7-01	vol_A-01	dsk7	0	2048	1/0	dsk7 ENA

In this case, LSM was able to grow the subdisks using contiguous space on the same disks. If the subdisks in a volume already map to the end of the public region on the disks, LSM uses available space in the disk group to create and associate new subdisks to the volume's plex or plexes. Alternatively, you can specify which disks LSM can use to create the new subdisks.

The following example shows what happens when you specify disks for LSM to use:

```
# volassist -g dg1 growto dataVol 100k dsk9 dsk10
# volprint -vht -g dg1 dataVol
```

V NAME	USETYPE	KSTATE	STATE	LENGTH	READPOL	PREFPLEX
PL NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WID MODE
SD NAME	PLEX	DISK	DISKOFFS	LENGTH	[COL/]OFF	DEVICE MODE
v dataVol	fsgen	ENABLED	ACTIVE	800	SELECT	dataVol-01
pl dataVol-01	dataVol	ENABLED	ACTIVE	1024	STRIPE	2/128 RW
sd dsk4-01	dataVol-01	dsk4	0	384	0/0	dsk4 ENA
sd dsk9-01	dataVol-01	dsk9	0	128	0/384	dsk9 ENA
sd dsk5-01	dataVol-01	dsk5	0	384	1/0	dsk5 ENA
sd dsk10-01	dataVol-01	dsk10	0	128	1/384	dsk10 ENA

Notice that LSM added the subdisk it created on dsk9 to the end of the first column (column 0) and the subdisk it created on dsk10 to the second column (column 1).

5.4.10 Shrinking LSM Volumes

You can decrease the size of a volume by specifying either an amount to shrink by or a size to shrink to. The size of any log plexes remains unchanged.

Cautions

- If the volume is used for an AdvFS file system, do not decrease the space in the domain by shrinking an underlying LSM volume. Instead, remove a volume from the domain (in AdvFS, a volume can be a disk, disk partition, or an LSM volume). For more information on removing volumes from a domain, see the *AdvFS Administration* manual.
- If the volume is used for a file system other than AdvFS, you must perform additional steps specific to the file system type before shrinking the volume, so that the file system can recognize and safely adjust to the decreased space.

There is no direct way to shrink a UFS file system other than backing up the data, destroying the original file system, creating a new file system of the smaller size, and restoring the data in the new file system.

For more information, see the *System Administration* manual.

If an application other than a file system uses the volume, make any necessary application modifications before shrinking the LSM volume.

Note

Shrinking a (primary or secondary) volume with an active FPA log disables the FPA log. When you reattach a migrant plex to a primary volume in this case, regardless of which volume shrank, a full resynchronization will occur as though no FPA log ever existed.

5.4.10.1 Shrinking LSM Volumes by a Specific Amount

To shrink a volume by a specific amount:

```
# volassist [-g disk_group] -f shrinkby volume length_change
```

For example, to shrink a volume by 100K bytes:

```
# volassist -g dg1 -f shrinkby dataVol 100k
```

5.4.10.2 Shrinking LSM Volumes to a Specific Size

To shrink a volume to a specific size:

```
# volassist [-g disk_group] -f shrinkto volume new_length
```

For example, to shrink a volume from 2MB to 1MB:

```
# volassist -g dg1 -f shrinkto vol_A 1m
```

5.4.11 Changing LSM Volume Permission, User, and Group Attributes

By default, the device special files for LSM volumes are created with read and write permissions granted only to the owner. Databases or other applications that perform raw I/O might require device special files to have other settings for the permission, user, and group attributes.

Note

Use LSM commands instead of the `chmod`, `chown`, or `chgrp` commands to change the permission, user, and group attributes for LSM volumes. The LSM commands ensure that settings for these attributes are stored in the LSM database, which keeps track of all settings for LSM objects.

To change Tru64 UNIX user, group, and permission attributes:

```
# voledit [-g disk_group] set \  
user=username group=groupname mode=permission volume
```

The following example changes the user, group, and permission attributes for an LSM volume named `vol01` in the `rootdg` disk group:

```
# voledit set user=new_user group=admin mode=0600 vol01
```

5.5 Managing Plexes

The following sections describe how to use LSM commands to manage plexes.

5.5.1 Displaying Plex Information

To display general information for all plexes:

```
# volprint -pt
```

```
Disk group: rootdg
```

PL NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WID	MODE
pl ka1-01	ka1	ENABLED	ACTIVE	2097152	CONCAT	-	RW
pl ka2-01	ka2	ENABLED	ACTIVE	2097152	CONCAT	-	RW
pl ka3-01	ka3	ENABLED	ACTIVE	2097152	CONCAT	-	RW
pl ka4-01	ka4	ENABLED	ACTIVE	2097152	CONCAT	-	RW
pl rootvol-01	rootvol	ENABLED	ACTIVE	524288	CONCAT	-	RW
pl swapvol-01	swapvol	ENABLED	ACTIVE	520192	CONCAT	-	RW
pl tst-01	tst	ENABLED	ACTIVE	2097152	CONCAT	-	RW
pl tst-02	tst	ENABLED	ACTIVE	2097152	CONCAT	-	RW
pl tst-03	tst	ENABLED	ACTIVE	LOGONLY	CONCAT	-	RW
pl vol-dsk25g-01	vol-dsk25g	ENABLED	ACTIVE	2296428	CONCAT	-	RW
pl vol-dsk25h-01	vol-dsk25h	ENABLED	ACTIVE	765476	CONCAT	-	RW

To display detailed information about a specific plex:

```
# volprint -lp plex
```

```
Disk group: rootdg
```

```
Plex:      tst-01
info:      len=2097152
type:      layout=CONCAT
state:     state=ACTIVE kernel=ENABLED io=read-write
assoc:     vol=tst sd=dsk0-01
flags:     complete
```

5.5.2 Adding a Data Plex (Mirroring LSM Volumes)

You can add a data plex to a volume to mirror the data in the volume. You cannot use a disk that is already used by the volume to create the mirror. A volume can have up to 32 plexes, which can be any combination of data and log plexes.

Adding a plex is one way to move the LSM volume to disks with better performance, if you specify the disks for LSM to use. You can also add a plex temporarily, so you can repair or replace disks in the original plex.

When you add a plex to a volume, the volume data is copied to the new plex. This process can take several minutes to several hours depending on the size of the volume.

Caution

To mirror the volumes for the boot disk and primary swap space (`rootvol` and `swapvol`) on a standalone system, you must use the `volrootmir` command (Section 3.4.1). The `volrootmir` command performs special operations that ensure the system can boot from either mirror in the root file system volumes.

To add a data plex:

```
# volassist mirror volume [disk]
```

Note

Adding a data plex does not add a DRL plex to the volume. Volumes with mirror plexes should have a DRL plex, except volumes for swap space. To add a DRL plex to a volume, see Section 5.5.3.

5.5.3 Adding a Log Plex

You can add a log plex (DRL or RAID 5 log) to a volume that has mirrored data plexes or a RAID 5 data plex. LSM automatically creates the log of a size appropriate to the size of the volume (65 blocks per gigabyte of volume size, by default).

Note

A DRL plex is not supported on `rootvol` (on a standalone system), `cluster_rootvol` (in a cluster), or any swap volumes (standalone systems or clusters).

To improve performance and avoid the risk of losing both data and the log if the disk fails, create the log plex on a disk that is not in use by one of the volume's data plexes and, if possible, on a disk that does not already support the log for other volumes.

To add a log plex to a volume using any available space in the disk group:

```
# volassist [-g disk_group] addlog volume [disk]
```

To add a log to a volume by specifying a disk:

1. Make sure the disk is not in use by the same volume:

```
# volprint -vht volume
```

For example:

```
# volprint -vht genvol
```

```
Disk group: rootdg
```

TY	NAME	ASSOC	KSTATE	LENGTH	PLOFFS	STATE	TUTILO	PUTILO
v	genvol	gen	ENABLED	204800	-	ACTIVE	-	-
pl	genvol-01	genvol	ENABLED	204800	-	ACTIVE	-	-
sd	dsk19-02	genvol-01	ENABLED	204800	0	-	-	-
pl	genvol-02	genvol	ENABLED	204800	-	ACTIVE	-	-
sd	dsk5-01	genvol-02	ENABLED	204800	0	-	-	-

In the previous output, the volume `genvol` uses space for its data plexes on disks `dsk19` and `dsk5`.

2. Optionally but recommended, make sure the disk does not already support log plexes for other volumes:

```
# volprint -ht | grep disk
```

For example:

```
# volprint -ht | grep dsk5
```

sd	dsk5-03	fsvol-03	dsk5	322912	245760	0	dsk5	ENA
sd	dsk5-01	genvol-01	dsk5	118112	204800	0	dsk5	ENA
sd	dsk5-02	vol_r5-01	dsk5	32768	85344	4/0	dsk5	ENA

In the previous output, space on `dsk5` is used by volumes `fsvol`, `genvol`, and `vol_r5`, but not for their logs. (However, `dsk5` is already heavily used and would not be a good candidate for the new log plex.)

3. Add the log to the volume, specifying an eligible disk:

```
# volassist [-g disk_group] addlog volume disk
```

For example, to add a log plex to volume `genvol` on `dsk11`, which is not used in any other volume in the `rootdg` disk group:

```
# volassist addlog genvol dsk11
```

5.5.4 Adding an FPA Plex

You can add one or more Fast Plex Attach (FPA) logs to a mirrored volume to provide redundancy for the FPA logs, just as you can add multiple dirty-region logs (DRLs) to a volume.

FPA logging is supported on any mirrored volume on a standalone system or a cluster, including the `rootvol` and `cluster_rootvol` volumes, but excluding volumes used for swap space.

Note

You cannot add an FPA log to a volume while a migrant plex is detached from the volume (attached to a secondary volume).

If the volume has a DRL log, the FPA log length will be the same as the DRL log length. If the volume has no DRL log, when you add the first FPA log to a mirrored volume, you can specify the length of the log with the `loglen=length` attribute, the number of FPA logs with the `nfpalog=count` attribute, and which disks it can or cannot use. To exclude storage, use the `!` prefix (or `\!` in the C shell).

To add an FPA log plex to a volume, optionally specifying the number of FPA log plexes or the disk or disks to use:

```
# volassist addfpa volume [nfpalog=count] [disk...]
```

For example, to add two FPA log plexes on disks `dsk5` and `dsk6` to a volume named `dvol_01`:

```
# volassist addfpa dvol_1 [nfpalog=2] dsk5 dsk6
```

The volume looks like the following:

Disk group: `rootdg`

TY NAME	ASSOC	KSTATE	LENGTH	PLOFFS	STATE	TUTILO	PUTILO
v dvol_1	fsgen	ENABLED	245760	-	ACTIVE	-	-
pl dvol_1-01	dvol_1	ENABLED	245760	-	ACTIVE	-	-
sd dsk1-01	dvol_1-01	ENABLED	245760	0	-	-	-

pl	dvol_1-02	dvol_1	ENABLED	245760	-	ACTIVE	-	-
sd	dsk2-01	dvol_1-02	ENABLED	245760	0	-	-	-
pl	dvol_1-03	dvol_1	ENABLED	LOGONLY	-	ACTIVE	-	-
sd	dsk1-02	dvol_1-03	ENABLED	65	LOG	-	-	-
pl	dvol_1-04	dvol_1	ENABLED	FPAONLY	-	ACTIVE	-	-
sd	dsk5-01	dvol_1-04	ENABLED	65	FPA	-	-	-
pl	dvol_1-05	dvol_1	ENABLED	FPAONLY	-	ACTIVE	-	-
sd	dsk6-01	dvol_1-05	ENABLED	65	FPA	-	-	-

When you create a backup volume using the `volassist snapfast` command (Section 5.4.2.2), LSM tracks the changes to the primary volume in all the FPA logs associated with the volume.

5.5.5 Detaching a Plex

Detaching a plex preserves the plex's association to a volume. You can detach a plex to perform temporary operations with it, with the intent of reattaching the plex to the same volume later.

Caution

If you detach all but one data plex from a mirrored volume, the volume's data is no longer redundant.

Note the following:

- If you detach the last log plex from a mirrored volume and the system fails, LSM resynchronizes the contents of all the plexes when the system restarts. The volume remains usable during this operation, but its performance might be significantly degraded.
- If you detach the last RAID 5 log plex from a RAID 5 volume and the system fails, LSM recalculates the parity for the entire volume; this involves reading back all the volume data, regenerating the parity for each stripe, and rewriting each stripe in the plex. The volume remains usable during this operation, but its performance might be significantly degraded.
- If you detach the last FPA log plex from a primary volume, FPA logging is disabled. When you return the migrant plex to the primary volume, a full resynchronization occurs, as if FPA were never enabled. To disable FPA logging on a volume, see Section 5.4.7.

To detach a plex from a volume:

```
# volplex [-f] det plex
```

Use the force (`-f`) option to detach a volume's last complete, enabled data plex or the FPA log plex from a primary volume.

5.5.6 Dissociating a Plex

Dissociating a plex removes its association to a volume. You can dissociate a plex from a volume to unmirror a volume (remove all but one data plex) or reduce the number of mirrors or logs in a volume, usually with the intent of completely removing the plex and reusing its disk space for another purpose. To support this, you can dissociate and recursively remove a plex and its components in one command. Alternatively, you can dissociate a plex and use it to create another volume; for example, to back up a volume.

Note

The recommended method for backing up a mirrored volume is with the `volassist snapfast` command (to use FPA logging) or the `volassist snapshot` command. For more information, see Section 5.4.2.

Note the following:

- If you dissociate the last log plex from a mirrored volume and the system fails, LSM resynchronizes the contents of all the plexes when the system restarts. The volume remains usable during this operation, but its performance might be significantly degraded.
- If you dissociate the last RAID 5 log plex from a RAID 5 volume and the system fails, LSM recalculates the parity for the entire volume; this involves reading back all the volume data, regenerating the parity for each stripe, and rewriting each stripe in the plex. The volume remains usable during this operation, but its performance might be significantly degraded.
- If you dissociate the last FPA log plex from a primary volume, FPA logging is disabled. When you return the migrant plex to the primary volume, a full resynchronization occurs, as if FPA were never enabled. To disable FPA logging on a volume, see Section 5.4.7.

Caution

If you dissociate all but one data plex from a mirrored volume, the volume's data is no longer redundant.

To remove the last data plex from a volume (remove a volume completely), see Section 5.4.6.

To dissociate a plex from a volume:

```
# volplex [-f] dis plex
```

Use the force (-f) option to dissociate a volume's last complete, enabled data plex or the FPA log plex from a primary volume.

To dissociate and recursively remove a plex from a volume:

```
# volplex [-f] -o rm dis plex
```

Recursively dissociating a plex removes both the plex and its subdisks. The disks remain under LSM control.

5.5.7 Reattaching Plexes

If you detached or dissociated (but did not recursively remove) a data plex or log plex from a volume, you can reattach it to the volume. You can attach a detached plex only to its original volume.

To reattach a detached data or log plex to a volume:

```
# volrecover volume
```

To reattach a dissociated data or log plex to a volume:

```
# volplex att volume plex
```

5.5.8 Changing the Plex Layout of LSM Volumes

For volumes that use one or more concatenated or striped plexes, you can change the plex layout from concatenated to striped or vice versa. For example, you can change from a concatenated plex to a striped plex to improve performance.

The steps involved include adding a plex to the volume and then dissociating the original plex or plexes from the volume.

5.5.8.1 Changing the Plex Layout from Concatenated to Striped

To change the plex layout of a volume from concatenated to striped:

1. Display the size of the volume whose plex layout you want to change:

```
# volprint [-g diskgroup] -ht volume
```

```
Disk group: rootdg
```

V NAME	USETYPE	KSTATE	STATE	LENGTH	READPOL	PREFPLEX		
PL NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WID	MODE	
SD NAME	PLEX	DISK	DISKOFFS	LENGTH	[COL/]OFF	DEVICE	MODE	
v volC	fsgen	ENABLED	ACTIVE	204800	SELECT	-		
pl volC-01	volC	ENABLED	ACTIVE	204800	CONCAT	-	RW	
sd dsk2-01	volC-01	dsk2	0	204800	0	dsk2	ENA	

In this example, the volume volC has one concatenated data plex of 204800 sectors (100 MB).

- Verify there is enough space in the same disk group to mirror the volume:

```
# voldg [-g disk_group] free
```

- Add a new plex with the characteristics you want to the volume.

For example, to convert the volume `volC` to a striped volume, add a striped plex to the volume, specifying the number of columns and, optionally, the disks you want LSM to use:

```
# volassist -g dg1 mirror volC layout=stripe \  
ncolumn=2 [disk...]
```

The volume now looks similar to the following, with the original concatenated plex and the new striped plex:

```
# volprint -g dg1 -ht volC
```

```
Disk group: dg1
```

V NAME	USETYPE	KSTATE	STATE	LENGTH	READPOL	PREFPLEX		
PL NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WID	MODE	
SD NAME	PLEX	DISK	DISKOFFS	LENGTH	[COL/]OFF	DEVICE	MODE	
v volC	fsgen	ENABLED	ACTIVE	204800	SELECT	volC-02		
pl volC-01	volC	ENABLED	ACTIVE	204800	CONCAT	-	RW	
sd dsk2-01	volC-01	dsk2	0	204800	0	dsk2	ENA	
pl volC-02	volC	ENABLED	ACTIVE	204800	STRIPE	2/128	RW	
sd dsk3-01	volC-02	dsk3	0	102400	0/0	dsk3	ENA	
sd dsk5-01	volC-02	dsk5	0	102400	1/0	dsk5	ENA	

- Remove and dissociate the original plex from the volume:

```
# volplex [-g diskgroup] -o rm dis old_plex
```

For example, to remove the original concatenated plex `volC-01` from `volC`:

```
# volplex -g dg1 -o rm dis volC-01
```

If the volume has only one plex and you add another plex with the `volassist mirror` command, the new plex will have the same layout as the current plex (subject to the free disk space in the disk group and other restrictions).

If the volume still has plexes with the original layout type, repeat the process of adding new plexes with the layout you want and removing the original plexes until all plexes use the new layout.

You can also add a Dirty Region Log (Section 5.5.3) to the volume if it does not have one.

5.5.8.2 Changing the Plex Layout from Striped to Concatenated

To change the plex layout of a volume from striped to concatenated:

1. Display the size of the volume whose plex layout you want to change:

```
# volprint [-g disk_group] -ht volume
```

Disk group: dg1

V NAME	USETYPE	KSTATE	STATE	LENGTH	READPOL	PREFPLEX		
PL NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WID	MODE	
SD NAME	PLEX	DISK	DISKOFFS	LENGTH	[COL/]OFF	DEVICE	MODE	
v vols	fsgen	ENABLED	ACTIVE	204800	SELECT	vols-01		
pl vols-01	vols	ENABLED	ACTIVE	204800	STRIPE	4/128	RW	
sd dsk10-01	vols-01	dsk10	0	51200	0/0	dsk10	ENA	
sd dsk11-01	vols-01	dsk11	0	51200	1/0	dsk11	ENA	
sd dsk12-01	vols-01	dsk12	0	51200	2/0	dsk12	ENA	
sd dsk14-01	vols-01	dsk14	0	51200	3/0	dsk14	ENA	

In this example, the volume `vols` has one striped data plex of 204800 sectors (100 MB).

2. Verify there is enough space in the same disk group to mirror the volume:

```
# voldg [-g disk_group] free
```

3. Add a new plex with the characteristics you want to the volume.

For example, to convert the volume `vols` to a concatenated volume, add a concatenated mirror to the volume, optionally specifying the disks you want LSM to use:

```
# volassist -g dg1 mirror vols layout=nostripe [disk...]
```

The volume now looks similar to the following, with the original striped plex and the new concatenated plex:

```
# volprint -g dg1 -ht vols
```

Disk group: dg1

V NAME	USETYPE	KSTATE	STATE	LENGTH	READPOL	PREFPLEX		
PL NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WID	MODE	
SD NAME	PLEX	DISK	DISKOFFS	LENGTH	[COL/]OFF	DEVICE	MODE	
v vols	fsgen	ENABLED	ACTIVE	204800	SELECT	vols-01		
pl vols-01	vols	ENABLED	ACTIVE	204800	STRIPE	4/128	RW	
sd dsk10-01	vols-01	dsk10	0	51200	0/0	dsk10	ENA	
sd dsk11-01	vols-01	dsk11	0	51200	1/0	dsk11	ENA	
sd dsk12-01	vols-01	dsk12	0	51200	2/0	dsk12	ENA	
sd dsk14-01	vols-01	dsk14	0	51200	3/0	dsk14	ENA	
pl vols-02	vols	ENABLED	ACTIVE	204800	CONCAT	-	RW	
sd dsk19-01	vols-02	dsk19	0	204800	0	dsk19	ENA	

4. Remove and dissociate the original plex from the volume:

```
# volplex [-g disk_group] -o rm dis old_plex
```

For example, to remove the original striped plex `vols-01` from `vols`:

```
# volplex -g dg1 -o rm dis vol5-01
```

If the volume has only one plex and you add another plex with the `volassist mirror` command, the new plex will have the same layout as the current plex (subject to the free disk space in the disk group and other restrictions).

If the volume still has plexes with the original layout type, repeat the process of adding new plexes with the layout you want and removing the original plexes until all plexes use the new layout.

You can also add a Dirty Region Log (Section 5.5.3) to the volume if it does not have one.

5.6 Managing Subdisks

The following sections describe how to use LSM commands to manage subdisks.

5.6.1 Displaying Subdisk Information

- To display general information for all subdisks:

```
# volprint -st
Disk group: rootdg

SD NAME      PLEX      DISK  DISKOFFS  LENGTH  [COL/]OFF  DEVICE  MODE
sd dsk2-01   vol_mir-01 dsk2   0         256     0         dsk2    ENA
sd dsk3-02   vol_mir-03 dsk3   0         65      LOG        dsk3    ENA
sd dsk3-01   vol_mir-02 dsk3   65        256     0         dsk3    ENA
sd dsk4-01   p1         dsk4   17        500     0         dsk4    ENA
sd dsk4-02   p2         dsk4   518       1000    0         dsk4    ENA
```

- To display detailed information about a specific subdisk:

```
# volprint -l subdisk
For example:
# volprint -l dsk12-01
Disk group: rootdg

Subdisk:  dsk12-01
info:    disk=dsk12 offset=0 len=2560
assoc:   vol=vol5 plex=vol5-02 (offset=0)
flags:   enabled
device:  device=dsk12 path=/dev/disk/dsk12g diskdev=82/838
```

5.6.2 Joining Subdisks

You can join two or more subdisks to form a single, larger subdisk. Subdisks can be joined only if they belong to the same plex and occupy adjacent regions of the same disk. For a volume with striped plexes, the subdisks

must be in the same column. The joined subdisk can have a new subdisk name or retain the name of one of the subdisks being joined.

To join subdisks:

```
# volsd join subdisk1 subdisk2 new_subdisk
```

5.6.3 Splitting Subdisks

You can divide a subdisk into two smaller subdisks. After the subdisk is split, you can move the data in the smaller subdisks to different disks. This is useful for reorganizing volumes or for improving performance. The new, smaller subdisks occupy adjacent regions within the same region of the disk that the original subdisk occupied.

You must specify a size for the first subdisk; the second subdisk consists of the rest of the space in the original subdisk.

If the subdisk to be split is associated with a plex, both of the resultant subdisks are associated with the same plex. You cannot split a log subdisk.

- To split a subdisk and assign each subdisk a new name:

```
# volsd -s size split original_subdisk \  
new_subdisk1 new_subdisk2
```

- To split a subdisk and retain the original name for the first subdisk and assign a new name to the second subdisk:

```
# volsd -s size split original_subdisk new_subdisk
```

5.6.4 Moving Subdisks to a Different Disk

You can move the data in subdisks to a different disk to improve performance. The disk space occupied by the data in the original subdisk is returned to the free space pool.

Ensure that the following conditions are met before you move data in a subdisk:

- Both source and destination subdisks must be the same size.
- The source subdisk must be part of an active plex on an active volume.
- The destination subdisk must not be associated with any other plex.

To move data from one subdisk to another:

1. Display the size of the subdisk you want to move:

```
# volprint subdisk
```

For example:

```
# volprint -l dsk20-01
```

```

Disk group: dg1

Subdisk: dsk20-01
info:    disk=dsk20 offset=0 len=204800
assoc:   vol=volS plex=volS-01 (offset=0)
flags:   enabled
device:  device=dsk20 path=/dev/disk/dsk20g diskdev=81/1350

```

2. Create a new subdisk of the same size on a different disk:

```
# volmake [-g diskgroup] sd subdisk_name disk len=length
```

For example:

```
# volmake -g dg1 sd dsk11-01 dsk11 len=204800
```

3. Move the data in the old subdisk to the new subdisk:

```
# volsd mv source_subdisk target_subdisk
```

For example:

```
# volsd -g dg1 mv dsk20-01 dsk11-01
```

This leaves the old subdisk unassociated with any volume. If it is not needed, you can remove the subdisk (Section 5.6.5). Removing the subdisk returns that space to the free pool.

5.6.5 Removing Subdisks

You can remove a subdisk that is not associated with or needed by an LSM volume. Removing a subdisk returns the disk space to the free space pool in the disk group. To remove a subdisk, you must dissociate the subdisk from a plex or volume, then remove it.

To remove a subdisk:

1. Display information about the subdisk to identify any volume or plex associations:

```
# volprint -l subdisk
```

- If the subdisk is associated with a volume, the following information is displayed:

```

Disk group: rootdg

Subdisk: dsk9-01
info:    disk=dsk9 offset=0 len=2048
assoc:   vol=newVol plex=myplex (column=1 offset=0)
flags:   enabled
device:  device=dsk9 path=/dev/disk/dsk9g diskdev=82/646

```

- If the subdisk has no associations to any plex or volume, the following information is displayed:

```

Disk group: dg1

Subdisk: dsk20-01
info:    disk=dsk20 offset=0 len=204800

```

```
assoc:    vol=(dissoc) plex=(dissoc)
flags:    enabled
device:   device=dsk20 path=/dev/disk/dsk20g diskdev=81/1350
```

2. Do one of the following to remove the subdisk:

- If the subdisk is associated with a volume:

```
# volsd [-g disk_group] -o rm dis subdisk
```

- If the subdisk is not part of a volume and has no associations:

```
# voledit [-g disk_group] rm subdisk
```


6

Troubleshooting

LSM helps you protect the availability and reliability of data but does not prevent I/O failure. LSM is simply another layer added to the I/O subsystem. LSM depends on the underlying disk device drivers and system files to decide on the availability of individual disks and to manage and report any failures.

This chapter describes how to troubleshoot common LSM problems, describes tools that you can use to learn about problems, and offers possible solutions.

The hot-spare feature provides the best protection for volumes that use mirror plexes or a RAID 5 plex. When enabled, the hot-spare feature allows LSM to automatically relocate data from a failed disk in a volume that uses either a RAID 5 plex or mirrored plexes. LSM writes the data to a designated hot-spare disk, or to free disk space, and sends you mail about the relocation. For more information about enabling the hot-spare feature, see Section 3.5.1.

6.1 Troubleshooting LSM Objects

You can use LSM commands to monitor the status of LSM objects. By doing so, you can understand how LSM works under normal conditions and watch for indications that an LSM object might need attention before a problem arises.

6.1.1 Monitoring LSM Events

By default, LSM uses Event Manager (EVM) software to log events. The events that LSM logs are defined in the EVM template called `/usr/share/evm/templates/sys/lsm.volnotify.evt`.

You can select, filter, sort, format, and display LSM events using EVM commands or the graphical event viewer, which is integrated with the SysMan Menu and SysMan Station.

- To display a list of logged LSM events:

```
# evmget -f "[name *.volnotify]" | evmshow -t "@timestamp @@"
```

- To display LSM events in real time:

```
# evmwatch -f "[name *.volnotify]" | \
evmshow -t "@timestamp @@"
```

For more information, see EVM(5).

You can also display events directly with the `volnotify` command. For more information, see `volnotify(8)`.

6.1.2 Monitoring Read and Write Statistics

You can use the `volstat` command to view:

- The number of successful or failed read and write operations
- The number of blocks read from and written to
- The average time spent on read and write operations. This time reflects the total time it took to complete a read or write operation, including the time spent waiting in a queue on a busy device.

Note

In TruCluster environments, the `volstat` command reports statistic information only for the system where the command is entered. It does not provide aggregate statistic information for the whole TruCluster environment.

Table 6–1 describes some of the options you can use with the `volstat` command.

Table 6–1: Common `volstat` Command Options

Option	Displays
<code>-v</code>	Volume statistics
<code>-p</code>	Plex statistics
<code>-s</code>	Subdisk statistics
<code>-d</code>	LSM disk statistics
<code>-i seconds</code>	The specified statistics continuously in the interval specified (in seconds).

You can also reset statistics for a specific LSM object (such as a volume or a disk) or for all LSM objects.

For information on all the `volstat` options, see `volstat(8)`.

6.1.2.1 Displaying Read and Write Statistics

To display read and write statistics for LSM objects:

```
# volstat [-g disk_group] -vpsd [-i number_of_seconds]
```

TYP NAME	OPERATIONS		BLOCKS		AVG TIME(ms)	
	READ	WRITE	READ	WRITE	READ	WRITE
dm dsk6	3	82	40	62561	8.9	51.2

```

dm dsk7      0      725      0      176464      0.0      16.3
dm dsk9      688      37      175872      592      3.9      9.2
dm dsk10    29962      0      7670016      0      4.0      0.0
dm dsk12      0      29962      0      7670016      0.0      17.8
vol v1       3       72       40      62541      8.9      56.5
pl v1-01     3       72       40      62541      8.9      56.5
sd dsk6-01   3       72       40      62541      8.9      56.5
vol v2       0       37       0       592       0.0      10.5
pl v2-01     0       37       0       592       0.0      8.0
sd dsk7-01   0       37       0       592       0.0      8.0
sd dsk12-01  0       0       0       0       0.0      0.0
pl v2-02     0       37       0       592       0.0      9.2
sd dsk9-01   0       37       0       592       0.0      9.2
sd dsk10-01  0       0       0       0       0.0      0.0
pl v2-03     0       6        0       12       0.0      13.3
sd dsk6-02   0       6        0       12       0.0      13.3

```

6.1.2.2 Displaying Failed Read and Write Statistics

To display failed I/O statistics:

```
# volstat [-g disk_group] -f cf LSM_object
```

For example:

```
# volstat -f cf testvol
```

TYP NAME	CORRECTED		FAILED	
	READS	WRITES	READS	WRITES
vol testvol	1	0	0	0

LSM corrects read failures for mirror plexes or a RAID 5 plex, because these plexes provide data redundancy.

6.1.3 Monitoring LSM Object States

The kernel and LSM monitor the state of LSM objects.

To display the state of LSM volumes, plexes, and subdisks:

```
# volprint -vps
```

To display the state of LSM volumes:

```
# volprint -vt
```

```
Disk group: rootdg
```

V NAME	USETYPE	KSTATE	STATE	LENGTH	READPOL	PREFPLEX
v ka1	fsgen	ENABLED	ACTIVE	2097152	SELECT	-
v ka2	fsgen	ENABLED	ACTIVE	2097152	SELECT	-
v ka3	fsgen	ENABLED	ACTIVE	2097152	SELECT	-
v ka4	fsgen	ENABLED	ACTIVE	2097152	SELECT	-
v rootvol	root	ENABLED	ACTIVE	524288	ROUND	-
v swapvol	swap	ENABLED	ACTIVE	520192	ROUND	-
v vol-dsk25g	fsgen	ENABLED	ACTIVE	2296428	SELECT	-
v vol-dsk25h	fsgen	ENABLED	ACTIVE	765476	SELECT	-

```
Disk group: dgl
```

V	NAME	USETYPE	KSTATE	STATE	LENGTH	READPOL	PREFPLEX
v	vols	fsgen	ENABLED	ACTIVE	204800	SELECT	-

The **KSTATE** column shows the kernel state of the LSM object. The **STATE** column shows the LSM state of the LSM object.

6.1.3.1 Overview of LSM Kernel States

The LSM kernel state indicates the accessibility of the LSM object as viewed by the kernel. Table 6–2 describes kernel states for LSM objects.

Table 6–2: LSM Volume Kernel States (KSTATE)

Kernel State	Means
ENABLED	The LSM object is accessible and read and write operations can be performed.
DISABLED	The LSM object is not accessible.
DETACHED	Read and write operations cannot be performed, but device operations are accepted.

6.1.3.2 Overview of LSM Object States

LSM monitors the states of volumes, plexes, and subdisks.

- Table 6–3 describes the LSM volume states. The meaning of some volume states differs depending on the kernel state (KSTATE).
- Table 6–4 describes the plex states.
- Table 6–5 describes the subdisk states.

Table 6–3: LSM Volume States (STATE)

State	Kernel State	Means
EMPTY	DISABLED	The volume contents are not initialized.
CLEAN	DISABLED	The volume is not started. <ul style="list-style-type: none"> • For mirrored volumes, plexes are synchronized. • For RAID 5 volumes, parity is good and stripes are consistent.
ACTIVE	ENABLED	The volume was started or was in use when the system was restarted.
ACTIVE	DISABLED	RAID 5 parity synchronization is not guaranteed or mirror plexes are not guaranteed to be consistent.

Table 6–3: LSM Volume States (STATE) (cont.)

State	Kernel State	Means
SYNC	ENABLED	The system is resynchronizing mirror plexes or RAID 5 parity.
SYNC	DISABLED	The system was in the process of resynchronizing mirror plexes or RAID 5 parity when the system restarted and therefore the volume still needs to be synchronized.
NEEDSYNC	—	The volume requires a resynchronization operation the next time it starts.
REPLAY	—	A RAID 5 volume is in a transient state as part of a log replay. A log replay occurs when it is necessary to reconstruct data using parity and data.

Some plex states are transient; that is, the plex is in a particular state temporarily, usually while being attached and synchronized with a volume.

Table 6–4: LSM Plex States

State	Means
EMPTY	The plex is not initialized. This state is also set when the volume state is EMPTY.
CLEAN	The plex was running normally when the volume was stopped. The plex was enabled without requiring recovery when the volume was started.
ACTIVE	The plex is running normally on a started volume.
LOG	The plex is a DRL or RAID 5 log plex for the volume.
STALE	The plex was detached, either by the <code>volplex det</code> command or by an I/O failure. STALE plexes are reattached automatically by <code>volplex att</code> when a volume starts.
OFFLINE	The plex was disabled explicitly by the <code>volmend off</code> operation.
IOFAIL	The <code>void</code> daemon places an ACTIVE plex in the IOFAIL state when it detects an error. The plex is disqualified from the recovery selection process at volume start time, ensuring that LSM uses only valid plexes for recovery. A plex marked IOFAIL is recovered if possible during a resynchronization.
SNAPATT	This is a snapshot plex being attached by the <code>volassist snapstart</code> command. When the attach is complete, the state for the plex changes to SNAPDONE. If the system fails before the attach completes, the plex and all of its subdisks are removed.
SNAPDONE	This is a fully attached snapshot plex created by the <code>volassist snapstart</code> command. You can turn a plex in this state into a snapshot volume with the <code>volassist snapshot</code> command. If the system fails before the attach completes, the plex and all of its subdisks are removed.
SNAPTMP	This is a snapshot plex being attached by the <code>volplex snapstart</code> command. When the attach is complete, the state for the plex changes to SNAPDIS. If the system fails before the attach completes, the plex is dissociated from the volume.
SNAPDIS	This is a fully attached snapshot plex created by the <code>volplex snapstart</code> command. You can turn a plex in this state into a snapshot volume with the <code>volplex snapshot</code> command. If the system fails before the attach completes, the plex is dissociated from the volume.
TEMP	The plex is being associated and attached to a volume with the <code>volplex att</code> command. If the system fails before the attach completes, the plex is dissociated from the volume.

Table 6–4: LSM Plex States (cont.)

State	Means
TEMPRM	The plex is being associated and attached to a volume with the <code>volplex att</code> command. If the system fails before the attach completes, the plex is dissociated from the volume and removed. Any subdisks in the plex are kept.
TEMPRMSD	The plex is being associated and attached to a volume with the <code>volplex att</code> command. If the system fails before the attach completes, the plex and its subdisks are dissociated from the volume and removed.

Table 6–5: LSM Subdisk States

State	Means
REMOVED	The subdisk (which might encompass the entire LSM disk) was removed from the volume, disk group, or from LSM control.
RECOVER	The subdisk is stale and must be recovered. Use the <code>volrecover</code> command.
RELOCATE	The subdisk has failed in a redundant (mirrored or RAID 5) volume. The <code>volwatch</code> daemon checks for this state to identify which subdisks need to be relocated to available hot-spare disks. After the data is relocated, the subdisk state is cleared.

6.2 Troubleshooting a Missing or Altered `sysconfigtab` File

During the boot disk encapsulation procedure, LSM adds the following entries to the `/etc/sysconfigtab` file to enable the system to boot from the LSM root volume:

```
lsm:  
lsm_rootdev_is_volume=1
```

If this file is deleted or the LSM-specific entries are deleted, the system will not boot. If this happens, do the following:

1. Boot the system interactively:

```
>>> boot -fl i  
:  
Enter kernel_name option_1... option_n: vmunix
```

2. Replace the LSM entries in the `/etc/sysconfigtab` with the following:

```
lsm:  
lsm_rootdev_is_volume=1
```

6.3 Troubleshooting LSM Startup and Command Problems

LSM requires that the `vold` and `voliod` daemons be running. These daemons are normally started automatically when the system boots. If these daemons are not running, the most obvious problems you might notice are that LSM commands fail to complete or do not respond as expected, which is an indication that LSM did not correctly start up.

The following sections describe how to determine whether the daemons are running and how to correct problems.

6.3.1 Checking the Volume Configuration Daemon (`vold`)

To determine the state of the volume (`vold`) daemon:

```
# voldctl mode
```

Table 6–6 lists the possible output of the `voldctl mode` command, what the output means, and the commands to enter if the `vold` daemon is disabled or not running.

Table 6–6: `vold` Status and Solutions

Command Output	Status	Enter
Mode: enabled	Running and enabled	—
Mode: disabled	Running but disabled	<code>voldctl enable</code>
Mode: not-running	Not running	<code>vold</code>

For more information, see `vold(8)`.

6.3.2 Restarting the Volume Configuration Daemon (`vold`)

If the volume configuration daemon (`vold`) stops running on a system or cluster member, you might see the message “Configuration daemon is not accessible” if you try to run an LSM command. You can attempt to restart the daemon. The `vold` daemon might stop running for varied and unpredictable reasons, such as errors that cause it to dump core.

The `vold` daemon manages changes you make to the LSM configuration, such as adding LSM disks, creating disk groups and volumes, adding or removing logs, and so on, but does not have any effect on the accessibility of LSM volumes. File systems and other applications that use LSM volumes for their underlying storage should not be affected by the temporary failure of the `vold` daemon. However, until the `vold` daemon is restarted, you cannot make configuration changes or use informational commands, such as `volprint` and `volstat`, which make use of the daemon to get and display the current configuration data.

Each member of a cluster runs its own `vold` daemon. Check the daemon status on all other running members (Section 6.3.1) and perform the following procedure on each member as necessary.

To restart the `vold` daemon on a standalone system or a cluster:

1. Reset the `vold` daemon and start it in disabled mode:

```
# vold -k -r reset -m disable
```

This command stops any `vold` process that is currently running (or hung) and starts a new `vold` in disabled mode.

If any volumes are in use, the `-r reset` option fails. In this case, identify the open volumes and stop them temporarily, then retry the command.

2. Restart the daemon:

```
# vold -k
```

3. Restart any volumes you had to stop.

If this procedure does not restart `vold` or if you subsequently try to run commands such as `volprint` and get different error messages, please contact your customer service representative.

6.3.3 Checking the Volume Extended I/O Daemon (`voliiod`)

The correct number of `voliiod` daemons automatically start when LSM starts. Typically several `voliiod` daemons are running at all times. The default is at least one `voliiod` daemon for each processor on the system or a minimum of two.

To display the number of the `voliiod` daemons running:

```
# voliiod
2 volume I/O daemons running
```

This is the only method for displaying `voliiod` daemons, because the `voliiod` processes are kernel threads and are not listed in the output of the `ps` command.

If no `voliiod` daemons are running or if you want to change the number of daemons, enter the following command where *n* is the number of I/O daemons to start:

```
# voliiod set n
```

Set the number of LSM I/O daemons to two or the number of central processing units (CPUs) on the system, whichever is greater.

For more information, see `voliiod(8)`.

6.4 Troubleshooting LSM Disks

The following sections describe troubleshooting procedures for failing and failed disks, including the boot disk.

6.4.1 Checking Disk Status

Disks can experience transient errors for a variety of reasons, such as when a power supply suffers a surge or a cable is accidentally unplugged. You can verify the status of disks through the output of the `volprint` and `voldisk` commands.

To see the status of an LSM disk:

```
# voldisk list
```

To verify the usability of an LSM disk:

```
# voldisk check disk
```

For example:

```
# voldisk check dsk5  
dsk5: Okay
```

The `voldisk` command validates the usability of the specified disks by testing whether LSM can read and write the disk header information. A disk is considered usable if LSM can write and read back at least one of the disk headers that are stored on the disk. If a disk in a disk group is found to be unusable, it is detached from its disk group and all subdisks stored on the disk become invalid until you replace the physical disk or reassign the disk media names to a different physical disk.

Caution

Because an LSM `nopriv` disk does not contain a disk header, a failed `nopriv` disk might incorrectly be reported as usable.

6.4.2 Recovering Stale Subdisks

Stale subdisks have a state of `RECOVER` (Table 6–5). LSM usually recovers stale subdisks when the volume starts. However, it is possible that:

- The recovery process might get killed.
- The volume might be started with an option to defer subdisk recovery.
- The disk supporting the subdisk might have been replaced without any recovery operations being performed.

To recover a stale subdisk in a volume:

```
# volrecover [-sb] volume
```

To recover all stale subdisks on an LSM disk:

```
# volrecover [-sb] disk
```

6.4.3 Recovering From Temporary Disk Failures

If a disk had a temporary failure but is not damaged (for example, the disk was removed by accident, a power cable was disconnected, or some other recoverable problem occurred that did not involve restarting the system), you can recover the volumes on that disk.

To recover from a temporary disk failure:

1. Make sure the disk is back on line and accessible; for example:
 - Confirm that the disk is firmly snapped into the bay.
 - Reconnect any loose cables.
 - Perform any other checks appropriate to your system.
2. Scan for all known disks to ensure the disk is available:

```
# voldctl enable
```

3. Recover the volumes on the disk:

```
# volrecover -sb disk
```

6.4.4 Moving LSM Volumes Off a Failing Disk

Often a disk has recoverable (soft) errors before it fails completely. If a disk is experiencing an unusual number of soft errors, move the volume from the disk to a different disk in the disk group and replace the failing disk.

Note

To replace a failed boot disk, see Section 6.4.6.

To move a volume off a failing disk:

1. Identify the size of the volume:

```
# volprint [-g disk_group] -ht [volume]
```

2. Ensure there is an equal amount of free space in the disk group:

```
# voldg [-g disk_group] free
```

If there is not enough space, add a new disk. For more information, see Section 5.2.2.

3. Move the volume to a disk other than the failing disk, as specified by the `!` operand. Use the appropriate shell quoting convention to correctly interpret the `!`. You do not need to specify a target disk.

```
# volassist [-g disk_group] move volume !disk
```

For more information on replacing a failed disk, see Section 6.4.5.

6.4.5 Replacing Failed Disks

When an LSM disk fails completely, its state becomes detached. For best results, replace the failed disk with the same size disk.

If hot-sparing was enabled at the time of the disk failure and successfully relocated redundant data, you do not need to follow the procedure below. Instead, you can initialize a new disk for LSM and optionally move the data from the hot-spare disk to the new disk (Section 5.1.6), or you can configure the new disk as a hot-spare disk (Section 3.5.1).

To replace a failed boot disk, see Section 6.4.6.

Note

If the failed disk is part of a hardware stripeset or RAIDset that you set up with a particular chunk size, recreate the same attributes on the replacement disk. For more information on LSM disks comprised of hardware sets, see the Best Practice entitled *Aligning LSM Disks and Volumes to Hardware RAID Devices* at the following URL:

```
http://www.tru64unix.compaq.com/docs/best\_practices/sys\_bps.html
```

To replace a failed disk:

1. Identify the disk media name of the failed disk using one of the following commands:
 - To display all disk, disk group, and volume information and the status of any volumes that are affected by the failed disk:

```
# volprint -Aht
```
 - To display only disk information:

```
# volprint -Adt
```
2. Remove the failed disk media name from its disk group, using the `-k` option to retain the disk media name:

```
# voldg [-g disk_group] -k rmdisk disk_media_name
```


3. Remove the failed disk access name from LSM control:

```
# voldisk rm disk_access_name
```

You must completely remove the device from LSM before running any non-LSM commands to remove and replace the failed disk, such as `hwmgr redirect`.

4. Physically remove the failed disk and replace it with a new disk.
5. Scan for the new disk:

```
# hwmgr scan scsi
```

The `hwmgr` command returns the prompt before it completes the scan. Confirm that the system has discovered the new disk before continuing, for example, by entering the `hwmgr show scsi` command until you see the new device.

6. Label and initialize the new disk, using one of the following procedures.
 - If you saved a copy of the disk label information for the failed disk when you initialized the disk originally (see Section 4.1.5 for more information), and you want to apply that disk label to the new disk, do the following:

- a. Apply the backup disk label to the new disk:

```
# disklabel -R disk_access_name file
```

- b. Initialize the disk for LSM as the same LSM disk type (simple, sliced, or nopriv) as the failed disk, with the same public or private region offset (if applicable) as the failed disk.

Specify the disk access name in the form of either `diskn` (for the entire disk) or `disknp` (for a partition of the disk):

- For a nopriv disk:

```
# voldisk -f init disk_access_name
```

- For a sliced disk:

```
# voldisk -f init disk_access_name \  
[puboffset=16]
```

Make sure the `puboffset` you specify is the same as it was on the failed disk.

- For a simple disk on a partition that begins at block 0 of the disk (for example, the `a` or `c` partition):

```
# voldisk -f init disk_access_name \  
privoffset=16
```

Make sure the `privoffset` you specify is the same as it was on the failed disk.

- For a simple disk on a partition that does not begin at block 0 of the disk (for example, the `b` or `d` partition):

```
# voldisk -f init disk_access_name
```

- If you do not have a backup disk label from the failed disk or if you want to initialize the new disk with default values, do the following:

- If the new disk is part of a hardware stripeset or raidset that you recreated, clear the old disk label:

```
# disklabel -z disk_access_name
```

- Apply a default disk label to the new disk:

```
# disklabel -rwn disk_access_name
```

- Initialize the new disk for LSM as a default sliced disk:

```
# voldisksetup -i disk
```

- Optionally (but recommended), create a backup copy of the new disk's disk label information:

```
# disklabel disk_access_name > file
```

- Add the new disk to the applicable disk group, assigning the old disk media name to the new disk:

```
# voldg [-g disk_group] -k adddisk \  
disk_media_name=disk_access_name
```

For example, if the failed disk media name was `dsk10` and the new disk access name is `dsk82`, and the disk group is `dg03`:

```
# voldg -g dg03 -k adddisk dsk10=dsk82
```

- Start recovery on all applicable LSM volumes:

```
# volrecover [-sb]
```

This command initiates plex attach operations, RAID 5 subdisk recovery, and resynchronization for all volumes requiring recovery, and resolves most of the problems resulting from a failed disk.

If this does not recover the volumes affected by the disk failure (for example, nonredundant volumes or volumes that had multiple disk failures), see Section 6.5.2 for information on recovering volumes, and Section 5.4.3 for information on restoring volumes from backup.

6.4.6 Replacing Failed Boot Disks

When the boot disk on a standalone system is encapsulated to an LSM volume with mirror plexes, failures occurring on the original boot disk are transparent to all users. However, during a failure, the system might:

- Write a message to the console indicating there was an error reading or writing to one plex
- Experience slow performance (depending on the problem encountered with the disk containing one of the plexes in the root or swap volumes)

If necessary, before you replace the original boot disk, you can restart the system from any disk that contains a valid plex for the `rootvol` volume. If all plexes in the `rootvol` volume are corrupted and you cannot boot the system, you must reinstall the operating system.

The following procedure requires an encapsulated the boot disk on a standalone system and mirror plexes for the boot disk volumes. Step 4 in this procedure creates a new (replacement) mirror on the new disk.

To replace a failed boot disk under LSM control with a new disk:

1. Display detail information about the root and swap volumes to ensure you use the name of the failed disk and failed plex in the following steps:

```
# volprint -vht
```

2. In the output, identify the name of the failed plex or plexes and the disk media name of the failed LSM disk or disks.

When you encapsulate the boot disk, LSM assigns special disk media names to the in-use partitions on the boot disk. In the following output from the `voldisk list` command, the original root disk is `dsk14`. The disk used to mirror the root and swap volumes is `dsk15`.

DEVICE	TYPE	DISK	GROUP	STATUS
dsk14a 1	nopriv	root01 2	rootdg	online
dsk14b 3	nopriv	swap01 4	rootdg	online
dsk14f 5	simple	dsk14f	rootdg	online
dsk14g 6	nopriv	dsk14g-AdvFS 7	rootdg	online
dsk14h 8	nopriv	dsk14h-AdvFS 9	rootdg	online
dsk15a	nopriv	root02	rootdg	online
dsk15b	nopriv	swap02	rootdg	online
dsk15f	simple	dsk15f	rootdg	online
dsk15g	nopriv	dsk15g-AdvFS	rootdg	online
dsk15h	nopriv	dsk15h-AdvFS	rootdg	online

The following list describes the output:

- 1** Disk access name for the root (/) partition.
- 2** Disk media name for the root (/) partition.
- 3** Disk access name for the primary swap partition.
- 4** Disk media name for the primary swap partition.
- 5** Disk access name for the LSM private region for the boot disk (same as its disk media name).
- 6** Disk access name for the /usr partition.

- 7 Disk media name for the `/usr` partition.
- 8 Disk access name for the `/var` partition.
- 9 Disk access name for the `/var` partition.

The same naming conventions apply to the disk used to mirror the root, swap, `/usr`, and `/var` partition volumes.

3. Dissociate the plexes on the failed disk from the root, swap, and user volumes, if `/usr` or `/var` were encapsulated on the boot disk.

For example:

```
# volplex -o rm dis rootvol-02 swapvol-02 vol-dsk0g-02
```

The `/usr`, and if separate, `/var` volume names are derived from the partition letter of the boot disk (for example, `vol-dsk0g`).

4. Remove the failed LSM disks for the boot disk:

- a. Remove the disks from the `rootdg` disk group:

```
# voldg rmdisk dskna dsknb dskng...
```

- b. Remove the LSM disks configured on the boot disk from LSM control:

```
# voldisk rm dskna dsknb dskng...
```

5. Physically remove and replace the failed disk.

You must completely remove the device from LSM before running any non-LSM commands to remove and replace the failed disk, such as `hwmgr redirect`.

6. Scan for the new disk:

```
# hwmgr scan scsi
```

The `hwmgr` command returns the prompt before it completes the scan. You need to confirm that the system has discovered the new disk before continuing, such as by entering the `hwmgr show scsi` command until you see the new device.

7. Modify the device special files, reassigning the old disk name to the new disk. Make sure you list the new disk first.

```
# dsfmgr -e new_name old_name
```

8. Label the new disk, setting all partitions to unused:

```
# disklabel -rw new_disk
```

9. Mirror the root volumes to the new disk:

```
# volrootmir -a new_disk
```

6.5 Troubleshooting LSM Volumes

The following sections describe how to solve common LSM volume problems. Alert icons and the Alert Monitor window might provide information when an LSM volume recovery is needed. (For more information about the Alert Monitor, see the *System Administration* manual.)

6.5.1 Recovering LSM Volumes After a System Failure

LSM can usually recover volumes automatically after a system crash. Using a DRL plex on all mirrored volumes (except those used as swap space) and using a RAID 5 log plex on all RAID 5 volumes speeds volume recovery.

Fast Plex Attach logs also resume operation after a system restart, with the exception of an active FPA log on the `rootvol` or `cluster_rootvol` volumes. In this case, LSM disables FPA logging on both the primary and secondary volumes. A full resynchronization occurs if you return the migrant plex to the primary volume.

6.5.2 Recovering LSM Volumes After Disk Failure

Use the following volume recovery procedures (Table 6–7) as applicable to recover volume data after replacing a failed disk or disks (Section 6.4.5).

Note

If the disk failures resulted in the loss of all active copies of the LSM configuration database in a disk group, see Section 5.3.2 before recovering the volumes.

Table 6–7: Volume Recovery Procedures

To Recover:	See:
Mirrored and RAID 5 volumes and logs after a single disk failure	Section 6.5.2.1
Nonredundant volumes	Section 6.5.2.2
Mirrored volumes with no valid (or known) plexes	Section 6.5.2.3
Mirrored volumes with one known valid plex	Section 6.5.2.4
RAID 5 plex after multiple disk failures	Section 6.5.2.5
RAID 5 log plex	Section 6.5.2.6

6.5.2.1 Recovering Mirrored and RAID 5 LSM Volumes

You can recover an LSM volume that has become disabled. Recovering a disabled LSM volume starts the volume and, if applicable, resynchronizes mirror plexes or RAID 5 parity.

If a redundant (mirrored or RAID 5) volume experienced a single disk failure in a data plex or a log plex, you can recover the volume with the `volrecover` command, which takes one of the following actions, appropriate to the situation:

- Starts plex resynchronization in a mirrored volume
- Initiates data and parity regeneration in a RAID 5 volume
- Reattaches a detached DRL or RAID 5 log plex

If hot-sparing is enabled, you might not need to do anything if suitable disk space was available for relocation at the time of the disk failure. For a RAID 5 log plex, relocation occurs only if the log plex is mirrored. If hot-sparing was disabled at the time of a failure, you might need to initiate the recovery.

To recover an LSM volume, enter the following command, specifying either a volume or a disk, if several volumes use the same disk:

```
# volrecover [-g disk_group] -sb volume|disk
```

The `-s` option immediately starts the volume but delays recovery, and the `-b` option runs the command in the background. (For more information on these and other options, see `volrecover(8)`.) If you do not specify a disk group, LSM volume name, or disk name, all volumes are recovered. If recovery of an LSM volume is not possible, restore the LSM volume from backup (Section 5.4.3).

For example, to recover an LSM volume named `vol101` in the `rootdg` disk group:

```
# volrecover -sb vol101
```

To recover all LSM objects (subdisks, plexes, or volumes) that use disk `disk5`:

```
# volrecover -sb disk5
```

Optionally, verify the volume is recovered (or that recovery is underway):

```
# volprint volume
```

6.5.2.2 Recovering a Nonredundant Volume

Nonredundant volumes are those with a single concatenated or striped plex. If a disk in the plex fails, the volume will be unstartable, and you must restore the volume's data from backup. Volumes can also become

nonredundant if disk failures happen in multiple plexes or in multiple columns of a RAID 5 plex.

You can display the volume's condition:

```
# volinfo -p
vol  tst          fsgen      Unstartable
plex tst-01      NODEVICE
```

Note

In the following procedure, assume the disk is still usable or has been replaced (Section 6.4.5).

To recover the volume:

1. Set the plex state to stale:

```
# volmend fix stale plex
```

LSM has internal state restrictions that require a plex to change states in a specific order. A plex state must be stale before it can be set to clean.

2. Set the plex state to clean:

```
# volmend fix clean plex
```

3. Start the volume:

```
# volume start volume
```

The volume is now running and usable but contains invalid data.

4. Do one of the following:

- **If the volume was used by a file system**, recreate the file system on the volume, and mount the file system. For more information on configuring a volume for a file system, see Section 4.5.
- **If you have a backup of the data**, restore the volume using the backup. For more information on restoring a volume from backup, see Section 5.4.3.
- **If you have no backup and the volume was used by an application such as a database**, see that application's documentation for information on restoring or recreating the data.

6.5.2.3 Recovering a Mirrored Volume with No Valid Plexes

If disks fail in all data plexes in a mirrored volume, the volume will be unstartable. If disks fail in several, but not all, data plexes, the volume's data might be corrupt or suspect. To recover a mirrored volume after multiple disk failures, you must restore the data from backup.

To recover a volume with no valid plexes:

1. Set the plex state for all the plexes in the volume to clean:

```
# volmend fix clean plex plex...
```

2. Start the volume:

```
# volume start volume
```

3. Depending on what was using the volume, do one of the following:

- **If the volume was used by a file system**, recreate the file system on the volume, and mount the file system. For more information on configuring a volume for a file system, see Section 4.5.
- **If you have a backup of the data**, restore the volume using the backup. For more information on restoring a volume from backup, see Section 5.4.3.
- **If you have no backup and the volume was used by an application such as a database**, see that application's documentation for information on restoring or recreating the data.

6.5.2.4 Recovering a Mirrored Volume with One Valid Plex

If one plex in a volume contains known valid data, you can use that plex to restore the others.

To recover a volume with one valid data plex:

1. Set the valid data plex's state to clean:

```
# volmend fix clean valid_plex
```

2. Set the state of all the other data plexes to stale:

```
# volmend fix stale plex plex...
```

3. Start the volume and initiate the resynchronization process (optionally in the background):

```
# volrecover -s [-b] volume
```

6.5.2.5 Recovering a RAID 5 Plex from Multiple Disk Failures

Volumes that use a RAID 5 plex remain available when one disk fails.

However, if disks in two or more columns of a RAID 5 data plex fail, LSM cannot use the remaining data and parity to reconstruct the missing data. You must replace the failed disks, restart the volume, and then restore the data.

Note

In the following procedure, assume the failed disks have been replaced (Section 6.4.5).

To recover the volume:

1. Stop the volume:

```
# volume stop volume
```

2. Set the volume state to empty to force parity recalculation when the volume starts:

```
# volmend -f fix empty volume
```

3. Start the volume, optionally running the operation in the background to return the system prompt immediately:

```
# volume [-o bg] start volume
```

The volume is usable while parity recalculation proceeds. If an I/O request falls in a region that has not been recalculated, LSM recalculates and writes the parity for the entire stripe before honoring the I/O request.

4. Restore the volume data from backup (Section 5.4.3).

If you have no backup and the volume was used by an application such as a database, see that application's documentation for information on restoring or recreating the data.

6.5.2.6 Recovering a RAID 5 Log Plex

A disk failure in a RAID 5 log plex has no direct effect on the operation of the volume; however, the loss of all RAID 5 logs on the volume makes the volume vulnerable to data loss in the event of a system failure.

The following output from the `volprint` command shows a failure within a RAID 5 log plex. The plex state is `BADLOG`, and the RAID 5 log plex `vol5-02` has failed. (In some cases, RAID 5 log plexes might have a state of `DETACHED` due to disk failures.)

Disk group: rootdg

V NAME	USETYPE	KSTATE	STATE	LENGTH	READPOL	PREFPLEX	
PL NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WID	MODE
SD NAME	PLEX	DISK	DISKOFFS	LENGTH	[COL/]OFF	DEVICE	MODE
v vol5	raid5	ENABLED	ACTIVE	409696	RAID	-	
pl vol5-01	vol5	ENABLED	ACTIVE	409696	RAID	8/32	RW
sd dsk3-01	vol5-01	dsk3	0	58528	0/0	dsk3	ENA
sd dsk4-01	vol5-01	dsk4	0	58528	1/0	dsk4	ENA
sd dsk5-01	vol5-01	dsk5	0	58528	2/0	dsk5	ENA
sd dsk6-01	vol5-01	dsk6	0	58528	3/0	dsk6	ENA

```

sd dsk7-01      vol15-01      dsk7      0          58528     4/0       dsk7      ENA
sd dsk8-01      vol15-01      dsk8      0          58528     5/0       dsk8      ENA
sd dsk9-01      vol15-01      dsk9      0          58528     6/0       dsk9      ENA
sd dsk10-01     vol15-01     dsk10     0          58528     7/0       dsk10     ENA
pl vol15-02     vol15        DISABLED  BADLOG     2560      CONCAT    -         RW
sd dsk11-01     vol15-02     dsk11     0          2560      0         -         RMOV

```

If the disk has failed, replace it (Section 6.4.5).

To recover a RAID 5 log plex, reattach the log plex to the volume:

```
# volplex att volume log_plex
```

For example:

```
# volplex att vol15 vol15-02
```

6.5.3 Starting Disabled LSM Volumes

If you cannot mount a file system that uses an LSM volume, or if an application cannot open an LSM volume, the LSM volume might not be started.

To determine whether or not the LSM volume is started:

```
# volinfo [-g disk_group] volume
```

The following output shows the condition of several volumes:

```

vol  bigvol      fsgen      Startable
vol  vol2         fsgen      Started
vol  datavol     gen        Unstartable

```

LSM volumes can have the following conditions:

- **Started** – The volume is enabled and running normally.
- **Startable** – The volume is not enabled, and at least one plex has a state of **ACTIVE** or **CLEAN**, indicating that the volume can be restarted.

Note

Normally, volumes will not be in this state unless you manually created a volume (not using the `volassist` command, which starts the new volume automatically), or you did something that disabled the volume, such as removing a plex. All startable volumes are started when the system restarts.

- **Unstartable** – The volume is not enabled and has a problem (such as a disk failure) that you must resolve before you can start the volume.

To replace a failed disk, see Section 6.4.5.

To start a startable volume:

```
# volume [-g disk_group] start volume
```

6.5.4 Checking the Status of Volume Resynchronization

If the system fails and restarts, LSM automatically recovers all volumes that were running normally at the time of the failure.

- **For volumes that use mirror plexes and have a DRL plex**, this involves resynchronizing all the dirty regions.
- **For volumes that use a RAID 5 plex and have a RAID 5 log plex**, this involves replaying the log plex to complete any outstanding writes.

Using redundant volumes with log plexes is the recommended method to speed the recovery of volumes after a system failure. Under normal circumstances, the recovery happens so quickly that there is no noticeable effect (such as performance lag) after the system is running again. However, if a volume has no log, the resynchronization can take a long time (minutes to hours or longer) depending on the size of the volume.

You can display the status of the volume resynchronization in progress to determine how long it will take. (You cannot check the status of plex resynchronization, which occurs when you replace a failed disk or add a new plex to a volume; the `volprint` command does not have access to that information. However, in these cases, the volume is usable while the resynchronization occurs.)

To calculate the time remaining for a volume resynchronization in progress:

1. Display the read/write flags for the volume to see the current recovery offset value:

```
# volprint -v1 volume | grep flags
```

The following information is displayed:

```
flags:    open rback (offset=121488) writeback
```

2. Display the flags again after some time has passed (120 seconds is ample) to see how far the recovery has progressed:

```
# sleep 120 ; volprint -v1 volume | grep flags
```

The following information is displayed:

```
flags:    open rback (offset=2579088) writeback
```

3. Calculate the rate of progress by dividing the difference between the offsets by the time that passed between the two displays. For example, in 120 seconds the resynchronization had completed 2457600 sectors. Each second, approximately 20480 sectors (10 MB) were resynchronized.

4. Multiply the resynchronization rate by the size of the volume in sectors. This indicates the approximate amount of time a complete resynchronization will take. For example, at a rate of 20480 sectors per second, a volume that is 200 GB will take about five and a half minutes to resynchronize.

The actual time required can vary, depending on other I/O loads on the system and whether the volume or the system experiences additional problems or failures.

6.5.4.1 Changing the Rate of Future Volume Resynchronizations

Although you cannot change the rate of (or stop) a volume resynchronization after it has begun, you can change the setting for the rate of future resynchronizations, if your volumes are large enough that the resynchronization has a noticeable impact on system performance during recovery.

Caution

Use this procedure only if you are a knowledgeable system administrator and you have evaluated the effect of volume resynchronization on system performance and determined it to be unacceptable. You should be familiar with editing system files and scripts.

To change the rate of volume resynchronization for future recoveries, use your preferred editor to modify the indicated line in the `/sbin/lsm-startup` script.

Example 6–1 shows the relevant section of the script, which has been edited for brevity and formatting.

Example 6–1: Volume Recovery Section of `/sbin/lsm-startup` Script

```
#!/sbin/sh
:
volrecover_iosize=64k
:
if [ "X`/sbin/voldctl mode 2> /dev/null`" = "Xmode: enabled" ]; then
  /sbin/volrecover -b -o iosize=$volrecover_iosize -s 1
  if [ $is_cluster -eq 1 -a $vold_locked -eq 1 ]
    :
fi
```

1 Change the indicated line to one of the following:

- To slow the rate of recovery, add `-o slow`:

```
/sbin/volrecover -b -o iosize=$volrecover_iosize -o slow -s
```

The `-o slow` option inserts a delay of 250ms between each recovery operation. This can considerably reduce the performance impact on the system, depending on the size of the volume and the number of plexes.

- To disable resynchronization, add `-o delayrecover`:

```
/sbin/volrecover -b -o iosize=$volrecover_iosize -o delayrecover -s
```

The `-o delayrecover` option requires that you manually begin a resynchronization at your discretion, such as when the system is not under peak demand. Until then, the volume remains in read-writeback mode, which means that every time a region of the volume is read, the data is written to all plexes in the volume. When you eventually initiate the resynchronization, all regions marked dirty are resynchronized, perhaps unnecessarily.

This option incurs performance overhead by writing all reads back to all plexes, which might be less than the impact of permitting the resynchronization to complete during periods of high system demand.

You can change the `/sbin/lsm-startup` script back to its original state at any time.

6.5.5 Calculating Sufficient Space to Create LSM Volumes

When you use the `volassist` command to create a volume with a striped plex, you might receive an error message indicating there is insufficient space for the volume even though you know there is enough space available.

The `volassist` command rounds up the length you specify on the command line to a multiple of the data unit size of 64K bytes by default, or the stripe width you specified, and then divides the total by the number of disks available to make the column. The smallest disk in the disk group limits the data unit size.

For example, you have two disks with differing free space in the `dg1` disk group:

```
# voldg -g dg1 free
```

GROUP	DISK	DEVICE	TAG	OFFSET	LENGTH	FLAGS
dg1	dsk1	dsk1	dsk1	0	2049820	-
dg1	dsk2	dsk2	dsk2	0	2047772	-

The total free space on these two disks is 4097592. Suppose you tried to create a volume with a striped plex with a length of 4095544 blocks (about 2 GB), which is less than the total space available:

```
# volassist -g dg1 make NewVol 4095544 layout=stripe
```

```
volassist: adjusting length 4095544 to conform  
to a layout of 2 stripes 128 blocks wide  
volassist: adjusting length up to 4095744 blks
```

```
volassist: insufficient space for a 4095744 block long volume in stripe,  
contiguous layout
```

The command returned an error message indicating insufficient space, because `volassist` rounded up the length you specified to an even multiple of the data unit size of 64K bytes (128 blocks) and divided that number by the number of disks (2). The result was larger than the space available on the smaller disk: $4095744 \div 2 = 2048796$.

- If your volume does not need to be precisely the size you specified, you can retry the command with a length that works with the data unit size and the number of disks. For example, multiply the size of the smallest free space by the number of disks: $2047772 \times 2 = 4095488$. Use this value in the command line:

```
# volassist -g dg1 make NewVol 4095488 layout=stripe
```

- If the volume you require is larger than the total free space in the disk group, or if the volume must be exactly the size you specify, add more (or larger) disks to that disk group. For more information on adding disks to a disk group, see Section 5.2.2.

To determine whether the disk group has enough space on enough disks to create the volume you want, use the `volassist maxsize` command, specifying all the properties of the volume except the size.

For example, to determine whether you can create a volume with three mirrored, striped plexes and a stripe width of 128K bytes in the `dg1` disk group:

```
# volassist -g dg1 maxsize layout=stripe nmirror=3 \  
stwid=128k  
Maximum volume size: 16424960 (8020Mb)
```

You can additionally specify other properties, such as the number of stripe columns; for example:

```
# volassist -g dg1 maxsize layout=stripe nmirror=3 \  
stwid=128k ncolumn=3  
lsm:volassist: ERROR: No volume can be created within  
the given constraints
```

If you receive a message similar to the previous output, you can try the same command in a different disk group. For example, to determine whether you can create a volume with the same properties in the `rootdg` disk group:

```
# volassist maxsize layout=stripe nmirror=3 \  
stwid=128k ncolumn=3
```

Maximum volume size: 35348480 (17260Mb)

6.5.6 Clearing Locks on LSM Volumes

When LSM makes changes to an object's configuration, LSM locks the object until the change is written. If a configuration change terminated abnormally, there might still be a lock on the object.

To determine whether an object is locked:

```
# volprint [-g disk_group] -vh
```

In the information displayed, the lock appears in the TUTIL0 column.

To clear the lock:

```
# volmend [-g disk_group] clear tutil0 object...
```

You might need to restart the volume (Section 5.4.4).

6.6 Troubleshooting Disk Groups

If you receive an error message or the command fails when trying to import a disk group, possible causes are:

- One or more of the disks contains the host ID of another system (Section 6.6.1).
- One or more of the disks might be inaccessible (Section 6.6.2).
- The disk group contains multiple `nopriv` disks (Section 7.3.3).

6.6.1 Resolving Mismatched Host IDs

If a disk group was not deported cleanly from its original system, the disks in the disk group might contain a record of the original host ID. This can happen if the system crashed, you were unable to restart it, and you decided to move its storage to another system or if you disconnected the storage without first deporting the disk group. To import the disk group on the new system, clear the previous host ID.

To determine whether one or more disks contains the host ID of another system:

```
# voldisk list disk_access_name | grep hostid  
hostid:      potamus.zk3.dec.com
```

If the host ID of the disk does not match that of the system where you are trying to import the disk group, clear the previous host ID:

```
# voldisk clearimport disk_access_name
```

After you resolve the mismatched host IDs for all the disks in the disk group, you can import the disk group. If the disk group has the same name as an existing disk group on the new host system, rename the disk group as you import it. For more information, see `voldg(8)`.

6.6.2 Importing Disk Groups with Failed Disks (Forced Import)

To forcibly import a disk group:

```
# voldg -f import disk_group
```

After the disk group is imported, you can identify and solve the problem.

If you cannot import the disk group even with the force (`-f`) option, it might be because LSM cannot find a copy of the configuration database for that disk group. This is unlikely but can happen in the following situations:

- All the disks with active copies for that disk group failed.
Normally, when LSM detects the failure of a disk with an active copy, it enables a copy on another disk in the disk group. For all disks with active copies to fail simultaneously is rare but possible. This is more likely to happen if you configured a disk group to have fewer than the default number of copies or if the disk group contains many `nopriv` disks and few `sliced` or `simple` disks. Only `sliced` or `simple` disks can store copies of the configuration database; if these disks fail, LSM has nowhere to place an active copy.
- All the disks with active copies for that disk group are on the same bus (this is not the default) and the bus failed or are in the same RAID array (for example, the same HSG80) and that array failed or is inaccessible.
Both situations can also result in problems with volumes; for example, plex detachments, loss of DRL or RAID 5 logs, or total volume loss for nonmirrored volumes.

To restore the configuration database for a disk group, see Section 5.3.2.

Special Cases

The following topics are covered in this chapter:

- Upgrading a system or cluster with an LSM configuration from Tru64 UNIX Version 4.0 to Version 5.0 or higher (Section 7.1)
- Adding a system with LSM to a cluster (Section 7.2)
- Moving disk groups between systems (Section 7.3)
- Unencapsulating the boot disk (standalone system) (Section 7.4)
- Migrating AdvFS domains from LSM volumes to physical storage (Section 7.5)
- Unencapsulating cluster members' swap devices (Section 7.6)
- Uninstalling the LSM software (Section 7.7)

7.1 Upgrading an LSM Configuration

If you are currently using LSM on a system running Tru64 UNIX Version 4.0 and you want to preserve your current LSM configuration for use with Tru64 UNIX Version 5.0 or higher, you must:

1. Increase the size of any block-change logs (BCLs) to at least two blocks per gigabyte of volume size for a standalone system or at least 65 blocks per gigabyte of volume size for a TruCluster Server environment (Section 7.1.1).
2. Back up the current LSM configuration (Section 7.1.2).
3. Optionally, deport any disk groups that you do not want to upgrade (Section 7.1.3).
4. Upgrade the LSM software (Section 7.1.4).
5. Manually convert any Version 4.0 disk groups that you deported before the upgrade (Section 7.1.5).
6. Optimize the restored LSM configuration databases (Section 7.1.6).

7.1.1 Increasing the Size of BCLs

The block-change logging (BCL) feature supported in LSM in Tru64 UNIX Version 4.0 was replaced with the dirty-region logging (DRL) feature in Version 5.0.

When you perform an upgrade installation, BCLs are automatically converted to DRLs if the BCL subdisk is at least two blocks. If the BCL subdisk is one block, logging is disabled after the upgrade installation.

Note

The conversion of BCLs to DRLs is not reversible.

Before you upgrade, increase the size of the BCLs to at least two blocks per gigabyte of volume size for standalone systems or at least 65 blocks per gigabyte of volume size for a TruCluster environment. If this is not possible, then after the upgrade, you can add a new log to those volumes using `volassist addlog`, which creates a DRL of the appropriate size by default.

For information on increasing the size of BCLs, see the LSM documentation for your current operating system version.

7.1.2 Backing Up the LSM Configuration

Backing up the LSM configuration creates a file that describes all the LSM objects in all disk groups. In case of a catastrophic failure, LSM can use this file to restore the LSM configuration.

Caution

The following procedure backs up only the configuration, not the volume data. You might also want to back up the volume data before performing the upgrade.

To back up the LSM configuration:

1. Enter the following command:

```
# volsave [-d dir]
```

```
LSM configuration being saved to /usr/var/lsm/db/LSM.20020312143345.hostname
```

```
volsave does not save configuration for volumes used for  
root, swap, /usr or /var.
```

```
LSM configuration for following system disks not saved:  
dsk3 dsk0a dsk2a dsk0b dsk2b dsk0g dsk0g
```

```
LSM Configuration saved successfully to /usr/var/lsm/db/LSM.20020312143345.hostname
```

By default, LSM configuration information is saved to a time-stamped file called a description set in the `/usr/var/lsm/db` directory. Make a note of the location and name of the file. You will need this information to restore the LSM configuration after you upgrade the Tru64 UNIX operating system software.

2. Optionally, confirm that the LSM configuration was saved:

```
# ls /usr/var/lsm/db/LSM.date.hostname
dg1.d          newdg.d       volboot
header        rootdg.d      voldisk.list
```

3. Save the LSM configuration to tape or other removable media.

7.1.3 Deporting Disk Groups (Optional)

The internal metadata format of LSM in Tru64 UNIX Version 5.0 and higher is not compatible with the metadata format of LSM in Tru64 UNIX Version 4.0. If an older metadata format is detected during the upgrade procedure, LSM automatically upgrades the old format to the new format. If you do not want certain disk groups to be upgraded, deport them before you upgrade LSM.

You cannot deport the `rootdg` disk group; `rootdg` must be converted to the new format to allow use of the LSM configuration on the upgraded system. After `rootdg` is converted, it cannot be used again on a system running Version 4.0.

To deport a disk group:

```
# voldg deport disk_group
```

If you later import a deported disk group, LSM upgrades the metadata format.

7.1.4 Upgrading the LSM Software

LSM comprises three software subsets, which are located on the CD-ROM containing the base operating system software for the Tru64 UNIX product kit.

Depending on the operating system versions you are upgrading from and to, you might have to perform a full installation instead of an update installation or a succession of update installations. (For a description of the supported update paths, see the *Installation Guide*.)

- During an update installation, you do not need to specify the LSM subsets if they are already installed on the system you are upgrading. The upgrade installation process automatically upgrades all subsets installed on the system.

- If LSM is not already installed on the system you are upgrading from Tru64 UNIX Version 4.0x to Version 5.0x, you cannot install it during an update installation. To install LSM, you must do a full operating system installation.

During a full installation, you have the option to install the base system's root (/), /usr, and /var file systems and swap space to LSM volumes. If you will either use this system to create a cluster or add this system to a cluster, skip this option. The base system's root LSM volumes are not used in a cluster.

Caution

Be careful that you do not install the operating system on one of the disks currently part of your LSM configuration; the installation process completely overwrites anything on the disk and you will lose your configuration.

After the update or full installation, the `rootdg` disk group is converted and ready to use. Any disk groups that remained connected to the system (and that were not deported) are also converted and available.

7.1.5 Manually Converting Version 4.0 Disk Groups

If you deported disk groups before upgrading a system or cluster from Version 4.0 to Version 5.0 or higher, you can manually import and convert those disk groups.

Disk groups that are connected to a system before it restarts, or before cluster creation, are automatically imported. Their metadata format is updated, and the `vollogcnvt` utility converts BCLs to DRLs where possible. (For more information, see `vollogcnvt(8)`.)

The following procedure applies only to disk groups that you deported before upgrading the operating system and have decided to import and convert.

This procedure:

- Converts the disk group's internal metadata format to the format used in Version 5.0 and higher.
- Converts BCL logs to DRL logs for all volumes with a BCL of at least two blocks.
- Notifies you about volumes whose BCL logs could not be converted to DRLs.

The volumes are usable but are not logging. To enable logging, you must manually remove the BCL subdisk and add a new DRL log.

To manually import and convert disk groups:

1. Physically connect the storage to the system or cluster.
In a cluster, connect the storage so that it is accessible by all cluster members.
2. Run the `hwmgr` command to update the system or cluster with the new disk information. For more information, see `hwmgr(8)`.
3. Import and convert the disk group.

```
# voldg -o convert_old import disk_group
```

The disk group is imported and the following information is displayed for volumes that use BCLs:

```
lsm:voldg:WARNING:Logging disabled on volume. Need to convert to DRL.  
lsm:voldg:WARNING:Run the vollogcnvt command to automatically convert logging.
```

4. Convert any BCLs to DRLs for each disk group:
5. If a BCL cannot be converted to a DRL and you want to restore logging for the volume:

```
# vollogcnvt -g disk_group
```

- a. Identify the disabled BCL subdisk:

```
# volprint [-g disk_group] volume
```

- b. Remove the BCL subdisk:

```
# volsd [-g disk_group] -o rm dis subdisk
```

- c. Add a new log to the volume:

```
# volassist [-g disk_group] addlog volume
```

6. Start the volumes in each newly imported disk group:

```
# volrecover -g disk_group
```

7.1.6 Optimizing Restored LSM Configuration Databases (Optional)

If you restored an LSM configuration on a system that you upgraded from Tru64 UNIX Version 4.0 to Tru64 UNIX Version 5.0 or higher, you can modify the configuration databases to allow LSM to automatically manage their number and placement.

Note

This procedure is an optimization and is not required.

On systems running Tru64 UNIX Version 4.0 and using LSM, you had to explicitly configure between four and eight disks per disk group to have

enabled databases. In Version 5.0 and higher, by default all LSM disks are configured to contain copies of the database, and LSM automatically maintains the appropriate number of enabled copies. The distinction between an enabled and disabled copy is as follows:

- Disabled — The disk's private region is configured to contain a copy of the configuration database, but this copy might be dormant (inactive). LSM enables a copy as needed; for example, when a disk with an enabled copy is removed or fails.
- Enabled — The disk's private region is configured to contain a copy of the configuration database, and this copy is active. All LSM configuration changes are recorded in each enabled copy of the configuration database as they occur.

Configure the private regions on all your LSM disks to contain one copy of the configuration database, unless you have a specific reason for not doing so, such as:

- The disk is old or slow.
- The disk is on a bus that is heavily used.
- The private region is too small (less than 4096 blocks) to contain a copy of the configuration database (such as disks that have been migrated from earlier releases of LSM).
- There is some other significant reason why the disk should not contain a copy.

Enabling the configuration database does not use additional space on the disk; it merely sets the number of enabled copies in the private region to 1.

To set the number of configuration database copies to 1:

```
# voldisk modddb disk nconfig=1
```

For disk groups containing three or fewer disks, each disk should have two copies of the configuration database to provide sufficient redundancy. This is especially important for systems with a small `rootdg` disk group and one or more larger secondary disk groups.

For more information on modifying the LSM configuration databases, see Section 5.3.3.

7.2 Adding a System with LSM to a Cluster

You can add a standalone system with LSM volumes to an existing cluster and incorporate its LSM volumes in the cluster, whether or not the cluster is also using LSM. Alternatively, you can move only the disk groups from one system to another or from a system to a cluster.

Note

If the standalone system is not running at least Tru64 UNIX Version 5.0, see Section 7.1 to upgrade the system and its LSM configuration before adding the system to a cluster.

Before you begin, decide what you want to do with the standalone system's `rootdg` disk group. There can be only one `rootdg` disk group in an LSM configuration.

- **If LSM is running on the cluster** and you want to use the disks in the standalone system's `rootdg` disk group, you must either rename the former `rootdg` as you import it to the cluster or add the LSM disks to different disk groups.
- **If LSM is not running on the cluster** and you do not want to reuse the `rootdg` on the cluster, then you need to find one or more unused disks to create the `rootdg` for the cluster.

For information on identifying unused disks, see Section 2.3.

Review the following information and recommendations, and take any necessary actions:

- There is no relationship between the base system LSM volumes for the root (`/`), `/usr`, and `/var` file systems and the cluster's `cluster_root`, `cluster_usr`, and `cluster_var` domains.

Likewise, there is no relationship between the primary swap volume (`swapvol`) on a standalone system and the private swap space for each member. There is no primary swap space in a cluster; each member has its own swap space.

The base system volumes are not used, and the file systems using those volumes are not available until explicitly mounted. If you halt the member and boot the base operating system again, these file systems are available.

All other LSM volumes in imported disk groups are available to all cluster members, as long as the storage is accessible by all members. (They should be shared.)

- Before adding the system to the cluster you can optionally unmirror any mirrored system volumes to make that disk space available for other uses in the cluster.

You do not need to completely unencapsulate the root file system (remove the boot partitions from LSM volumes entirely); you can remove just the mirrors and return that disk space to LSM's free space pool, then create the cluster.

However, if the standalone system will not be used again as such, you can delete all the base system volumes and reinitialize the disks for LSM as sliced or simple disks, before adding the system to a cluster.

- If any of the standalone system's LSM disks are internal, they will be private to that member. If that member crashes, the cluster loses access to those volumes. If you want that data to be available to the rest of the cluster, move the data on those disks to storage that is shared across the cluster.

Likewise, consider making any external storage connected to the standalone system available to the whole cluster, or use those disks only for data that is member specific.

- For performance reasons, the LSM configuration on the standalone system might use nondefault values and attributes.

For example, if mirrored volumes on the standalone system have log subdisks that are less than 65 blocks per gigabyte of volume size, remove the old logs and add new ones, which will be sized appropriately by default, to support migration to a cluster environment. Otherwise, logging is disabled on these volumes, but the volumes themselves are usable.

To add a standalone system using LSM to a cluster not running LSM:

1. If applicable, reconfigure the log subdisks on all mirrored volumes to use the default DRL size.

- a. Identify mirrored volumes with nondefault-size log plexes:

```
# volprint -pht | grep -p LOGONLY
```

Information similar to the following is displayed. In this example, the log plex `vol11-03` is only 2 blocks long, but the log plex `vol12-03` is 65 blocks:

```
pl vol11-03    vol11    ENABLED  ACTIVE  LOGONLY  CONCAT  -      RW
sd dsk27a-01  vol11-03 dsk27a   0       2        LOG     dsk27a  ENA

pl vol12-03    vol12    ENABLED  ACTIVE  LOGONLY  CONCAT  -      RW
sd dsk27a-02  vol12-03 dsk27a   2       65       LOG     dsk27a  ENA
```

- b. Delete the nondefault-size DRL plex from its volume:

```
# volplex [-g disk_group] -o rm dis log_plex
```

For example:

```
# volplex -o rm dis vol11-03
```

- c. Add a new DRL plex to the volume, which will automatically be sized correctly:

```
# volassist addlog volume
```


For example:

```
# volassist addlog voll
```

2. Stop all volumes in each disk group:

```
# volume -g disk_group stopall
```

3. Deport each disk group except `rootdg`:

```
# voldg deport disk_group
```

4. Display the disk group ID for `rootdg`:

```
# voldg list rootdg | grep id
dgid:      1007697459.1026.hostname
```

5. Make a note of the disk group ID. You will need this information to import the `rootdg` disk group on the cluster.

6. Halt the system and add it to the cluster. Make sure all its storage is connected to the cluster (preferably as shared storage).

This step involves using the `clu_add_member` command and possibly other hardware-specific or cluster-specific operations that are not covered here.

7. Run the `hwmgr` command to update the cluster with the new disk information. For more information, see `hwmgr(8)`.

8. Initialize LSM using one of the following methods:

- To reuse the standalone system's `rootdg` as the `rootdg` for the cluster:

- a. Set up the LSM device special files:

```
# volinstall
```

- b. Start LSM in disabled mode:

```
# vold -m disable
```

- c. Initialize the LSM daemons with the source system's host name:

```
# voldctl init hostname
```

- d. Make sure the source system's `rootdg` storage is attached to the cluster, preferably as shared storage.

- e. Initialize the source system's `rootdg` disk group, making it the `rootdg` disk group for the cluster:

```
# voldg init rootdg
```

This allows the `rootdg` disk group to be autoimported, but it still has the source system's host name.

- f. Restart LSM in enabled mode:

```
# vold -k
```

This automatically imports all the other disk groups and, if necessary, converts their metadata to the latest format and sets them to be shared.

- g. Reset the host name in the `/etc/vol/volboot` file (and consequently on all disks in the new `rootdg` disk group) to the cluster alias name:

```
# voldctl hostid cluster_alias
```

- To create a new `rootdg` for the cluster:

- a. Specify at least two disks or disk partitions; for example:

```
# volsetup dsk19 dsk20
```

- b. Import and rename the old `rootdg` using its disk group ID:

```
# voldg -o shared -n newname import id=rootdg_dgid
```

- c. Import the other disk groups and set them to be shared:

```
# voldg -o shared import disk_group
```

9. Synchronize LSM throughout the cluster by entering the following command on all members except the member where you performed step 8:

```
# volsetup -s
```

7.3 Moving Disk Groups Between Systems

You can move an LSM disk group between standalone systems, between clusters, from a standalone system to a cluster, and vice versa and retain the LSM objects and data on those disks, as long as either of the following is true:

- The disk group was deported with the `voldg` command.
A disk group that requires recovery (for example, due to a system crash) cannot be moved between systems. The disk group must first be recovered in the environment where it was last used.
- The system originally using the disk group was cleanly shut down. (For `rootdg`, this must be true because you cannot deport `rootdg`.)

Moving a disk group between systems causes the new host system to assign new disk access names to the disks. For LSM `nopriv` disks (created when you encapsulate disks or partitions), the association between the original disk access name and its disk media name might be lost or might be reassociated incorrectly. To prevent this, you must manually reassociate the disk media names with the new disk access names. For LSM `sliced` and `simple` disks, LSM manages this reassociation.

If possible, before moving the disk group, migrate the data from `nopriv` disks to `sliced` or `simple` disks, which have a private region and will be reassociated automatically. For more information on moving data to a different disk, see Section 5.1.5.

If you cannot move the data to `sliced` or `simple` disks, see Section 7.3.3.

You can change the disk group's name or host ID when you move it to the new host; for example, to reduce the chance for confusion if the new host has a disk group with a similar name. You must change the disk group's name if the new host has a disk group with the same name.

You can change the disk group's host ID to that of the receiving system as you deport it from the original system. This allows the system receiving the disk group to import it automatically when it starts. If the new host is already running, the disk group's host ID is changed when you import the disk group on the new host.

7.3.1 Moving the `rootdg` Disk Group to Another System

You can move the `rootdg` disk group from one standalone system to another with the following restrictions:

1. If the system's root disk and swap space are encapsulated to LSM volumes, remove them from LSM control. The root file system cannot be reused on another system.

For information on removing the system volumes, see Section 7.4.

2. If other system-specific file systems use LSM volumes, also remove them from LSM control.

There can be no duplication of file systems on one system. Only file systems or applications that are not critical to the system's operation, or that do not exist on the target system, can be moved between systems.

For more information on unencapsulating AdvFS domains or UFS file systems, see Section 5.4.6.1.

3. If `rootdg` contains any internal system disks, remove those disks from `rootdg` (and, if necessary, from LSM control).
4. You cannot deport `rootdg`; to move it, either shut down the system, or stop running LSM on the system. This involves stopping all volumes (and stopping access from any file systems and applications that use those volumes) and stopping the LSM daemons. After the `rootdg` disk group is removed from a system, the system can no longer run LSM (unless you create a new `rootdg`).
5. If you do not plan to run LSM again on the system, edit the `/etc/inittab` file to remove the LSM startup routines:

```

lsmr:s:sysinit:/sbin/lsmbootstrap -b </dev/console >/dev/console 2>&1 ##LSM
lsm:23:wait:/sbin/lsmbootstrap </dev/console >/dev/console 2>&1 ##LSM
vol:23:wait:/sbin/vol-reconfig -n </dev/console >/dev/console 2>&1 ##LSM

```

6. Optionally, recursively delete the `/dev/vol/` and `/etc/vol/` directories.

7.3.2 Moving Disk Groups to Another System

To move a disk group other than `rootdg` to another system:

1. Stop all activity on the volumes in the disk group and unmount any file systems.
2. Deport the disk group from the originating system:
 - To deport the disk group with no changes:


```
# voldg deport disk_group
```
 - To deport the disk group and assign it a new host ID or a new name:


```
# voldg [-n new_name] [-h new_hostID] deport disk_group
```
3. Physically move the disks to the new host system.
4. Enter the following command on the new host system to scan for the disks:

```
# hwmgr scan scsi
```

The `hwmgr` command returns the prompt before it completes the scan. You need to confirm that the system has discovered the new disk before continuing, such as by entering the `hwmgr show scsi` command until you see the new device.

5. Make LSM aware of the newly added disks:


```
# voldctl enable
```
6. Import the disk group to the new host:
 - If the disk group contains `nopriv` disks whose disk media names no longer correspond to their original disk access names, you might need to use the `force (-f)` option.
 - If the disk group is moving from a standalone system to a cluster, use the `-o shared` option.
 - If the disk group is moving from a cluster to a standalone system, use the `-o private` option.
 - If the disk group has the same name as another disk group on the system, rename it as you import it.

```
# voldg [-f] [-o shared|private] [-n new_name] import \  
disk_group
```

7. If applicable, associate the disk media names for the `nopriv` disks to their new disk access names:


```
# voldg -g disk_group -k adddisk \  
disk_media_name=disk_access_name...
```
8. Recover and start all startable volumes in the imported disk group. The following command performs any necessary recovery operations as a background task after starting the volumes:


```
# volrecover -g disk_group -sb
```
9. Optionally, identify any detached plexes.


```
# volinfo -p
```

If the output lists any volumes as Unstartable, see Section 6.5.2.2 for information on how to proceed.
10. If necessary, start the remaining Startable volumes:


```
# volume -g disk_group start volume1 volume2...
```

7.3.3 Moving Disk Groups with `nopriv` Disks to Another System

When LSM disks are moved to a different system or added to a cluster, the operating system assigns them new device names (LUNs) that are not likely to be the same as their previous device names. LSM bases the disk access name on the device name and maintains an association between the disk access name and the disk media name, which can be anything you assign, such as `big_disk`. If you move a disk group to another system, LSM uses this association (stored in in the configuration database) to remap the disk media names to the new device names (disk access names) for `sliced` and `simple` disks, but not for `nopriv` disks.

Moving a disk group with multiple `nopriv` disks involves a lengthy, careful process that includes identifying the disks while they are connected to the original system or cluster and making a detailed record of the `nopriv` disks with enough information to help you properly identify them in the new environment.

You might create a list of the disk access name, disk media name, and disk group name for each `nopriv` disk and physically label each disk (with a sticker or adhesive tape, for example) with this same information. Then you can move the disks to the new environment and use commands such as `hwmgr flash light` to physically locate the disks. When you determine their new disk access names, you can import the disk group and then associate the old disk media names to the new disk access names for the `nopriv` disks.

Because `nopriv` disks require additional effort to manage, we strongly advise that you use them only to place data under LSM control (through

encapsulation) and then immediately move those volumes to sliced or simple disks.

Before you begin, you might want to review the syntax for `hwmgr flash` by displaying its online help:

```
# hwmgr -h flash light
```

```
Usage: hwmgr flash light
      [ -dsf <device-special-filename> ]
      [ -bus <scsi-bus> -target <scsi-target> -lun <scsi-lun> ]
      [ -seconds <number-of-seconds> ] (default is 30 seconds)
      [ -nopause ] (do not pause between flashes)
The "flash light" operation works only on SCSI disks.
```

You can use the `-seconds` option with the `-nopause` option to cause the disk's light to remain on constantly for the length of time you specify. Without the `-nopause` option, the light flashes on and off for the specified duration. In a busy environment, you might not be able to tell whether a light is flashing because of your command or because of I/O.

If there is only one `nopriv` disk in the disk group, there is only one device to reassociate. As long as you are not connecting other devices to the new host at the same time, you might not need this information. For two or more `nopriv` disks, having precise identification beforehand is crucial.

To move a disk group with multiple `nopriv` disks to a different system or cluster:

1. On the original host, use the following commands to identify all the `nopriv` disks in the disk group by their current disk access name and disk media name and a unique identifier (such as the disk's SCSI world-wide identifier) that will not change or can be tracked when the disk is connected to the new system. Create a list (or a printable file) containing this information.

- a. List the disks in the disk group:

```
# voldisk -g disk_group list
```

DEVICE	TYPE	DISK	GROUP	STATUS
dsk21	sliced	dsk21	datadg	online
dsk22	sliced	dsk22	datadg	online
dsk23c	nopriv	dsk23c-4.2BSD	datadg	online
dsk24c	nopriv	dsk24c-database	datadg	online
dsk26c	nopriv	dsk26c-AdvFS	datadg	online
dsk27g	nopriv	dsk27g-raw	datadg	online

- b. Find the hardware IDs (HWIDs) of the disks:

```
# hwmgr show scsi
```

HWID:	SCSI DEVICEID	HOSTNAME	DEVICE TYPE	DEVICE SUBTYPE	DRIVER OWNER	NUM PATH	DEVICE FILE	FIRST VALID	PATH
-------	------------------	----------	----------------	-------------------	-----------------	-------------	----------------	----------------	------

```

-----
:
88: 22      lsmtmp  disk    none    2       1      dsk21  [5/3/0]
89: 23      lsmtmp  disk    none    2       1      dsk22  [5/4/0]
90: 24      lsmtmp  disk    none    2       1      dsk23  [5/5/0]
91: 25      lsmtmp  disk    none    2       1      dsk24  [5/6/0]
92: 26      lsmtmp  disk    none    2       1      dsk25  [6/1/0]
93: 27      lsmtmp  disk    none    2       1      dsk26  [6/3/0]
94: 28      lsmtmp  disk    none    2       1      dsk27  [6/5/0]

```

- c. Use the HWID value for each `nopriv` disk to find its world-wide ID (WWID):

```
# hwmgr show scsi -full -id HWID
```

For example:

```
# hwmgr show scsi -full -id 90
```

```

          SCSI
HWID:  DEVICEID  HOSTNAME  DEVICE  DEVICE  DRIVER  NUM  DEVICE  FIRST
-----
          90:  24      lsmtmp   disk    none    2       1      dsk23  [5/5/0]

          WWID:04100024:"DEC      RZ1CF-CF (C) DEC      50060037"

          BUS  TARGET  LUN  PATH STATE
          ----
          5    5      0   valid

```

2. Physically label each `nopriv` disk with its disk access name, disk media name, and WWID.
3. Deport the disk group on the original host:

```
# voldg deport disk_group
```

4. Physically connect the disk group to the new environment.

Keep track of the before-and-after bus locations of each `nopriv` disk as you move it between systems. Then when you scan for the disks on the new host, you will know which new disk access name associated with the new bus location belongs to which disk media name. You can move each disk individually and use the `hwmgr` command to scan for it each time to be sure.

5. On the new system or cluster, enter the following command to discover and assign device names to the newly attached storage:

```
# hwmgr scan scsi
```

6. Import the disk group using the force (`-f`) option, which forces LSM to import the disk group despite not being able to import the `nopriv` disks.

```
# voldg -f [-o shared|private] import disk_group
```

- If the disk group is moving from a standalone system to a cluster, mark it for use in a cluster with the `-o shared` option.
 - If the disk group is moving from a cluster to a standalone system, mark it as private with the `-o private` option.
7. Make a note of the disks that LSM reports were not found.
 8. Display the disks in the imported disk group:


```
# voldisk -g disk_group list
```

The output shows only `sliced` and `simple` disks. The `nopriv` disks are still not imported.
 9. Compare the disk access names with the output of the following command:


```
# hwmgr show scsi
```

The new device names that appear in the output from step 9 but not in the output from step 8 are probably the `nopriv` disks.

The device special file name for each device name appears in the `DEVICE FILE` column; use that identifier in step 10.
 10. For each suspect device name, run the following command:


```
# hwmgr flash light -dsf device_special_filename \  
-seconds duration -nopause
```
 11. Find the disk with the constantly-on light.

If the disk is one of the labeled `nopriv` disks that came from the other system, write down the disk media name and correlate it to the new device name. For example, write the new device name next to the old disk media name on your list from step 1.
 12. Add each `nopriv` disk to the disk group, associating its disk media name with its new device (disk access) name:


```
# voldg -g disk_group -k adddisk media_name=device_name
```
 13. Start, and if necessary, recover the volumes on the `nopriv` disks:


```
# voldg -g disk_group startall  
# volrecover -g disk_group -sb
```

7.4 Unencapsulating the Boot Disk (Standalone System)

If you encapsulated the root file systems (`/`, `/usr`, and `/var`) and the primary swap partition on a standalone system (Section 3.4.1) and later decide you want to stop using LSM volumes and return to using physical disk partitions, you can do so by unencapsulating the boot disk and primary swap space. This process involves restarting the system.

Note

To stop using LSM volumes for the clusterwide root, /usr, and /var file system domains, use the `volunmigrate` command. For more information, see Section 7.5 and `volunmigrate(8)`.

The unencapsulation process changes the following files:

- If the root file system is AdvFS, the links in the `/etc/fdmns/*` directory for domains associated with the boot disk are changed to point to disk partitions instead of LSM volumes.
- If the root file system is UFS, the `/etc/fstab` file is changed to use disk partitions instead of LSM volumes.
- In the `/etc/sysconfigtab` file, the `swapdevice` entry is changed to use the original swap partition instead of the LSM `swapvol` volume and the `lsm_rootdev_is_volume` entry is set to 0.

To unencapsulate the system partitions:

1. If the system volumes (root, swap, /usr, and /var) are mirrored, do the following. If not, go to step 2.
 - a. Display detailed volume information for the boot disk volumes:

volprint -g rootdg -vht

V NAME	USETYPE	KSTATE	STATE	LENGTH	READPOL	PREFPLEX		
PL NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WID	MODE	
SD NAME	PLEX	DISK	DISKOFFS	LENGTH	{COL}/OFF	DEVICE	MODE	
v rootvol	root	ENABLED	ACTIVE	524288	ROUND	-		
pl rootvol-02	rootvol	ENABLED	ACTIVE	524288	CONCAT	-	RW	
sd root02-02p	rootvol-02	root02	0	16	0	dsk16a	ENA	
sd root02-02	rootvol-02	root02	16	524272	16	dsk16a	ENA	
pl rootvol-01	rootvol	ENABLED	ACTIVE	524288	CONCAT	-	RW	
sd root01-01p	rootvol-01	root01	0	16	0	dsk14a	ENA	
sd root01-01	rootvol-01	root01	16	524272	16	dsk14a	ENA	
v swapvol	swap	ENABLED	ACTIVE	520192	ROUND	-		
pl swapvol-02	swapvol	ENABLED	ACTIVE	520192	CONCAT	-	RW	
sd swap02-02	swapvol-02	swap02	0	520192	0	dsk16b	ENA	
pl swapvol-01	swapvol	ENABLED	ACTIVE	520192	CONCAT	-	RW	
sd swap01-01	swapvol-01	swap01	0	520192	0	dsk14b	ENA	
v vol-dsk14g	fsgen	ENABLED	ACTIVE	2296428	SELECT	-		
pl vol-dsk14g-02	vol-dsk14g	ENABLED	ACTIVE	2296428	CONCAT	-	RW	
sd dsk16g-01	vol-dsk14g-02	dsk16g-AdvFS	0	2296428	0	dsk16g	ENA	
pl vol-dsk14g-01	vol-dsk14g	ENABLED	ACTIVE	2296428	CONCAT	-	RW	
sd dsk14g-01	vol-dsk14g-01	dsk14g-AdvFS	0	2296428	0	dsk14g	ENA	
v vol-dsk14h	fsgen	ENABLED	ACTIVE	765476	SELECT	-		
pl vol-dsk14h-02	vol-dsk14h	ENABLED	ACTIVE	765476	CONCAT	-	RW	
sd dsk16h-01	vol-dsk14h-02	dsk16h-AdvFS	0	765476	0	dsk16h	ENA	
pl vol-dsk14h-01	vol-dsk14h	ENABLED	ACTIVE	765476	CONCAT	-	RW	
sd dsk14h-01	vol-dsk14h-01	dsk14h-AdvFS	0	765476	0	dsk14h	ENA	

Examine the output and decide which plexes you want to remove based on which disk each plex uses. Typically, the plexes with the -01 suffix are those using the original disk or disk partition and therefore are the ones you want to unencapsulate.

Note

In the previous example, the `rootvol` volume contains subdisks labeled `root01-01p` and `root02-02p`. These are phantom subdisks, and each is 16 sectors long. They provide write-protection for block 0, which prevents accidental destruction of the boot block and disk label. These subdisks are removed in the course of this procedure.

If the root file system and the primary swap space originally used different disks, the plexes you want to unencapsulate can be on different disks; for example, the `rootvol-01` plex can be on `dsk14` but the `swapvol-01` plex can be on `dsk16`.

- b. Remove all plexes except the one using the disk you want to unencapsulate. The remaining plex should be on the disk that you want to unencapsulate. This is the disk the system partitions will use after the unencapsulation completes.

```
# volplex -o rm dis plex-nn
```

For example, to remove secondary plexes for the volumes `rootvol`, `swapvol`, and `vol-dsk0g`:

```
# volplex -o rm dis rootvol-02
# volplex -o rm dis swapvol-02
# volplex -o rm dis vol-dsk14g-02
# volplex -o rm dis vol-dsk14h-02
```

2. Change the boot disk environment variable to point to the physical boot disk; in this case, the disk for plex `rootvol-01`:

```
# consvar -s bootdef_dev boot_disk
```

For example:

```
# consvar -s bootdef_dev dsk14
set bootdef_dev = dsk14
```

3. Unencapsulate the boot disk and primary swap disk (if different).

```
# volunroot -a -A
```

This command also removes the LSM private region from the system disks and prompts you to restart the system.

Information similar to the following is displayed. Enter **now** at the prompt.

```
This operation will convert the following file systems on the
system/swap disk dsk14 from LSM volumes to regular disk partitions:
```

```
Replace volume rootvol with dsk14a.
Replace volume swapvol with dsk14b.
Replace volume vol-dsk14g with dsk14g.
Replace volume vol-dsk14h with dsk14h.
Remove configuration database on dsk14f.
```

```
This operation will require a system reboot. If you choose to
continue with this operation, your system files will be updated
to discontinue the use of the above listed LSM volumes.
/sbin/volreconfig should be present in /etc/inittab to remove
the named volumes during system reboot.
```

```
Would you like to either quit and defer volunroot until later
or commence system shutdown now? Enter either 'quit' or time to be
used with the shutdown(8) command (e.g., quit, now, 1, 5): [quit] now
```

When the system restarts, the root file system and primary swap space use the original, unencapsulated disks or disk partitions.

If the system volumes were mirrored, the LSM disks that the mirror plexes used remain under LSM control as members of the `rootdg` disk group.

To reuse these LSM disks within LSM or for other purposes:

1. Display the LSM disks in the `rootdg` disk group:

```
# voldisk -g rootdg list

DEVICE      TYPE      DISK      GROUP      STATUS
:
dsk16a      nopriv   root02    rootdg     online
dsk16b      nopriv   swap02    rootdg     online
dsk16f      simple   dsk16f    rootdg     online
dsk16g      nopriv   dsk16g-AdvFS rootdg     online
dsk16h      nopriv   dsk16h-AdvFS rootdg     online
:
```

In this case, the LSM disks for the system volume mirror plexes have the disk media names `root02`, `swap02`, `dsk16g-AdvFS`, and `dsk16h-AdvFS`. All these LSM disks are on the same physical disk `dsk16`. The private region for `dsk16` has the disk media name `dsk16f`.

2. Remove these LSM disks from the `rootdg` disk group using their disk media names; for example:

```
# voldg rmdisk root02 swap02 dsk16g-AdvFS dsk16h-AdvFS dsk16f
```

3. Remove the disks from LSM control using their disk access names (in the `DEVICE` column); for example:

```
# voldisk rm dsk16a dsk16b dsk16f dsk16g dsk16h
```

The physical disk (in this case, `dsk16`) is no longer under LSM control and its disk label shows all partitions marked unused.

7.5 Migrating AdvFS Domains from LSM Volumes to Physical Storage

You can stop using LSM volumes for AdvFS domains and return to using physical disks or disk partitions with the `volunmigrate` command. This command works on both standalone systems and clusters. The domains remain mounted and in use during this process; no reboot is required.

You must specify one or more disk partitions that are not under LSM control, ideally on a shared bus, for the domain to use after the migration. These partitions must be large enough to accommodate the domain plus at least 10 percent additional space for file system overhead. The `volunmigrate` command examines the partitions that you specify to ensure they meet both qualifications and returns an error if either or both is not met. For more information, see `volunmigrate(8)`.

To migrate an AdvFS domain from an LSM volume to physical storage:

1. Display the size of the domain volume:

```
# volprint -vt domain_vol
```

2. Find one or more disk partitions on a shared bus that are not under LSM control and are large enough to accommodate the domain plus file system overhead of at least 10 percent:

```
# hwmgr view devices -cluster
```

3. Migrate the domain, specifying the target disk partitions:

```
# volunmigrate domain_name dsknp [dsknp...]
```

After migration, the domain uses the specified disks; the LSM volume no longer exists.

7.6 Unencapsulating a Cluster Member's Swap Devices

You can remove a cluster member's swap devices from LSM volumes and resume using physical disk partitions. This process is called unencapsulation and requires that you reboot the member.

When you originally encapsulated the swap device, LSM created two separate LSM disks: a `nopriv` disk for the swap partition itself, and a `simple` disk for LSM private data on another partition of the disk. The unencapsulation process removes only the `nopriv` disk.

To unencapsulate a member's swap devices:

1. Display the names of LSM volumes in the `rootdg` disk group. (All swap volumes must belong to `rootdg`.)

```
# volprint -g rootdg -vht
TY NAME          ASSOC          KSTATE  LENGTH  ...
v hughie-swap01  swap          ENABLED  16777216 ...
pl hughie-swap01-01 hughie-swap01 ENABLED  16777216 ...
sd dsk4b-01      hughie-swap01-01 ENABLED  16777216 ...
```

In the output (edited for brevity), look for the following:

- The name of the member's swap volume in the form `nodename-swapnn`; for example, `hughie-swap01`.
 - The disk partition (subdisk) used by the swap volume in the form `dsknp`; for example, `dsk4b`.
2. Edit the `/cluster/members/member{n}/boot_partition/etc/sysconfigtab` file for the member to remove the `/dev/vol/rootdg/nodename-swapnn` entry from the `swapdevice=` line.
 3. Reboot the member:

```
# shutdown -r now
```

When the member starts again, it no longer uses the LSM swap volume.

4. Log back in to the same member.
5. Remove the swap volume:

```
# voleit -rf rm nodename-swapnn
```

6. Find the LSM simple disk associated with the encapsulated swap device; for example:

```
# voldisk -g rootdg list | grep dsk4
dsk4b      nopriv    dsk4b      rootdg      online
dsk4f      simple   dsk4f      rootdg      online
```

7. Remove the LSM simple disk and the `nopriv` disk from the `rootdg` disk group and from LSM control; for example:

```
# voldg -g rootdg rmdisk dsk4b dsk4f
# voldisk rm dsk4b dsk4f
```

8. Set the cluster member to swap on the original disk partition (the former `nopriv` disk); for example:

```
# swapon /dev/disk/dsk4b
```

9. Edit the `/etc/sysconfigtab` file as follows:

- Add the `/dev/disk/dsknp` entry to the line `swapdevice=` so that the line reads:

```
swapdevice=/dev/disk/dsknp
```

For example:

```
swapdevice=/dev/disk/dsk4b
```

- If you removed the last LSM swap device for this member, set the value for `lsm_root_dev_is_volume=` to 0.

The cluster member uses the specified disk partition for its swap device and the LSM swap volume no longer exists.

7.7 Uninstalling the LSM Software

This section describes how to completely remove the LSM software from a standalone system or a cluster. This process involves:

- Backing up user data
- Unencapsulating disks or data
- Removing LSM objects and the software subsets
- Reconfiguring the kernel and restarting the system or cluster member

Caution

Uninstalling LSM causes any current data in LSM volumes to be lost. Before proceeding, back up any needed data.

To uninstall the LSM software:

1. Reconfigure any system-specific file systems and swap space, so they no longer use an LSM volume.
 - On a standalone system, unencapsulate the root file systems and primary swap partition (Section 7.4).
If additional (secondary) swap space uses LSM volumes, remove those volumes (Section 5.4.6).
 - In a cluster, migrate all AdvFS domains that use LSM volumes, including `cluster_root`, `cluster_usr`, and `cluster_var` from the LSM volumes to disk partitions (Section 7.5).
 - Unencapsulate all cluster members' swap devices (Section 7.6).
2. Unmount any other file systems that are using LSM volumes, so all LSM volumes can be closed.
 - a. Update the `/etc/fstab` file if necessary, so that it no longer mounts any file systems on an LSM volume.

- b. Stop applications that are using raw LSM volumes and reconfigure them, so that they no longer use LSM volumes.
3. Identify the disks that are currently configured under LSM:


```
# voldisk list
```
4. Restart LSM in disabled mode (in a cluster, on only one member):


```
# vold -k -r reset -d
```

This command fails if any volumes are open.
5. Stop all LSM volume and I/O daemons (in a cluster, on every member):


```
# voliod -f set 0
# voldctl stop
```
6. Update the disk labels for the disks under LSM control (in the output from step 3).
 - For each LSM `sliced` disk, apply a default disk label to the entire disk:


```
# disklabel -rw dskn
```
 - For each LSM `simple` disk, change the partition's `fstype` field to `unused`:


```
# disklabel -s dsknP unused
```
 - For each LSM `nopriv` disk, change the partition's `fstype` field to either `unused` or the appropriate value, depending on whether the partition still contains valid data.

For example:

 - To change the `fstype` field for partition `dsk2h`, which contains no valid data:


```
# disklabel -s dsk2h unused
```
 - To change the `fstype` field for partition `dsk2g`, which contains a valid UFS file system:


```
# disklabel -s dsk2g 4.2BSD
```
7. Remove the LSM directories:


```
# rm -r /etc/vol /dev/vol /dev/rvol /etc/vol/volboot
```
8. Delete the following LSM entries in the `/etc/inittab` file (in a cluster, for every member):


```
lsmr:s:sysinit:/sbin/lsmbootstrap -b </dev/console >/dev/console 2>&1 ##LSM
lsm:23:wait:/sbin/lsmbootstrap </dev/console >/dev/console 2>&1 ##LSM
vol:23:wait:/sbin/vol-reconfig -n </dev/console >/dev/console 2>&1 ##LSM
```
9. Display the installed LSM subsets:

```
# setld -i | grep LSM
```

10. Delete the installed LSM subsets:

```
# setld -d OSFLSMBASEnnn OSFLSMBINnnn OSFLSMCLSMTOOLSnnn
```

11. In the `/sys/conf/hostname` file (in a cluster, for every member), change the value of the pseudo-device `lsm` entry from 1 to 0.

In a cluster, the *hostname* is the member name, not the cluster alias.

You can make this change either before or while running the `doconfig` command; for example:

```
# doconfig -c hostname
```

12. Copy the new kernel to the root (`/`) directory (in a cluster, on every member):

```
# cp /sys/hostname/vmunix /
```

13. Restart the system or cluster member.

For information the appropriate way to restart each member, see the *Cluster Administration* manual.

When the system restarts, or after every cluster member restarts, LSM will no longer be installed.

A

The Storage Administrator GUI (lsmsa)

This chapter describes how to manage AdvFS domains, filesets, and volumes and LSM objects, including disks, disk groups, volumes, plexes, and subdisks using the Storage Administrator GUI, also called `lsmsa`.

For more information on how to track Storage Administrator activities and how to customize the Storage Administrator GUI, see Appendix B.

A.1 Overview of the Storage Administrator

The Storage Administrator is a Java-based graphical user interface (GUI) for LSM. The Storage Administrator displays a hierarchical view of LSM objects, AdvFS domains, and their relationships. You use the Storage Administrator to view and manage LSM objects and AdvFS domains on a local or remote (client) system.

The Storage Administrator consists of a server (daemon) and a client. The Storage Administrator server runs on a system that you want to administer. If LSM is not initiated, then the interface starts up in AdvFS-only mode. The Storage Administrator client runs on any machine that supports the Java run-time environment.

The Storage Administrator provides dialog boxes in which you enter information to create or manage LSM objects or AdvFS domains. Completing a dialog box might be the equivalent of entering several commands.

Note the following considerations when using the Storage Administrator:

- If you are working in a TruCluster environment, some restrictions apply. Dialog box options for invalid tasks are grayed out.
- Mirrors are allowed in clusters except for the individual member boot partitions and the quorum disk.
- Software-based RAID 5 technology is not supported in clusters. If you want RAID 5 functionality, use hardware RAID devices.
- AdvFS file systems are supported in all modes.
- UFS file systems are supported in read-only mode.

A.1.1 Installing and Starting the Storage Administrator GUI

To install the Storage Administrator, choose the LSM GUI option during the LSM installation. Install the Storage Administrator on all systems on which you want to use the Storage Administrator to remotely manage the LSM software. For information on initializing the LSM software, see Chapter 3.

To use the Storage Administrator, you must log in as `root` unless your user name is in the `/etc/group` file for the system, in the group defined for Storage Administrator administration (`lsmsa_admin` by default).

To start the Storage Administrator:

```
# /usr/bin/lsmsa
```

The Session Initiation dialog box is displayed. Enter the following information, then click on Ok:

- The name of the system to be administered in the Host Server field
- `root` or your user name
- The password associated with the account

When the Storage Administrator is started, the system attempts to connect it with the server process on the indicated host.

- If the Storage Administrator cannot connect to the server process, the system then attempts to connect it to the indicated host at the `initlsmsad` port, defined in the `/etc/services` file, and `vrts.remote.server.initLsmsadPort` port defined in the `/usr/lib/java/applications/lsmsa/properties` file. The port number defined in these two files must be the same.

When the Storage Administrator connects to the `initlsmsad` port, the `inetd` server executes the `/usr/lib/java/applications/lsmsa/initlsmsad` program, which creates a subprocess where the `/usr/sbin/lsmsad` script will run. The `lsmsad` script starts the LSMSA server processes `VMServerImpl`, `VRTSRegistry` and `cmdserver`. After the LSMSA server processes are started, the Storage Administrator then connects and operates normally.

- If the Storage Administrator cannot connect to a port, the error message “Cannot connect to the server” is displayed. When the Storage Administrator exits and disconnects from the server, the server continues to exist in an idle state until another GUI connects or an LSM configuration event occurs (such as creation, deletion, or modification of an LSM object). If the server receives notification of an LSM configuration event and if no GUIs are connected, the server exits. When the server exits, all the Storage Administrator server processes exit as well.

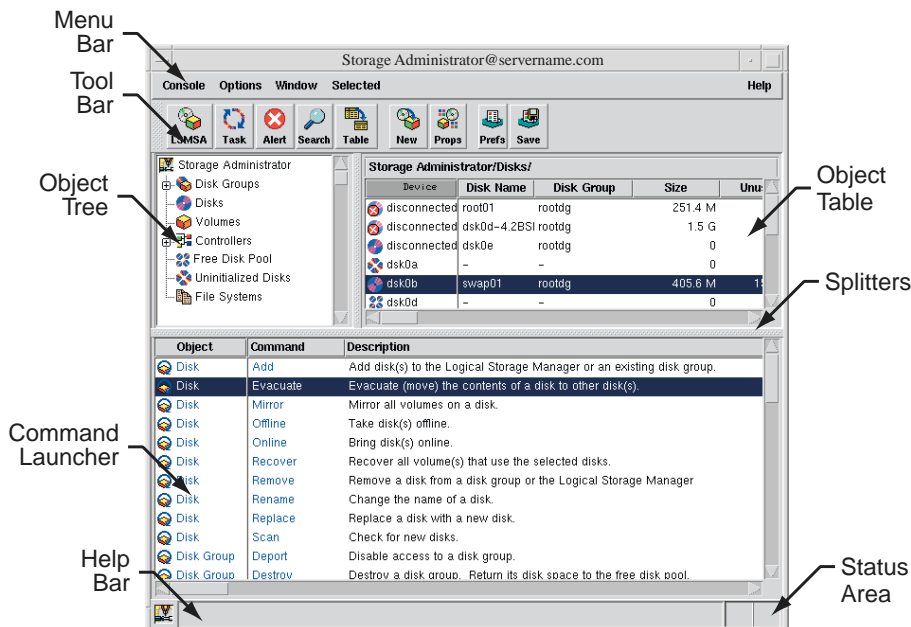
If the Storage Administrator cannot connect to the server, try the following:

- Examine the `/var/lmsmsa/logs/server.log` file for startup and error messages.
- Run the `/usr/lib/java/applications/lmsmsa/initlmsmsad` program to view error messages. You must be `root` user.
- On a very slow network, you might need to adjust the value assigned to the `CONNECTION_TIMEOUT` variable in the `/usr/sbin/lmsmsad` script. This value is the amount of time after startup that the `lmsmsa` server will wait for a connection from the client. The default value is 30 seconds. If the server process receives an LSM configuration event and if no GUIs have been or are currently connected, and if the `CONNECTION_TIMEOUT` seconds have elapsed, the server exits.

A.1.2 Storage Administrator Main Window Components

The Storage Administrator main window consists of two panes. The left pane displays a hierarchical tree of objects. The right pane displays an object table that lists the properties of the object you select on the left. The window also has a Menu Bar and a Command Launcher that you can hide or display to initiate LSM options. Figure A–1 shows the components of the main window.

Figure A–1: Storage Administrator Main Window



The Object Tree displays the hierarchical relationship between LSM objects and between objects in AdvFS domains. Each object represents a group of components of the same type. When you select the icon to the left of an object, components of that type appear in the right pane. You can expand objects (by clicking on the plus sign) to display their hierarchy.

The Object Table displays component objects that belong to the currently selected object in the Object Tree. The Object Table is dynamic and constantly updates its contents to reflect changes to the system.

The Command Launcher displays a list of tasks that you can perform on objects. When you click on a task in the Command Launcher list, the task starts and the dialog box for the task appears.

The Menu Bar contains the following menus:

- Console menu — Provides access to the New menu, which creates volumes, disk groups, or file systems. It also closes the Main window, provides access to an object properties dialog box, or exits the Storage Administrator.
- Options menu — Provides access to the Preferences dialog box and saves or loads user preferences for Storage Administrator components. The Options menu also removes alert icons from the Status Area.
- Window menu — Opens additional Storage Administrator Main windows, the Task Request Monitor, the Alert Monitor window, the Search window, a copy of the Object Table, or the Command Launcher.
- Selected menu — By default, the Selected menu is grayed out and changes its options based on the type of object that you select. Until you have selected an object, you see the grayed Selected entry on the Menu Bar. When you select an object, for example Volumes, the Selected entry changes to Volumes.
- Help menu — Provides access to online Storage Administrator help.

The toolbar consists of the following buttons that provide access to the following windows:

- LSMSA button — Launches an additional Storage Administrator Main window.
- Task button — Launches the Task Request Monitor window.
- Alert button — Launches the Alert Monitor window.
- Search button — Launches the Object Search window.
- Table button — Launches a window that contains a copy of the main Object Table.

- New button — Launches the New Volume dialog box that is used to create a volume.
- Props button — Launches the Object Properties dialog box for a selected object.
- Prefs button — Launches the Preferences dialog box.
- Save button — Saves the current preference settings for use in future Storage Administrator sessions.

A.2 Performing Tasks with the Storage Administrator

You perform most tasks by selecting objects or tasks, then providing information in resulting dialog boxes. You perform tasks using:

- The Menu Bar
- A pop-up menu
- The Command Launcher

A.2.1 Selecting Objects

To select a single object, click on it. To deselect the object, click on it again.

To select or deselect multiple objects, hold down the Control key while selecting the objects. The objects that you select do not have to be adjacent.

To select a range of adjacent objects, select the first object and then hold down the Shift key while selecting the last object in the range. You can also select multiple adjacent objects by dragging the mouse over the desired objects while pressing the Shift key.

A.2.2 Using the Console Menu and Selected Menu

You can launch tasks from the Console and Selected menus in the Menu Bar.

Choose New from the Console menu to create:

- LSM volumes and disk groups
- File systems
- AdvFS domains, volumes, and filesets

The context-sensitive Selected menu changes to reflect which object you selected in the Object Table.

For example, to change a volume name, select Volumes in the Object Tree and the volume you want to rename in the Object Table. From the Volumes menu choose Rename, enter information in the Rename Volume dialog box, and click on OK.

A.2.3 Using the Right Mouse Button to Display a Pop-Up Menu

Click on the right mouse button to access a context-sensitive pop-up menu to display common task information that you can apply to the selected object. Additional tasks are available through the Menu Bar or the Command Launcher.

For example, to create a new volume in a disk group, select Disk Groups in the Object Tree, right-click on the disk group in the Object Table, choose New Volume from the pop-up menu, enter information in the New Volume dialog box, and click on OK.

A.2.4 Using the Command Launcher Window

The Command Launcher window contains a list of objects and associated tasks. To display the Command Launcher window, choose Command Launcher from the Window menu. To hide the Command Launcher window, choose Command Launcher again from the Window menu.

To perform a task on a specific type of object, select the appropriate object-command combination from the Command Launcher list. For example, to create a volume, choose Volume–New from the Command Launcher, enter information in the New Volume dialog box, and click on OK.

Caution

The Command Launcher does not restrict context. If you choose inappropriate commands and ignore the warning dialogs, you can perform operations that might result in permanent loss of data.

A.2.5 Exiting the Storage Administrator

To exit the Storage Administrator, choose Exit from the Console menu.

A.3 Managing LSM Disks

The following sections describe LSM disk management tasks that you can complete using the Storage Administrator.

A.3.1 Adding an LSM Disk

When you add a disk for use with the LSM software, the disk is either initialized or encapsulated. If the disk is not set up, initialize it. If you want to use a disk with partitions that are in use, encapsulate it. Encapsulation preserves any existing data on the disk in the form of volumes. Initialization

destroys any existing data on the disk. Initialized disks are placed in the free disk pool and are available to add to a disk group.

To add a disk for use with the LSM software:

1. In the Object Tree, select Disks and in the Object Table, select a disk to place under LSM control.

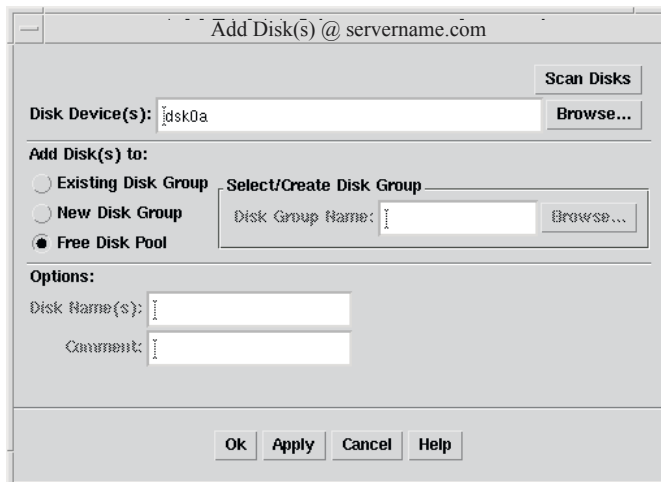
If the disk is new and does not show up in the Object Table when you highlight Disks, click on Storage Administrator (at the top of the Object Tree) and choose Scan Disks from the System menu.

In the Scan Disks dialog box, click on Ok to begin the search. From the Object Tree displayed, select a disk to add.

2. From the Disks menu, choose Add.

The Add Disk(s) dialog box (Figure A–2) is displayed.

Figure A–2: Add Disk(s) Dialog Box



3. In the Add Disk(s) dialog box:
 - a. If the correct disk device name is not displayed in the Disk Device field, enter the disk device name or click on Browse to select the disk. You can enter more than one name separated by spaces.
 - b. Specify where to add disks:
 - To add disks to an existing disk group, select Existing Disk Group. Enter the disk group name in the Disk Group Name field or click on Browse to select a disk group.

- To add disks to a new disk group, select New Disk Group. Enter the name of the new disk group in the Disk Group Name field. The new disk group is created.
 - To place disks in the free disk pool, select Free Disk Pool. Disks in the free disk pool are under LSM control (initialized) but do not belong to a disk group and cannot be used to create volumes.
- c. In the Options section:
- i. Specify the LSM disk names for the disks by typing a disk name in the Disk Name(s) field. This name must be unique within the disk group. If no LSM disk name is specified, the Storage Administrator assigns a default name to the disk.
 - ii. Enter a comment if desired.
- d. Click on Ok.

A.3.2 Adding a Hot-Spare Disk

You can add one or more unused disks to a disk group and designate them as hot-spare disks. If an I/O failure occurs, the hot-spare feature automatically relocates any redundant (mirrored or RAID 5) subdisks to the hot-spare disk and restores the affected LSM objects and data. You are notified of the failure and relocation details by electronic mail. For more information on the hot-spare feature, see Section 3.5.

If you designate a hot-spare disk, provide at least one per disk group. In the event of disk failure, the hot-spare disk automatically replaces the failed disk. Volumes can use hot-spare disks only from within the same disk group.

To add a disk as a hot-spare disk:

1. In the Object Tree, select Disk Groups and in the Object Table, select the LSM disk to be designated as a hot-spare disk.
2. From the Disk Groups menu, choose Properties.
3. In the Disk Properties window:
 - a. Select the General tab.
 - b. Select Spare.
 - c. Click on Ok.

Note

The Properties dialog box is associated with the disk you have highlighted when you choose Properties. If you

select a different disk from the Object Table, you must open a new Properties dialog box.

A.3.3 Moving Data Off an LSM disk

You can evacuate (or move) the contents of a disk to other disks in the same disk group if there is sufficient free space. If no target disk is specified, LSM uses available disks with sufficient free space. Evacuating a disk is useful in the event of disk failure.

If the disk being evacuated contains part of a mirrored, striped, or RAID 5 volume, do not move the contents of the disk to another disk containing a copy of the mirrored volume or part of the striped or RAID 5 volume.

To evacuate a disk:

1. In the Object Tree, select Disks and in the Object Table, select the disk that contains the objects and data to be moved.
2. From the Disks menu, choose Evacuate.

The Evacuate Disk dialog box (Figure A-3) is displayed.

Figure A-3: Evacuate Disk Dialog Box



3. In the Evacuate Disk dialog box:
 - a. If the correct disk name is not displayed in the Disk Name field, enter the disk name or click on Browse then click on the Object Tree to select the disk.
 - b. Enter the name of the target disk to which you want to move the contents of the evacuated disk or click on Browse then click on the Object Tree to select one or more target disks.

If you choose Browse, the total evacuated space (in kilobytes) is displayed so you can choose your target disk accordingly.
 - c. Click on Ok.

A.3.4 Mirroring an LSM disk

You can mirror all LSM objects on a disk to provide high availability for the volumes on that disk. Mirroring also improves read performance, because multiple reads to the same volume can be done simultaneously using the multiple copies of data.

If possible, mirror the objects to a disk on a different bus to reduce the risk of a single point of failure for the volumes.

To mirror all volumes on a disk:

1. In the Object Tree, select Disks and in the Object Table, select the disk that contains the volumes to be mirrored to another disk.
2. From the Disks menu, choose Mirror.

The Mirror Disk dialog box (Figure A–4) is displayed.

Figure A–4: Mirror Disk Dialog Box



3. In the Mirror Disk dialog box:
 - a. If the correct disk name is not displayed in the Disk Name field, enter the disk name or click on Browse to select the disk.
If you choose Browse, the mirrored total space is displayed so you can choose your target disk accordingly.
 - b. To specify the disks to contain the new mirrors, enter the target disk name or click on Browse and complete the Target Disk dialog box.
 - c. Click on Ok.

A.3.5 Placing an LSM Disk Off Line

You can place a disk off line to prevent LSM from accessing it. You must remove a disk from its disk group before you place it off line. An offline disk remains unavailable until you restore access to the disk by placing it on line.

You place a disk off line to protect it from unintentional use; for example, if attempts to access it might have a negative effect on the system. You cannot place a disk that is in use off line.

To place a disk off line:

1. In the Object Tree, select Disks and in the Object Table, select the disk to be placed off line.
2. From the Disks menu, choose Offline.
The Offline Disk dialog box is displayed.
3. In the Offline Disk dialog box:
 - a. If the correct disk name is not displayed in the Disk Name field, enter the disk name or click on Browse to select the disk.
 - b. Click on Ok.

A.3.6 Placing an LSM disk On Line

Placing a disk on line restores access to a disk that is off line. The disk is placed in the free disk pool and is accessible to LSM again. After placing a disk back on line, the disk must be added to a disk group before it can be used for volumes.

Only disks that are off line can be placed on line.

To place a disk on line:

1. In the Object Tree, select Disks and in the Object Table, select the disk to be placed on line.
2. From the Disks menu, choose Online.
The Online Disk dialog box is displayed.
3. In the Online Disk dialog box:
 - a. If the correct disk name is not displayed in the Disk Name field, enter the disk name or click on Browse to select the disk.
 - b. Click on Ok.

A.3.7 Recovering LSM Volumes on an LSM disk

A recovery operation depends on the types of volumes on the disk and includes starting disabled volumes, resynchronizing mirrors in mirrored volumes, and resynchronizing parity in RAID 5 volumes.

Alert icons and the Alert Monitor window might indicate when a volume recovery is needed.

If recovery of a volume is not possible, restore the volume from backup.

To recover all volumes on a disk:

1. In the Object Tree, select Disks and in the Object Table, select the disk that contains the volumes to be recovered.
2. From the Disks menu, choose Recover.
The Recover Disk dialog box is displayed.
3. In the Recover Disks dialog box:
 - a. If the correct disk name is not displayed in the Disk Name field, enter the disk name or click on Browse to select the disk.
 - b. Click on Ok.

A.3.8 Removing LSM Disks from Disk Groups

An LSM disk no longer in use can be removed from a disk group.

Caution

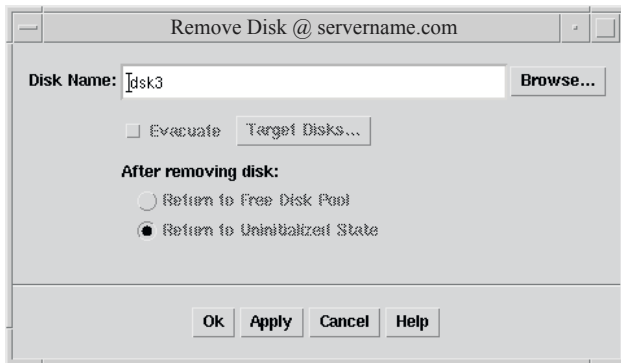
Do not remove LSM disks that are currently in use (for example, disks that contain subdisks for a volume); doing so can result in loss of data or of data redundancy.

After an LSM disk is removed from a disk group, it is still initialized for use with the LSM software. Therefore, after removing the disk from a disk group, it can be either immediately added to another disk group, removed from LSM, or left for later use.

To remove an LSM disk from a disk group:

1. In the Object Tree, select Disks and in the Object Table, select the disk to be removed.
2. From the Disks menu, choose Remove.
The Remove Disk dialog box (Figure A-5) is displayed.

Figure A–5: Remove Disk Dialog Box



3. In the Remove Disk dialog box:
 - a. If the correct disk name is not displayed in the Disk Name field, enter the disk name or click on Browse to select the disk.
 - b. To move the contents of the disk to another disk before the disk is removed, select Evacuate. Click on Target Disks to specify one or more disks to which you want the contents moved.
 - c. Specify how to handle the disk after removal:
 - To remove the disk from its disk group and place it in the free disk pool, select Return to Free Disk Pool. The disk remains under LSM control.
 - To remove the disk from LSM control, select Return to Uninitialized State.
 - d. Click on Ok.

A.3.9 Renaming an LSM disk

Because disk access names are defined by the operating system and disk media names are defined by you, you can rename only disk media names for disks in a disk group.

To rename the disk media name for an LSM disk:

1. In the Object Tree, select Disks and in the Object Table, select the disk to be renamed.
2. From the Disks menu, choose Rename.
The Rename Disk dialog box is displayed.

3. In the Rename Disk dialog box:
 - a. If the correct disk name is not displayed in the Disk Name field, enter the disk name or click on Browse to select the disk.
 - b. Enter the new LSM disk name.
 - c. Click on Ok.

A.3.10 Replacing an LSM disk

You might need to replace a disk if the disk fails and needs to be removed and repaired. You can replace an existing disk with a new physical disk, move volumes to the new disk, and attempt to recover any redundant (mirrored or RAID 5) volumes on the disk. You cannot recover nonredundant volumes. You must restore nonredundant volumes from backup. If the disk being replaced is a boot disk, you can set up the new disk as a boot disk.

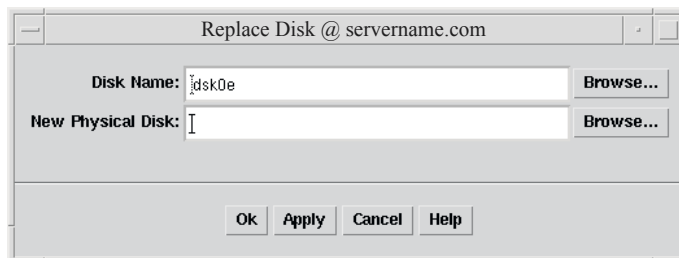
If you replace a good disk, you need to remove the disk from its disk group before you replace the disk. If you replace a disk that has failed and is disconnected, you do not need to remove the disk from the disk group.

To replace a disk:

1. In the Object Tree, select Disks and in the Object Table, select the disk to be replaced.
2. From the Disks menu, choose Replace.

The Replace Disk dialog box (Figure A–6) is displayed.

Figure A–6: Replace Disk Dialog Box



3. In the Replace Disk dialog box:
 - a. If the correct disk name is not displayed in the Disk Name field, enter the LSM disk name for the disk to be replaced or click on Browse to select the disk.
 - b. Enter the physical disk name for the new (replacement) disk or click on Browse to select a disk.
 - c. Click on Ok.

A.3.11 Scanning for New Disks

You can search your configuration for disks that are not under LSM control. Disks that are found are added to the free disk pool.

To scan for a new disk:

1. Select a disk from the Object Table.
2. From the Disks menu, choose Add.
3. Click on Scan Disk.
4. To view disks that are found, click on Free Disk Pool in the Object Tree.

A.4 Managing Disk Groups

The following sections describe disk group management tasks that you can complete using the Storage Administrator.

A.4.1 Creating Disk Groups

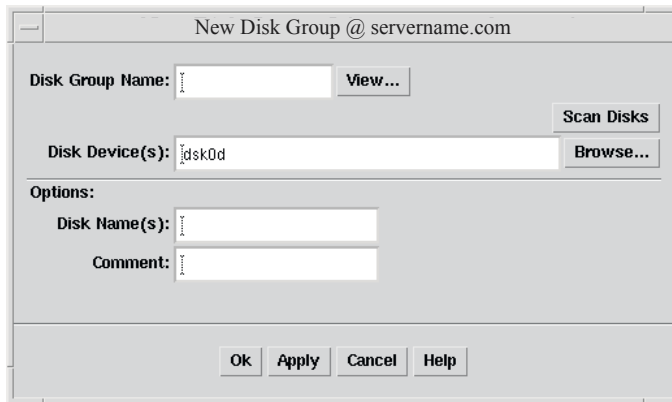
You must place disks into a disk group before you can use them to create volumes. The default disk group (`rootdg`) is created during LSM installation and always exists on a system running LSM. You can create additional disk groups to organize your disks into logical sets.

Each new disk group must contain at least one disk and must have a unique name. You can use only disks that are on line and do not already belong to a disk group.

To create a disk group:

1. From the Console menu, choose New then Disk Group.
The New Disk Group dialog box (Figure A-7) is displayed.

Figure A–7: New Disk Group Dialog Box



2. In the New Disk Group dialog box:
 - a. Enter the name of the disk group to be created. Click on View to view the names of existing disk groups.
 - b. To set up any new disks on the system, click Scan Disks. This runs the disk setup commands appropriate for the operating system.
 - c. Enter the disk devices to be placed in the new disk group or click on Browse to select the devices.
 - d. There are two options:
 - To specify the LSM disk name for the disk, enter a disk name in the Disk Name(s) field. If no LSM disk name is specified, the Storage Administrator assigns a default name to the disk.
 - Enter a comment if desired.
 - e. Click on Ok.

A.4.2 Adding LSM Disks to Disk Groups

To add a disk to a disk group, follow the instructions for adding a disk (Section A.3.1). In the Add Disk dialog box, specify an existing disk group.

The LSM disk name must be unique within the disk group. If multiple disks are specified in the Disk Device(s) field and only one disk name is specified in the Disk Name(s) field, LSM appends numbers to the disk name so that each disk name is unique within its disk group.

You must place disks that belong to a disk group in the free disk pool before you can add them to another disk group. You must add disks in the free disk pool to a disk group before you can use them to create volumes.

Disks must be on line before you can add them to a disk group or the free disk pool. You cannot add disks to deported disk groups.

You must place the root disk in the root disk group (`rootdg`). If the root disk is placed in any other disk group, you cannot use the root disk to boot the system.

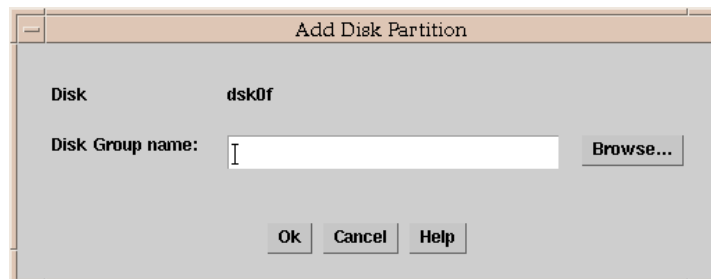
A.4.3 Adding Disk Partitions to Disk Groups

To add a disk partition to an LSM disk group:

1. In the Object Tree, select Free Partitions and in the Object Table, select the partition to be added.
2. From the Free Partitions menu, choose Add.

The Add Disk Partition dialog box (Figure A–8) is displayed.

Figure A–8: Add Disk Partition Dialog Box



3. Enter the name of the Disk Group to which you will add a partition or click on Browse to select a name.
4. Click on Ok.

A.4.4 Deporting Disk Groups

After a disk group is created, the LSM software automatically imports it for use whenever the system is booted.

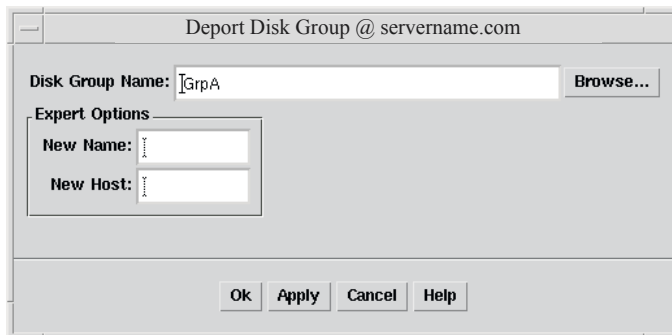
To disable access to a disk group, you deport the disk group. You must stop all the volumes within the disk group before deporting it.

To deport a disk group:

1. In the Object Tree, select Disk Groups and in the Object Table, select the disk group to deport.
2. From the Disk Groups menu, choose Deport.

The Deport Disk Group dialog box (Figure A–9) is displayed.

Figure A–9: Deport Disk Group Dialog Box



3. In the Deport Disk Group dialog box:
 - a. If the correct disk group name is not displayed in the Disk Group Name field, enter the disk group name or click on Browse to select the disk group.
 - b. Use the following Expert Options with caution:
 - To change the name of the disk group at deport, enter a new disk group name in the New Name field.
 - To set up a host machine to import the deported disk group at restart, enter the host ID in the New Host field.
 - c. Click on Ok.

A.4.5 Importing a Deported Disk Group

You can import a disk group to make a deported (inaccessible) disk group and its volumes accessible again. To import a deported disk group, you must know the disk group's former name. This disk group name must have remained unused. In addition, at least one disk formerly assigned to the deported disk group must remain unused. If all disks associated with a deported disk group were reused because the disk group was deported, that disk group cannot be imported.

The import might fail for a number of reasons; for example, if the host cannot find one or more disks in the disk group. If the import fails because a disk has failed, you can import the disk group by selecting the Force Import Expert option. If the import fails for another reason, a forced import can cause serious problems.

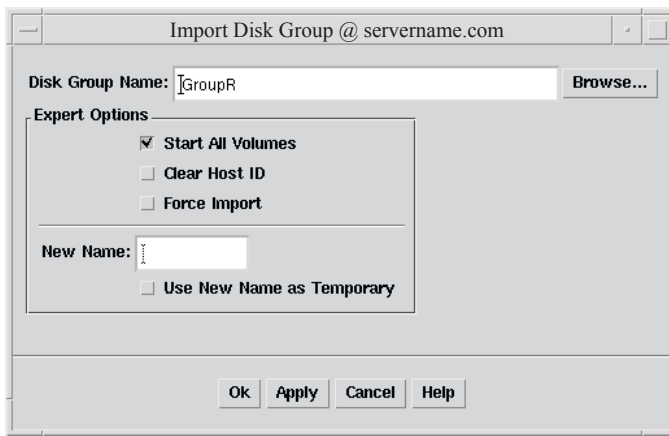
When you import a disk group, the system stamps its host ID on all disks in the disk group. A disk group import fails if one of the disks is stamped with a host ID that does not match the others. This ensures that dual-ported disks cannot be managed (and possibly corrupted) by two systems at the

same time. If you are sure that the disk group is not in use by another host, you can clear the host IDs and import the disk group by selecting the Clear Host ID Expert option.

To import a deported disk group:

1. In the Object Tree select Disk Groups.
2. From the All Disk Groups menu, choose Import Disk Group.
The Import Disk Group dialog box (Figure A–10) is displayed.

Figure A–10: Import Disk Group Dialog Box



3. In the Import Disk Group dialog box:
 - a. If the correct disk group name is not displayed in the Disk Group Name field, enter the disk group name or click on Browse to select the disk group.
 - b. Use the following Expert Options with caution:
 - To start all volumes in the disk group at import, select Start All Volumes.
 - To clear the existing host ID stamp on all disks in the disk group at import, select Clear Host ID. Do not use this option if another host is using any disk in the disk group.
 - To force the disk group import when the host cannot access all disks in the disk group, select Force Import.
 - c. Enter the name of the disk group to be imported or click on Browse to select the disk group.
 - d. To change the name of the disk group at import, enter a new disk group name in the New Name field. To indicate that the name change is temporary, select Use New Name as Temporary. If you

indicate a temporary name change, the original name is restored when the system is restarted.

- e. Click on Ok.

A.4.6 Recovering LSM Volumes in Disk Groups

You can recover volumes in a disk group. The recovery operations depend on the types of volumes in the disk group and include starting disabled volumes, resynchronizing mirrors in mirrored volumes, and resynchronizing parity in RAID 5 volumes.

Alert icons and the Alert Monitor window might indicate when volume recovery is needed.

In some cases, recovery might not be possible. If the volume recovery fails, you can attempt to restore the volume from backup.

To recover all volumes in a disk group:

1. In the Object Tree, select Disk Groups and in the Object Table, select the disk group containing the volumes to be recovered.
2. From the Disk Groups menu, choose Recover.

In the Recover Disk Groups dialog box, click Yes to recover the volumes in the disk group.

A.4.7 Renaming Disk Groups

You can rename a disk group. The new disk group name must be unique. If volumes in the disk group are in use (mounted), the disk group cannot be renamed.

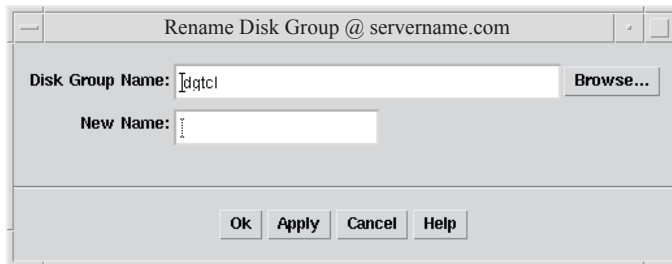
Renaming a disk group updates the `/etc/fstab` file.

To rename a disk group:

1. In the Object Tree, select Disk Groups and in the Object Table, select the disk group to be renamed.
2. From the Disk Groups menu, choose Rename.

The Rename Disk Group dialog box (Figure A–11) is displayed.

Figure A–11: Rename Disk Group Dialog Box



3. In the Rename Disk Group dialog box:
 - a. If the correct disk group name is not displayed in the Disk Group Name field, enter the disk group name or click on Browse to select the disk group.
 - b. Enter the new name for the disk group.
 - c. Click on Ok.

A.4.8 Moving Disk Groups to Another System

You can move a disk group (and LSM objects in that disk group) from one system to another. LSM and the Storage Administrator (server) must be running on both systems.

To move a disk group from one system to another:

1. Stop and unmount all volumes in the disk group to be moved.
2. Follow the instructions in Section A.4.4 to deport the disk group to be moved to the other system.
3. Attach all the physical disks in the disk group to the new system.
4. On the new system, follow the instructions in Section A.4.5 to import the disk group.
5. In the Object Tree, select the Storage Administrator.
6. From the System menu, choose Scan Disks to set up the newly attached disks on the system.
7. Follow the instructions in Section A.4.6 to restart and recover all volumes in the disk group on the new system.

A.4.9 Destroying Disk Groups

You can destroy a disk group permanently to remove the group from LSM control. This process reinitializes all the disks in the disk group as empty disks and places them in the free disk pool for reuse. You cannot destroy a

disk group if any volumes in that disk group are in use. When a disk group is destroyed, the volumes in the disk group are removed.

Destroy a disk group only if you are sure that you no longer need the volumes and data in the disk group. Because the last disk in an existing disk group cannot be removed, destroying a disk group is a way to free the last disk in a disk group for reuse.

You cannot destroy the `rootdg` disk group.

To destroy a disk group:

1. In the Object Tree, select Storage Administrator.
2. From the System menu, choose Destroy Disk Group.
The Destroy Disk Group dialog box is displayed.
3. In the Destroy Disk Group dialog box:
 - a. Enter the name of the disk group to be destroyed in the Disk Group Name field or click on Browse to select the disk group.
 - b. Click on Ok.

A.5 Managing Subdisks

Subdisks are created as the result of creating a volume. You cannot use the Storage Administrator to create subdisks.

The following sections describe the subdisk management tasks that you can complete using the Storage Administrator.

A.5.1 Joining Subdisks

You can join two or more subdisks to form a single, larger subdisk. Subdisks can be joined only if they belong to the same volume and occupy adjacent regions of the same disk and mirror. The joined subdisk can retain the name of one of the subdisks being joined.

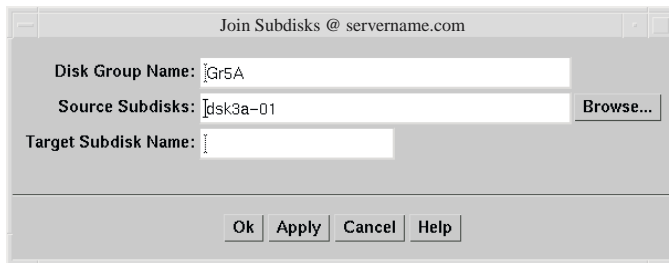
For a volume with a striped plex, the subdisks must be in the same column.

To join subdisks:

1. In the Object Tree, select Volumes and in the Object Table, select the volume with the subdisks to be joined.
2. From the Volumes menu, choose Show Layout.
3. In the Volume Layout Details window, hold down the Shift key and click to select the subdisks to be combined. Subdisks must be contiguous.
4. From the Subdisks menu, choose Join.

The Join Subdisks dialog box (Figure A–12) is displayed.

Figure A–12: Join Subdisks Dialog Box



5. In the Join Subdisks dialog box:
 - a. If the correct disk group name is not displayed in the Disk Group Name field, enter the name of the disk group that contains the subdisks to be joined.
 - b. If the correct subdisk names are not displayed, enter the subdisk names or click on Browse to select the subdisks. Specify at least two subdisk names separated by a space.
 - c. In the Target Subdisk Name field, enter the name of the new, combined subdisk.
 - d. Click on Ok.
6. Optionally, view the new volume layout by selecting Update from the View menu.
7. Close the Volume Layout Details window.

A.5.2 Splitting Subdisks

You can divide a subdisk into two or more subdisks. Once split, the new, smaller subdisks can be moved elsewhere or rejoined later. This feature is useful for reorganizing volumes or for improving performance. The original subdisk must contain a sufficient number of sectors for the specified split to work.

The name of the first new subdisk remains the same as the original subdisk. The other subdisks are automatically named by the Storage Administrator. The new, smaller subdisks occupy the same regions of the disk that the original subdisk occupied.

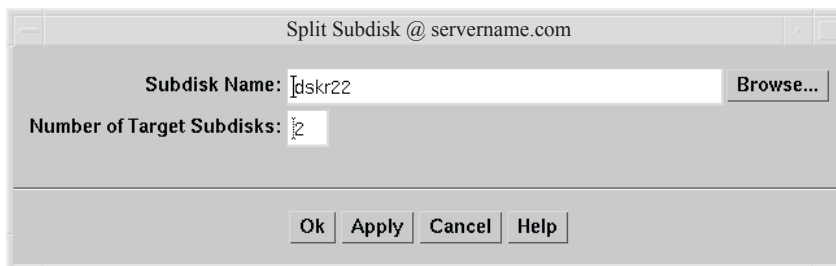
You cannot split a log subdisk.

To split a subdisk into multiple subdisks:

1. In the Object Tree, select Volumes and in the Object Table, select the volume with the subdisk to be split.
2. From the Volumes menu, choose Show Layout.
3. In the Volume Layout Details window, select the subdisk to be split into multiple subdisks.
4. From the Subdisks menu, choose Split.

The Split Subdisk dialog box (Figure A–13) is displayed.

Figure A–13: Split Subdisk Dialog Box



5. In the Split Subdisk dialog box:
 - a. If the correct subdisk name is not displayed, enter the subdisk name or click on Browse to select the subdisk.
 - b. In the Number of Target Subdisks field, enter the number of subdisks into which the subdisk will be split. A subdisk can be split into two or more subdisks.
 - c. Click on Ok.
6. Optionally, view the new volume layout by selecting Update from the View menu.
7. Close the Volume Layout Details window.

A.5.3 Moving Subdisks

You can move the data in subdisks to a different disk to improve performance. The disk space occupied by the original subdisk is returned to the free space pool.

Do not move a subdisk in a mirrored, striped, or RAID 5 volume to a disk that already contains a copy or part of that volume.

If the process of moving a subdisk leaves some unused subdisks (that is, subdisks that are not associated with a volume) on the system, you can

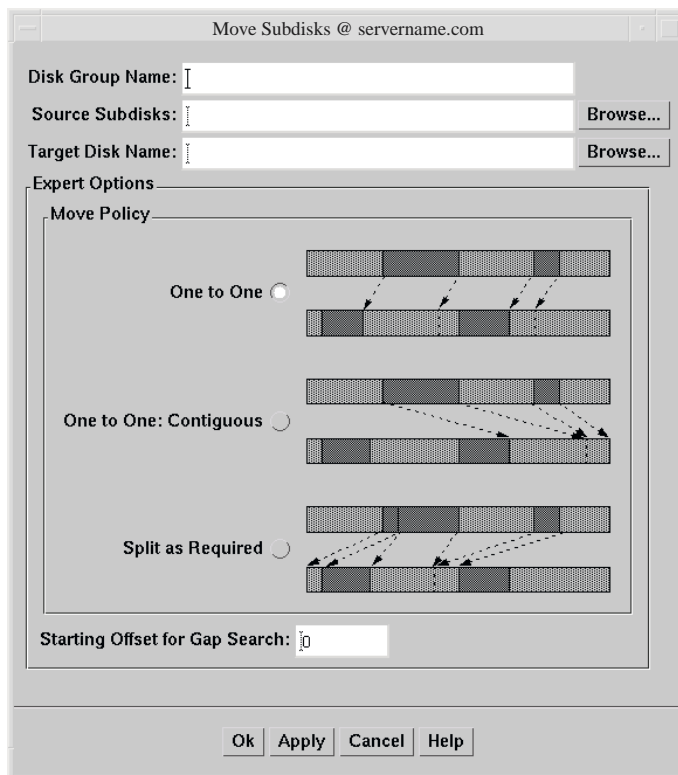
remove the subdisk (Section A.5.4) to free the space occupied by the unused subdisks.

To move a subdisk:

1. In the Object Tree, select Volumes and in the Object Table, select the volume with the subdisk to be moved.
2. From the Volumes menu, choose Show Layout.
3. In the Volume Layout Details window, select the subdisk to be moved to another disk.
4. From the Subdisks menu, choose Move.

The Move Subdisks dialog box (Figure A–14) is displayed.

Figure A–14: Move Subdisks Dialog Box



5. In the Move Subdisks dialog box:
 - a. If the correct disk group name is not displayed in the Disk Group Name field, enter the name of the disk group that contains the subdisk to be moved.

- b. If the correct source subdisk name is not displayed in the Source Subdisks field, enter the subdisk's name or click on Browse to select the subdisk.
 - c. In the Target Disk Name field, enter the name of the target disk to which the subdisk should be moved or click on Browse to select a disk.
 - d. Choose the Move Policy to specify whether the subdisk can be split into smaller subdisks that fit in available space(s) on the target disk. The One to One options do not split the subdisk. The Split as Required option allows the subdisk to be split if needed.
 - e. In the Starting Offset for Gap Search field, enter the minimum disk offset for the subdisk.
6. Click on Ok.
 7. Optionally, view the new volume layout by selecting Update from the View menu.
 8. Close the Volume Layout Details window.

A.5.4 Removing Subdisks

You can remove a subdisk that is not associated with a volume. The disk space occupied by the unused subdisks is returned to the free space pool.

To remove a subdisk:

1. In the Object Tree, select Disks and in the Object Table, select the disk with the subdisk to be removed.
2. From the Disks menu, choose Properties.
3. In the Disk Properties window, click the Subdisks tab.
4. Select the subdisk to remove.
5. From the Subdisk menu, choose Remove.
In the Remove Subdisks dialog box, click Yes to remove the subdisk.
6. Close the Disk Properties window.

A.6 Managing LSM Volumes

The following sections describe the volume management tasks that you can complete by using the Storage Administrator. Most tasks described in this section are appropriate only for UFS.

Note

AdvFS volumes are not the same as LSM volumes. This section deals with LSM volumes. For information about AdvFS volumes, see Section A.7.2.

In a cluster, AdvFS file systems are supported in all modes; UFS file systems are supported in read-only mode.

A.6.1 Creating LSM Volumes

You can create a volume that is less than or equal to the available free space on the LSM disks. If you do not specify the disks, the Storage Administrator uses any available space on disks in the selected disk group.

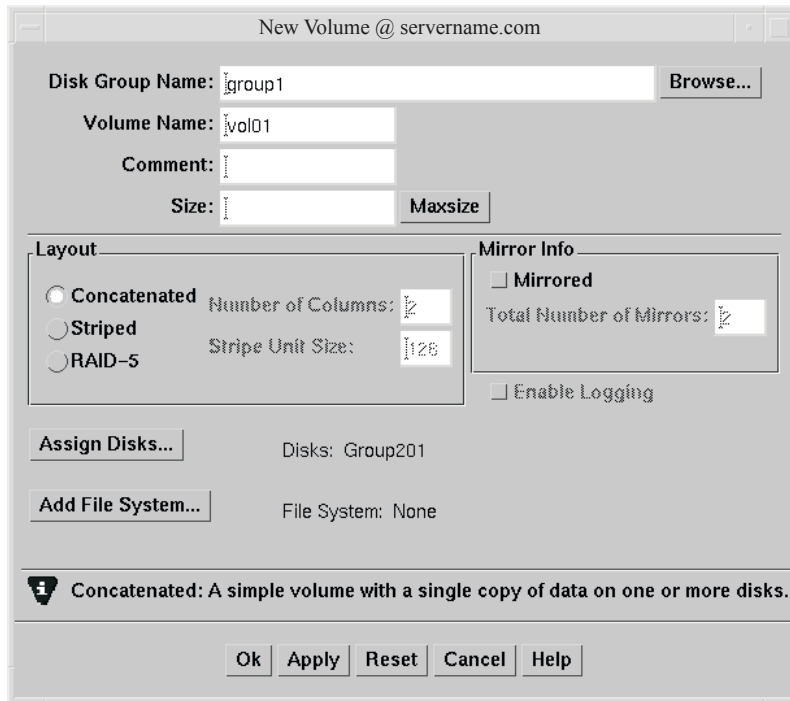
The data in a striped or concatenated volume is not protected against disk failure unless the volume is mirrored.

To create a volume:

1. From the Console menu, choose New then Volume.

The New Volume dialog box (Figure A-15) is displayed.

Figure A–15: New Volume Dialog Box



2. In the New Volume dialog box:
 - a. If the correct disk group name is not displayed, enter the disk group name or click on Browse to select the disk group.
 - b. Accept the default new volume name or enter a new volume name.
 - c. Enter a comment if desired.
 - d. Enter the volume size:
 - To specify a size unit, attach an *s* (sectors), *k* (kilobytes), *m* (megabytes), or *g* (gigabytes) to the size. The default size unit is sectors.
 - To display the largest possible size for the volume, click Maxsize. Units are displayed in kilobytes.
 - e. Choose the volume layout:
 - Concatenated
 - Striped — Enter the number of columns and stripe unit size.
 - RAID 5 — Enter the number of columns and stripe unit size. This option is not available in a cluster.

- f. If you have chosen a concatenated or striped volume, you can choose to mirror it.
 - i. To mirror the volume, select Mirrored.
 - ii. In the Total Number of Mirrors field, enter the total number of mirrors for the volume.
Each plex is a mirror, so if you create a volume with one mirror, the total number of plexes is 2.
- g. The Enable Logging box is selected by default when you create a mirrored or RAID 5 volume. To disable logging for the volume, deselect Enable Logging.
- h. To place the volume on a specific disk, click Assign Disks.
 - Select the disk you want to use from the Space Allocation–New Volume dialog box and click on Ok.
- i. To place a file system on the volume, click Add File System.
- j. In the Add File System dialog box:
 - i. Enter the mount point for the file system. The mount point must be an absolute pathname; that is, it must begin with root (/). If the path specified for the mount point does not exist, it is created.
 - ii. Select Mount at Boot if you want the `/etc/fstab` file automatically updated and the file system mounted at system startup.
 - iii. Select the file system type. If you select AdvFS, enter the Domain Name and the Fileset name. If you select UFS, you can also select Extra Options in the Newfs Details dialog box. This option is not available in a cluster environment.
 - iv. Click on Mount Details to make the file system read-only or to add extra options. Click on Help in the Mount Details dialog box for more information.
 - v. Click on Ok to close the Add File System dialog box.
- k. Click on Ok to close the New Volume dialog box.

A.6.2 Adding Mirrors to LSM Volumes

You can create a mirror (copy) of a volume on a disk that is not being used. When mirrored, the data in the volume is redundant. If a disk fails, the data remains available on another mirror. A volume can have multiple mirrors, but each mirror must use different disks. Sufficient disk space must be available. You cannot mirror a RAID 5 volume.

You can use only disks in the same disk group to create a mirror. If you do not specify the disks, LSM uses available disk space to create the mirror. Adding a mirror requires resynchronization, so this task might take some time.

A volume can contain up through 32 plexes, which can be any combination of mirrors and logs.

To add one or more mirrors to an existing volume:

1. In the Object Tree, select Volumes and in the Object Table, select the volume to be mirrored.
2. From the Volumes menu, choose Mirror then Add.

The Add Mirror dialog box (Figure A-16) is displayed.

Figure A-16: Add Mirror Dialog Box



3. In the Add Mirror dialog box:
 - a. If the correct volume name is not displayed in the Volume Name field, enter the volume name or click on Browse to select the volume.
 - b. Click on Layout to:
 - Specify the layout for the mirror (concatenated or striped) and, if striped, the stripe unit size.
 - Add more than one mirror and supply comments.
 - c. Click on Assign Disks to place the mirror on a specific disk.
 - d. Click on Ok.

A.6.3 Adding Logs to LSM Volumes

When you add a log to a mirrored volume, dirty region logging (DRL) is activated for that volume. The DRL plex tracks the regions of the volume that change due to I/O writes. If a system failure occurs, LSM uses the information in the DRL plex to recover only the portions of the volume that need recovery. This speeds up recovery time for mirrored volumes.

You can create additional DRL plexes (on different disks) to mirror the DRL information.

When you add a log to a RAID 5 volume, LSM maintains in the log a copy of the data and parity being written to the volume at any given time. The RAID 5 log speeds up the resynchronization time after a system failure. If a system failure occurs, when the system starts up LSM copies the data and parity that was being written at the time of failure from the log to the appropriate areas of the RAID 5 volume.

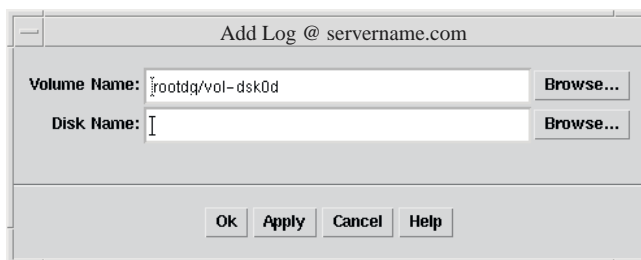
You can create multiple RAID 5 logs (on different disks) to mirror the log information. Ideally, each RAID 5 volume should have at least two logs.

To add a log to a volume:

1. In the Object Tree, select Volumes and in the Object Table, select the volume to contain the log.
2. From the Volumes menu, choose Log then Add.

The Add Log dialog box (Figure A-17) is displayed.

Figure A-17: Add Log Dialog Box



3. In the Add Log dialog box:
 - a. If the correct volume name is not displayed in the Volume Name field, enter the volume name or click on Browse to select the volume.
 - b. To place the log on a specific disk, enter the name of the disk in the Disk Name field or click on Browse to select a disk.
 - c. Click on Ok.

A.6.4 Mounting UFS File Systems on LSM Volumes

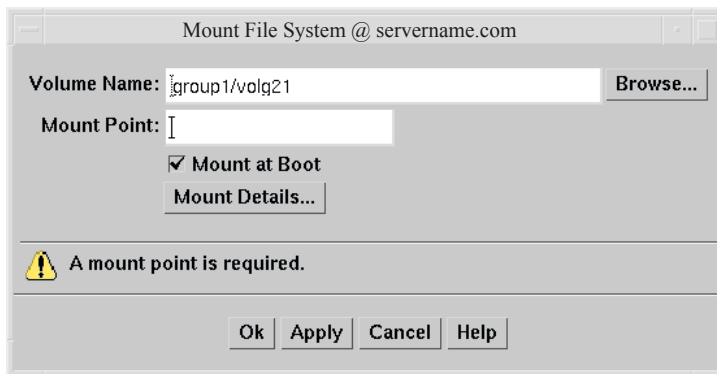
You can mount an existing UFS file system on a volume. You can choose to have the `/etc/fstab` file automatically updated.

To mount a UFS file system on an existing volume:

1. In the Object Tree, select Volumes and in the Object Table, select the volume on which to mount the UFS file system.
2. From the Volumes menu, choose File System then Mount.

The Mount File System dialog box (Figure A–18) is displayed.

Figure A–18: Mount File System Dialog Box



3. In the Mount File System dialog box:
 - a. If the correct volume name is not displayed in the Volume Name field, enter the volume name or click on Browse to select the volume.
 - b. In the Mount Point field, enter the mount point for the file system. The mount point must be an absolute pathname; that is, it must begin with root (/). If the path specified for the mount point does not exist, it is created.
 - c. Select Mount at Boot if you want the `/etc/fstab` file updated and the file system mounted at system startup.
 - d. Click on Mount Details to make the file system read-only or to add extra options. Click on Help in the Mount Details dialog box for more information.
 - e. Click on Ok.

A.6.5 Adding File Systems to LSM Volumes

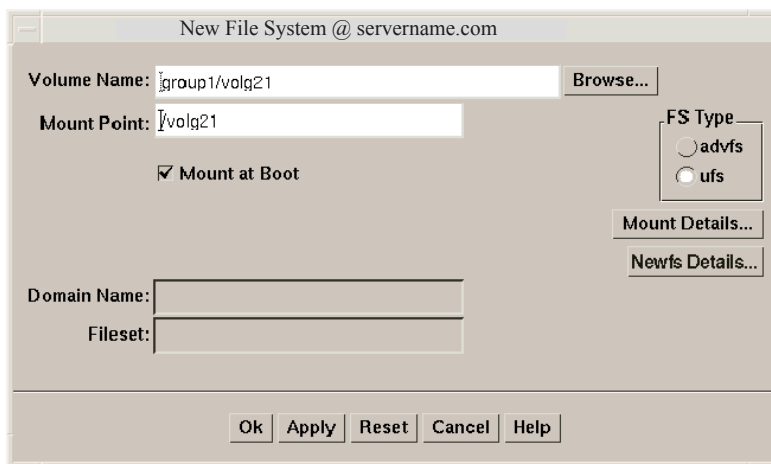
You can place a new file system on an existing volume and mount the file system. You can choose to have the `/etc/fstab` file automatically updated.

To add a file system to an existing volume:

1. In the Object Tree, select Volumes and in the Object Table, select the volume to contain the file system.

2. From the Volumes menu, choose File System then New.
The New File System dialog box (Figure A–19) is displayed.

Figure A–19: New File System Dialog Box



3. In the New File System dialog box:
 - a. If the correct volume name is not displayed in the Volume Name field, enter the volume name or click on Browse to select the volume.
 - b. In the Mount Point field, enter the mount point for the file system. The mount point must be an absolute pathname; that is, it must begin with root (/). If the path specified for the mount point does not exist, it is created.
 - c. Select Mount at Boot if you want the `/etc/fstab` file updated automatically, and the file system mounted at system startup.
 - d. Select the file system type:
 - If you select AdvFS, enter the Domain Name and the Fileset name.
To add a volume to an existing AdvFS domain, you must have an AdvFS Advanced Utilities license. If you supply an existing domain name and do not have a license, an error message is displayed. You can create a new domain for the volume without the Advanced Utilities license.
 - If you select UFS, you can also select Extra Options in the Newfs Details dialog box. Click on Help in the Newfs Details dialog box for more information. This option is not available in a cluster.

- e. Click on Mount Details to make the file system read-only or to add extra options. Click on Help in the Mount Details dialog box for more information.
- f. Click on Ok.

A.6.6 Checking UFS File Systems

Checking a file system applies only to UFS file systems; AdvFS does not use the `fsck` utility. You can check the file system with or without repairing it. Checking the file system might take some time.

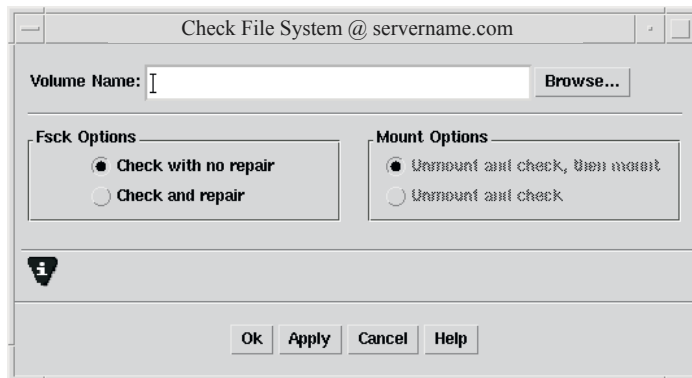
If you are running a cluster, UFS file systems are mounted read-only, so the Storage Administrator cannot check the file system.

To check a UFS file system on a volume:

1. In the Object Tree, select File System and in the Object Table, select the UFS file system to check.
2. From the File Systems menu, choose Check.

The Check File System dialog box (Figure A–20) is displayed.

Figure A–20: Check File System Dialog Box



3. In the Check File System dialog box:
 - a. If the correct file system name is not displayed in the File System Name field, enter the file system name or click on Browse to select the file system.
 - b. Choose the Fck option:
 - Check with no repair
 - Check and repair
 - c. Choose the Mount option:

- Unmount and check, then mount
 - Unmount and check
- d. Click on Ok.
- When the file system check is complete, the File System Check confirmation window appears indicating that the file system is okay.
- e. Click Close to close the File System Check window.

A.6.7 Unmounting UFS File Systems on LSM Volumes

You can unmount a mounted UFS file system on a volume.

To unmount a UFS file system on a volume:

1. In the Object Tree, select Volumes and in the Object Table, select the volume containing the file system to be unmounted.
2. From the Volumes menu, choose File System then Unmount.
The Unmount File System dialog box is displayed.
3. In the Unmount File System dialog box:
 - a. If the correct volume name is not displayed, enter the volume name or click on Browse to select the volume.
 - b. Click on Ok.

A.6.8 Mapping and Analyzing LSM Volumes, Disks, and Subdisks

One of the most important management features of the Storage Administrator is the ability to view a map of your volumes, disks, or subdisks. Using the Volume to Disk Mapping window (Figure A–21) to view volumes or their associated disks or subdisks can save you time, especially if you have a large number of volumes and disks.

You can also use the Volume to Disk Mapping window to display information about the performance of volumes, disks, and subdisks.

The statistical values for the performance data are represented by different colors. When the analysis starts or changes, the color behind the object changes. Clicking on any of the green dots in the table highlights the path between the volume and its related disks or subdisks. Statistics can be collected only on volumes. Only disks and subdisks associated with volumes can be analyzed.

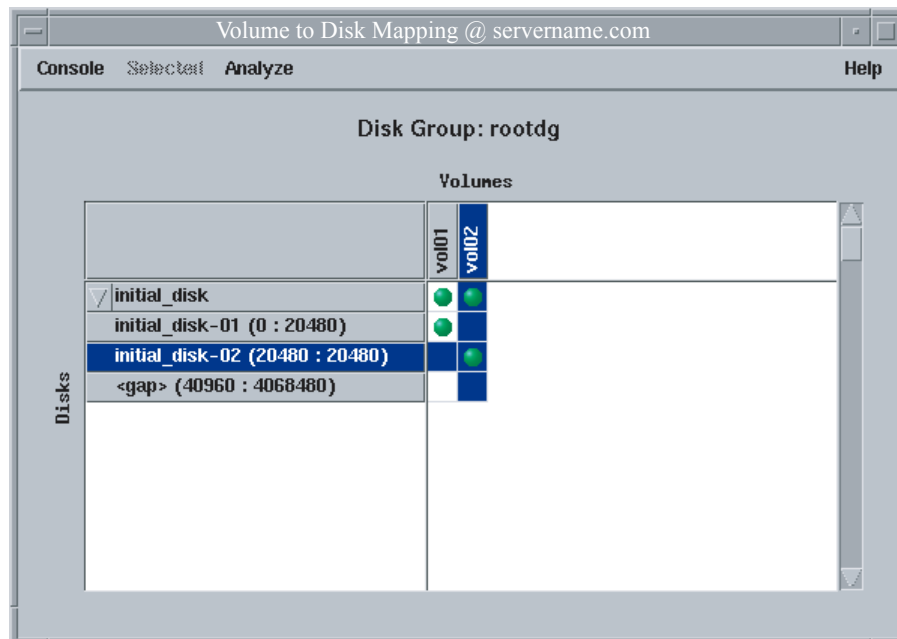
You can open Volume to Disk Mapping windows for more than one disk group; however, only one disk group can be analyzed at a time.

To open the Volume to Disk Mapping window:

1. In the Object Tree, select Disk Groups and in the Object Table, select the disk group to map.
2. From the Disk Groups menu, choose Disk/Volume Map.

The Volume to Disk Mapping window (Figure A–21) is displayed.

Figure A–21: Volume to Disk Mapping Window



3. In the Volume to Disk Mapping window:
 - To view the associated volumes, disks, or subdisks, click on the green dot at the intersection between the disk and volume.
 - To display all the subdisks and gaps on a particular disk, click the arrow to the left of the disk name.
 - To select a disk, subdisk or volume for analysis, click on the name of that object. The background for the object changes color when selected.
 - To analyze the use of an object, select a volume, disk, or subdisk and choose a command from the Analyze menu:
 - Start Analysis — Adds the selected item or items to the list of objects being analyzed. The background color of the selected items changes to show their state during analysis.

- Stop Analysis — The selected items return to their normal state (colors disappear), while all other items being analyzed continue to display their performance characteristics.
- Analyze All — Starts analysis on all volumes and LSM disks in a view.
- Stop All — Stops analysis for all items in all views. All items return to their normal state. (The colors disappear.)
- Parameters — Opens the Analysis Parameters dialog box, which lets you set the high and low threshold values for each object under analysis.

Note

When you choose the Start Analysis and Stop Analysis command on the Analyze menu, you must have selected an object. When you choose the Analyze All command you need not select an object.

- To see the Statistics form showing numerical equivalents for the colors, right-click on the volume or disk being analyzed.
- To use the pop-up menu to perform management tasks on volumes, disks, or subdisks, right-click on the disk or volume object.
- To close the Volume to Disk Mapping window, choose Close from the Console menu.

A.6.9 Disabling a Mirror in LSM Volumes

You can disable a mirror to temporarily detach the mirror from its volume. However, this can result in a loss of data redundancy, because the mirroring process is not occurring. A detached mirror is inaccessible for reads and writes but is still associated with its volume.

A disabled mirror remains detached from its volume until you either reattach the mirror or restart the volume. When a volume is restarted, any disabled (detached) mirrors are reattached to the volume automatically.

If a volume has two mirrors and one mirror is disabled, the volume is not redundant while the mirror is disabled.

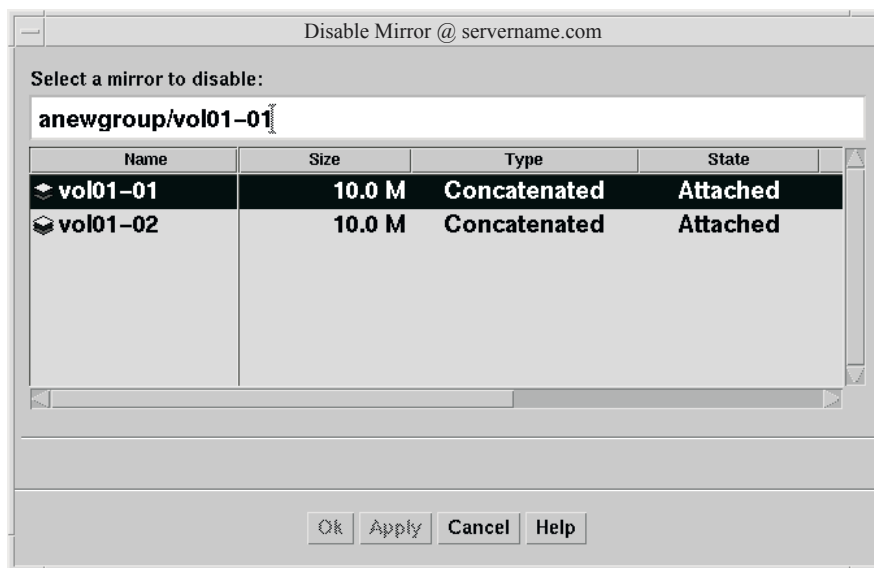
The last mirror in a volume cannot be disabled.

To disable a mirror in a volume:

1. In the Object Tree, select Volumes and in the Object Table, select the volume that contains the mirror to be disabled.

2. From the Volumes menu, choose Mirror then Disable.
The Disable Mirror dialog box (Figure A–22) is displayed.

Figure A–22: Disable Mirror Dialog Box



3. In the Disable Mirror dialog box:
 - a. Select a mirror to disable.
The mirror name appears in the Select a Mirror to Disable field.
 - b. Click on Ok.In the Disable Mirror dialog box, click Yes to disable the selected mirror.

A.6.10 Repairing Disabled Mirrors

You can repair a disabled mirror and reattach it to its volume. This process copies the data from an active mirror on the volume to the mirror being repaired and attaches the repaired mirror to the volume. An attached mirror is accessible for reads and writes. The mirror is recovered so that it has the same contents as other mirrors in the volume.

Alert icons and the Alert Monitor window might indicate when a mirror needs to be repaired.

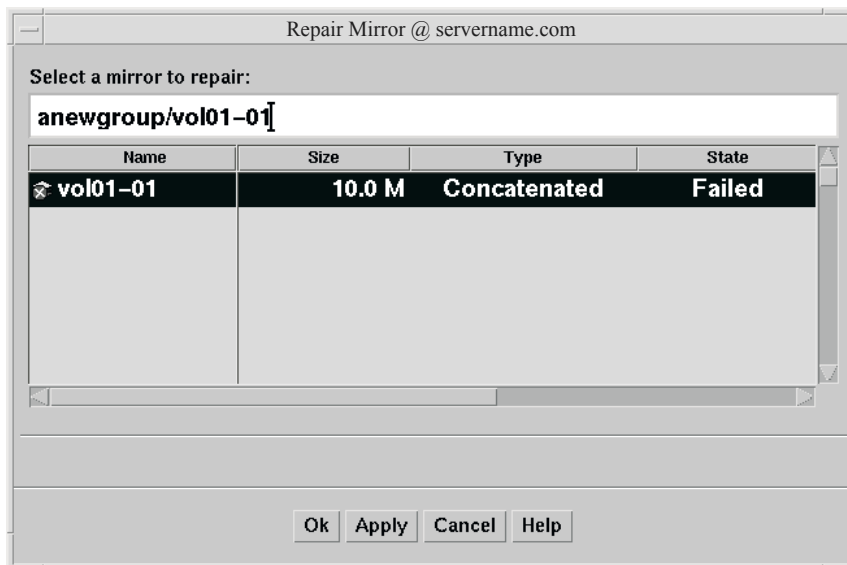
Depending on the amount of data in the volume, this task might take some time.

To repair a mirror:

1. In the Object Tree, select Volumes and in the Object Table, select the volume that contains the mirror to be repaired.
2. From the Volumes menu, choose Mirror then Repair.

The Repair Mirror dialog box (Figure A-23) is displayed.

Figure A-23: Repair Mirror Dialog Box



3. In the Repair Mirror dialog box:
 - a. Select the mirror to be repaired.
 - b. Click on Ok.

A.6.11 Creating Snapshots of Volume Data

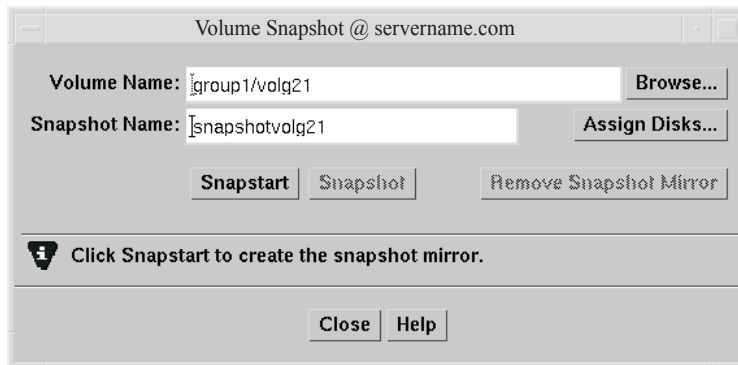
You can create a snapshot (temporary mirror) of a volume that you can then use to create a temporary volume for backup. You do not need to stop the original volume or place it off line. In a cluster, you cannot take a snapshot of the clusterwide root.

To create or stop a volume snapshot:

1. In the Object Tree, select Volumes and in the Object Table, select a volume for which to create or stop a snapshot.
2. From the Volumes menu, choose Snapshot.

The Volume Snapshot dialog box (Figure A-24) is displayed.

Figure A–24: Volume Snapshot Dialog Box



3. In the Volume Snapshot dialog box:
 - a. If the correct volume name is not displayed in the Volume Name field, enter the volume name or click on Browse to select the volume.
 - b. Accept the default snapshot name (volume) or enter a snapshot volume name.
 - c. Click on Assign Disks to select the disks for the snapshot volume.
4. To create the snapshot, start, or stop it:
 - Click on Snapstart to create the snapshot mirror.
 - Click on Snapstop to stop the snapshot process and create a new volume to attach to the snapshot mirror so you can access it.
 - Click on Remove Snapshot Mirror to remove the temporary volume that was created.

A.6.12 Recovering LSM Volumes

The volume recovery operations depend on the type of volume and include starting disabled volumes, resynchronizing mirrors in mirrored volumes, and resynchronizing parity in RAID 5 volumes. After successful recovery, the volume will be available for use.

Alert icons and the Alert Monitor window might indicate when a volume recovery is needed.

In some cases, recovery might not be possible. If the volume recovery fails, you can attempt to restore the volume from backup.

To recover a failed volume:

1. In the Object Tree, select Volumes and in the Object Table, select the volume to be recovered.

2. From the Volumes menu, choose Recover.
3. In the Recover Volume dialog box, click Yes to recover the volume.

A.6.13 Preparing LSM Volumes for Restoration

If a volume's data is corrupted and you need to restore the volume from backup, you must prepare the volume for restoration. To restore a volume from backup, you can stop the volume, set the volume to an uninitialized state, and restart the volume (without resynchronizing the volume's mirrors). This procedure will not work for an AdvFS file domain.

If the volume contains a mounted UFS file system, you must unmount the file system before you restore the volume from backup. This task does not remount the file system.

To prepare to restore a volume from backup:

1. In the Object Tree, select Volumes and in the Object Table, select the volume to be restored from backup.
2. From the Volumes menu, choose Prepare For Restore.
In the Prepare Volume For Restore dialog box, click Yes to prepare the volume for restoration.

A.6.14 Removing Mirrors from LSM Volumes

Removing a mirror from a volume breaks the link between the mirror and its volume and returns the mirror's disk space to the free space pool for reuse.

If a volume has only two mirrors and one mirror is removed, the volume is no longer redundant. This leaves the volume unprotected against disk failure. The last mirror cannot be removed from a volume, for that is equivalent to removing the volume.

To remove a mirror from a volume:

1. In the Object Tree, select Volumes and in the Object Table, select the volume that contains the mirror to be removed.
2. From the Volumes menu, choose Mirror then Remove.
The Remove Mirror dialog box is displayed.
3. In the Remove Mirror dialog box, select a mirror to remove and click Ok.

A.6.15 Removing Logs from LSM Volumes

You can remove a DRL plex or a RAID 5 log from a volume.

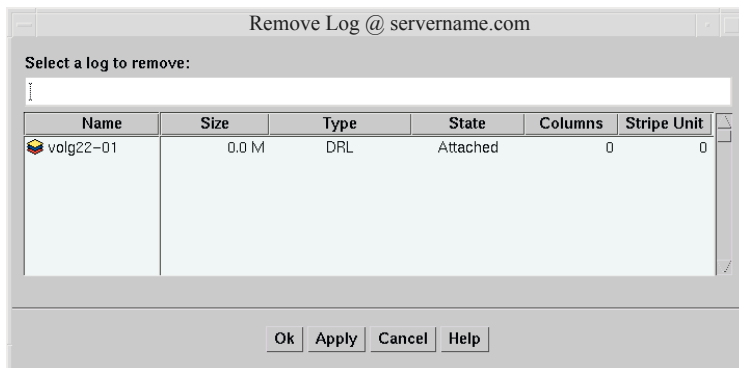
If you remove a volume's only log, logging (either DRL or RAID 5 logging) is no longer in effect for that volume. If logging is disabled, recovery time increases.

To remove a log from a volume:

1. In the Object Tree, select Volumes and in the Object Table, select the volume that contains the RAID 5 or DRL log to be removed.
2. From the Volumes menu, choose Log then Remove.

The Remove Log dialog box (Figure A-25) is displayed.

Figure A-25: Remove Log Dialog Box



3. In the Remove Log dialog box, select the log to remove and click on Ok.

A.6.16 Removing LSM Volumes

Removing a volume destroys all the data in that volume. Remove a volume only if you are sure that you do not need the data in the volume (or the data is backed up elsewhere). When a volume is removed, the space it occupied is returned to the free space pool.

Removing a volume that has a file system on it works only if the file system is UFS.

To remove a volume:

1. In the Object Tree, select Volumes and in the Object Table, select the volume to remove.
2. From the Volumes menu, choose Remove.

The Remove Volume dialog box is displayed.

3. In the Remove Volume dialog box, click Yes to remove the volume.

A.6.17 Renaming LSM Volumes

When you rename a volume, the new name must be unique within the disk group. If the volume has a file system, renaming the volume automatically updates the `/etc/fstab` file and lets you specify a new mount point for the file system. You cannot rename volumes that are part of an AdvFS domain.

To rename a volume:

1. In the Object Tree, select Volumes and in the Object Table, select the volume to be renamed.
2. From the Volumes menu, choose Rename.

The Rename Volume dialog box (Figure A–26) is displayed.

Figure A–26: Rename Volume Dialog Box



3. In the Rename Volume dialog box:
 - a. If the correct volume name is not displayed, enter the volume name or click on Browse to select the volume.
 - b. Enter the new name for the volume.
 - c. Click on Ok.

If the volume is open or mounted, the Open Volumes dialog box is displayed.

4. In the Open Volumes dialog box, click Yes to rename the volume.

A.6.18 Resizing LSM Volumes

You can increase or decrease the size of a volume, with the following exceptions:

- You cannot resize an AdvFS file domain with the Storage Administrator. If you want to resize a domain, use the AdvFS commands `addvol` and `rmvol`. For more information, see the *AdvFS Administration* manual.
- A volume containing an unmounted file system cannot be shrunk.

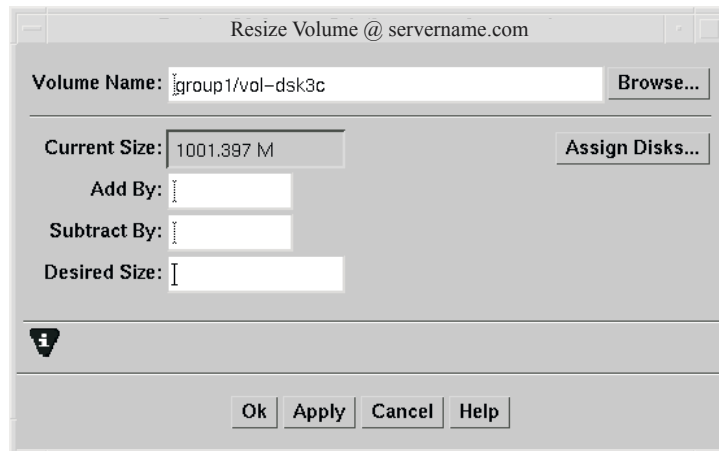
You can specify either the desired size or the amount of space to add to or subtract from the volume size. When a volume is shrunk, the resulting extra space is returned to the free space pool. To increase the volume size, sufficient disk space must be available. When increasing the size of a volume, LSM assigns the necessary new space from available disks.

To resize a volume:

1. In the Object Tree, select Volumes and in the Object Table, select the volume to be resized.
2. From the Volumes menu, choose Resize.

The Resize Volume dialog box (Figure A–27) is displayed.

Figure A–27: Resize Volume Dialog Box



3. In the Resize Volume dialog box:
 - a. If the correct volume name is not displayed, enter the volume name or click on Browse to select the volume.
 - b. To use a specific disk for the additional space, click Assign Disks and select the disk you want to use from the Space Allocation–Resize dialog box.
 - c. Specify one of the following:
 - To increase the volume size by a specific amount of space, use the Add By field to specify how much space to add.
 - To decrease the volume size by a specific amount of space, use the Subtract By field to specify how much space to remove.
 - To specify the new volume size, enter the size in the Desired Size field.

To specify a size unit, attach an *s* (sectors), *k* (kilobytes), *m* (megabytes), or *g* (gigabytes) to the size. The default unit is sectors.

4. Click on Ok.

A.6.19 Restarting LSM Volumes

Under normal circumstances, volumes are automatically started when the system restarts. You can restart a volume that you stopped manually or that was stopped in some other manner. If you cannot start a volume, the volume remains unusable. If the volume contains an AdvFS file domain, you cannot start it using the following procedure.

Restarting a RAID 5 volume enables the volume and resynchronizes parity, if necessary. Restarting a mirrored volume enables the volume and resynchronizes the mirrors to ensure that they are consistent. When a volume is successfully restarted, the volume is again available for use.

To restart a volume:

1. In the Object Tree, select Volumes and in the Object Table, select the stopped volume to be started.
2. From the Volumes menu, choose Start.
The Start Volume dialog box is displayed.
3. In the Start Volume dialog box:
 - a. If the correct volume name is not displayed, enter the volume name or click on Browse to select the volume.
 - b. Click on Ok.

The volume's state in the Object Table changes to Started.

A.6.20 Stopping LSM Volumes

When you stop a volume, it is not available for use until you restart it. You cannot stop a volume if it is in use or it has a mounted file system. If the volume contains an AdvFS file domain, you cannot stop it using the following procedure.

To stop a volume:

1. In the Object Tree, select Volumes and in the Object Table, select the volume to be stopped.
2. From the Volumes menu, choose Stop.
The Stop Volume dialog box is displayed.

3. In the Stop Volume dialog box:
 - a. If the correct volume name is not displayed, enter the volume name or click on Browse to select the volume.
 - b. Click on Ok.

The volume's state in the Object Table changes to Stopped.

A.7 Performing AdvFS Operations

The following sections describe Advanced File System (AdvFS) tasks that you can complete using the Storage Administrator.

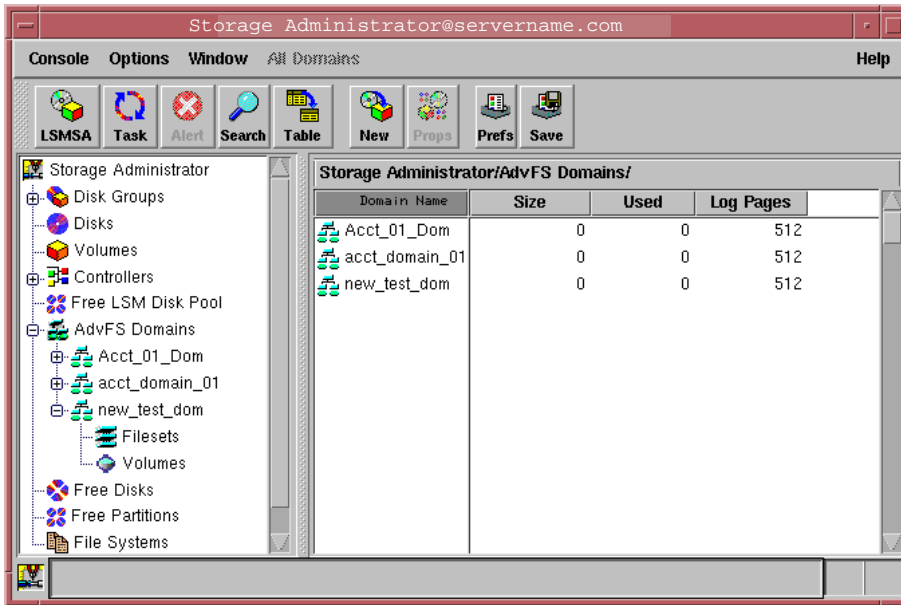
A.7.1 Creating AdvFS Domains

You can create an AdvFS domain to hold all disks or partitions on your system or create several domains to hold disks specific to particular applications. When you create a domain, you must specify one volume. An AdvFS volume can be created from a complete disk, a disk partition, or an LSM volume.

Existing data on the volume you assign to a new domain is destroyed when the domain is created.

Figure A-28 shows the main Storage Administrator main window with the AdvFS Domains entry fully expanded to include Filesets and Volumes.

Figure A–28: Storage Administrator Main Window

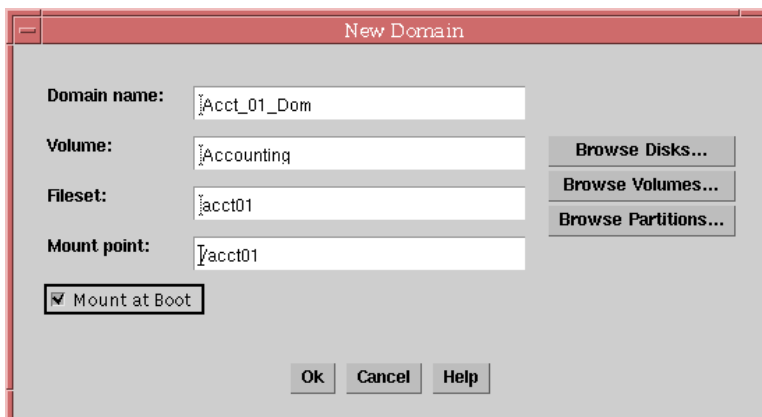


To create a domain:

1. In the Object Tree, select AdvFS Domains and from the All Domains menu choose New Domain.

The New Domain dialog box (Figure A–29) is displayed.

Figure A–29: New Domain Dialog Box



2. In the New Domain dialog box:
 - a. Enter the domain name. Characters invalid for domain names are white-space characters (tab, new line, space and so on) and punctuation marks and symbols (/ # : * ?).
 - b. Enter the volume name used to create the new domain, or click on Browse Disks, Browse Volumes, or Browse Partitions to select one.
 - c. Enter the fileset name. Characters invalid for fileset names are white-space characters (tab, new line, space and so on) and punctuation marks and symbols (/ # : * ?).
 - d. Enter the mount point for the fileset. The mount point must be an absolute pathname; that is, it must begin with root (/).
 - e. Select Mount at Boot if you want the fileset to be automatically mounted at system startup.
 - f. Click on Ok.

A.7.2 Adding LSM Volumes to AdvFS Domains

An AdvFS volume can be a single disk partition, an entire disk, or an LSM volume.

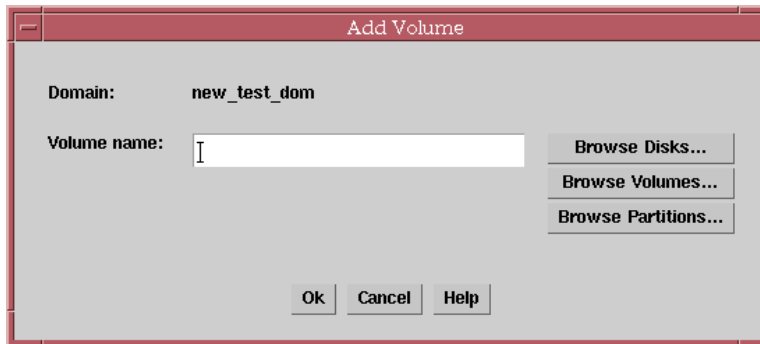
To see volumes already contained in a domain, open AdvFS Domains in the Main Window by selecting the plus sign, select the domain you want to work on, then select the volume icon.

If there are volumes within the AdvFS domain, they are displayed in the Object Table.

To add a volume to an AdvFS domain:

1. In the Object Tree, select AdvFS Domains.
2. Select the domain to contain the volume.
3. From the Domains menu, choose Add volume.
The Add Volume dialog box (Figure A-30) is displayed.

Figure A–30: Add Volume Dialog Box



4. In the Add Volume dialog box:
 - Enter the name of the storage device to be used to create the volume and click on Ok.
 - Click Browse Disks, Browse Volumes, or Browse Partitions to view a list of possible storage devices that can be used to create the volume.
5. Enter the new name for the volume to be added to the domain.
6. Click on Ok.

A.7.3 Adding Filesets to AdvFS Domains

You can add a fileset to an existing AdvFS domain. Fileset names must be unique within a domain. You can choose to have the `/etc/fstab` file updated automatically.

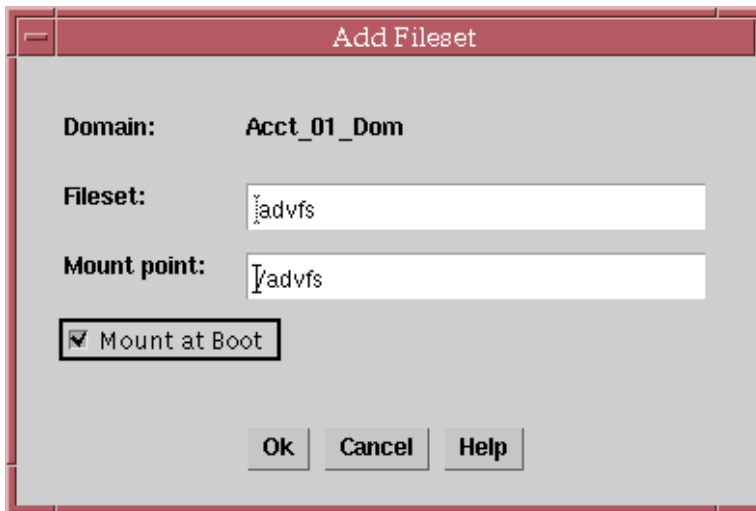
You can limit the number of files in a fileset by assigning quotas.

To add a fileset to an existing AdvFS domain:

1. In the Object Tree, select AdvFS domains and in the Object Table select the domain to which to add the fileset.
2. From the Domains menu, choose Add Fileset.

The Add Fileset dialog box (Figure A–31) is displayed.

Figure A–31: Add Fileset Dialog Box



3. In the Add Fileset dialog box:
 - a. In the Fileset field, enter the name of the fileset to add to the domain.
 - b. In the Mount Point field, enter the mount point for the fileset. The mount point must be an absolute pathname; that is, it must begin with /.
 - c. Select the Mount at Boot option if you want to update the /etc/fstab file and automatically mount the fileset at system startup.
 - d. Click on Ok.

A.7.4 Defragmenting AdvFS Domains

Before you can defragment an AdvFS domain, all filesets in the domain must be mounted. If you try to defragment an active domain that includes unmounted filesets, Storage Administrator displays an error message.

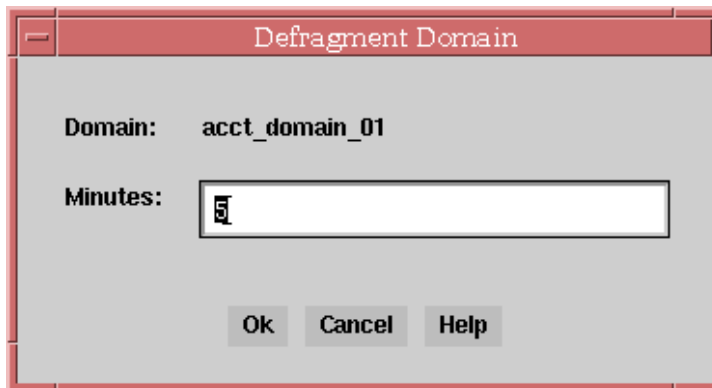
Defragment an AdvFS domain only when it is needed for system performance.

To defragment an AdvFS domain:

1. In the Object Tree, select AdvFS Domains and in the Object Table, select a domain to defragment.
2. From the Domains menu, choose Defragment.

The Defragment Domain dialog box (Figure A–32) is displayed.

Figure A–32: Defragment Domain Dialog Box



3. Enter the number of minutes for the defragment utility to run.
If the utility is still performing an operation when the specified time has elapsed, the procedure continues until the operation is complete.
4. Click on Ok.

A.7.5 Cloning AdvFS Filesets

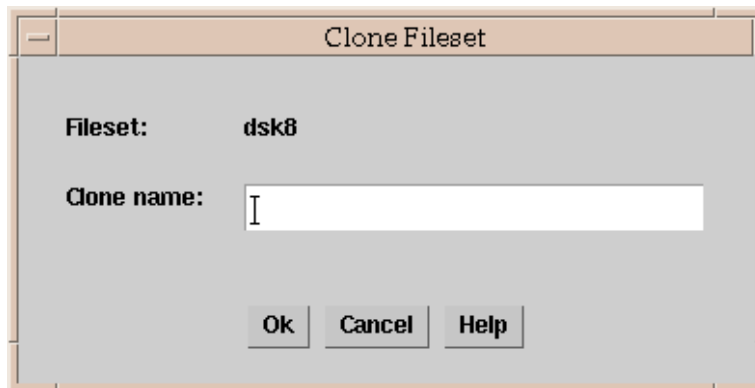
At any time, only one clone can be created for an AdvFS fileset. A RAID 5 volume cannot be cloned.

To clone an AdvFS fileset:

1. In the Object Tree, click the plus sign to open the AdvFS Domains, then click the plus sign to open the domain containing the fileset to clone.
2. In the Object Tree, select Filesets.
3. In the Object Table, select the fileset you want to clone.
4. From the Filesets menu, choose Clone.

The Clone Fileset dialog box (Figure A–33) is displayed.

Figure A–33: Clone Fileset Dialog Box



5. Enter a name for the cloned fileset.
6. Click on Ok.

A.7.6 Changing Fileset Quotas

You can change fileset quota limits. When a quota limit is reached, no more disk space allocations or file creations that would exceed the limit are allowed.

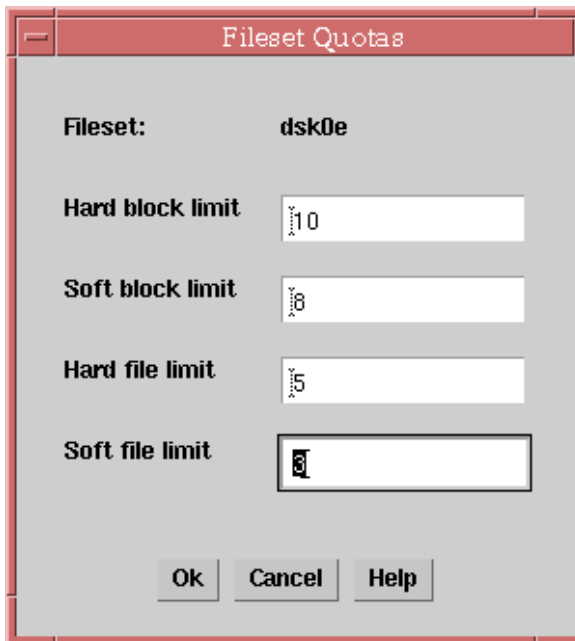
The quota numbers for block limits are displayed in kilobytes. Fileset quotas are not kept for the root user.

To change a fileset quota:

1. In the Object Tree, click the plus sign to open the AdvFS Domains, then click the plus sign to open the domain containing the fileset with limits to change.
2. In the Object Tree, select Filesets.
3. Select the fileset with limits to change.
4. From the Filesets menu, select Quotas.

The Fileset Quotas dialog box (Figure A–34) is displayed.

Figure A-34: Fileset Quotas Dialog Box



5. In the Fileset Quotas dialog box:
 - Enter the Hard and Soft Block limits in whole kilobyte increments
 - Enter the Hard and Soft file limits in whole number increments
6. Click on Ok.

A.7.7 Unmounting AdvFS Filesets

You must unmount an AdvFS fileset to rename the fileset or to resize the underlying LSM volume.

To unmount a fileset:

1. In the Object Tree, click the plus sign to open the AdvFS Domains, then click the plus sign to open the domain containing the fileset to unmount.
2. In the Object Tree, select Filesets.
3. In the Object Table, select the fileset to unmount.
4. From the Filesets menu, choose Unmount.
5. Verify that the name of the fileset is correct.
6. Click on Yes.

A.7.8 Renaming AdvFS Filesets

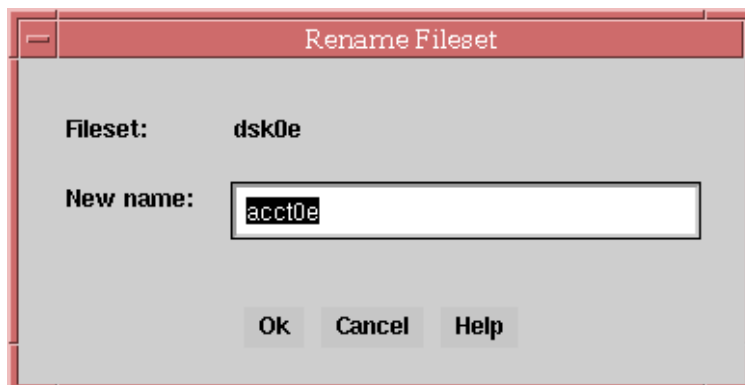
You can rename an AdvFS fileset. You must unmount the fileset before you can rename it. When you rename a fileset, the `/etc/fstab` file is updated.

To rename an AdvFS fileset:

1. In the Object Tree, click the plus sign to open the AdvFS Domains, then click the plus sign to open the domain containing the fileset to rename.
2. In the Object Tree, select Filesets.
3. In the Object Table, select the fileset you want to rename.

The Rename Fileset dialog box (Figure A–35) is displayed.

Figure A–35: Rename Fileset Dialog Box



4. Enter the new name for the fileset.
5. Click on Ok.

A.7.9 Mounting AdvFS Filesets

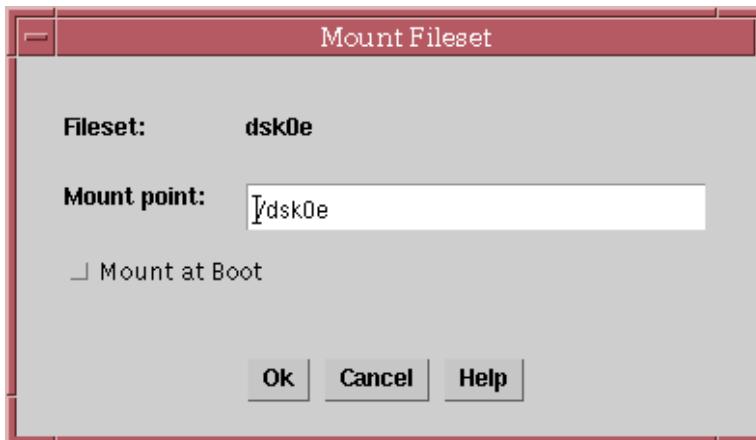
You can mount an AdvFS fileset that you unmounted to rename or to resize the underlying LSM volume.

To mount an AdvFS fileset:

1. In the Object Tree, click the plus sign to open the AdvFS Domains, then click the plus sign to open the domain containing the fileset to mount.
2. In the Object Tree, select Filesets.
3. In the Object Table, select the fileset to mount.
4. From the Filesets menu, choose Mount.

The Mount Fileset dialog box (Figure A–36) is displayed.

Figure A–36: Mount Fileset Dialog Box



5. In the Mount Fileset dialog box:
 - Select Mount at Boot if you want to automatically update the `/etc/fstab` file and mount the fileset at system startup.
 - Enter a mount point for the fileset. The mount point must be an absolute pathname; that is, it must begin with `/`.
6. Click on Ok.

A.7.10 Removing AdvFS Filesets

To remove an AdvFS fileset:

1. In the Object Tree, click the plus sign to open the AdvFS Domains, then click the plus sign to open the domain containing the fileset to remove.
2. In the Object Tree, select Filesets.
3. In the Object Table, select the fileset you want to remove.
4. From the Filesets menu, choose Remove.
The Remove Fileset window is displayed.
5. Click on Yes to remove the indicated fileset.

B

Understanding and Customizing the Storage Administrator GUI

This appendix describes how to track Storage Administrator activities, how to use the Storage Administrator dialog box, how to view objects and object properties, and how to use shortcuts to efficiently perform operations with the Storage Administrator GUI. For more information on using the Storage Administrator to complete a particular task for an LSM or AdvFS object, see Appendix A.

B.1 Tracking Storage Administrator Activities

Three log files keep track of the Storage Administrator:

- A command log tracks Storage Administrator tasks
- An access log tracks Storage Administrator logins
- A server log collects LSM and startup information

The same server log collects LSM and AdvFS information and error messages.

By default, a log maintenance shell script called `/usr/lib/java/applications/lmsa/logMaintenance` runs once a week to save and compress each log file. Compressed files are saved in a file called `logfilename.gz.X`, where X is the version number. Each week the previous week's saved file suffix is increased by one and a new `logfilename.gz.1` is created. Files are saved for ten weeks. You can change the number of files to save by editing the root `crontab` file.

B.1.1 Overview of the Command Log File

The command log file contains a description of each Storage Administrator task and information such as the user who performed the task, the task status, the start and finish times, and the commands used to perform the task. For failed tasks, the command log includes relevant error messages. By default, the command log is `/var/lmsa/logs/command`.

The following example shows a command log entry for a successful volume creation:

```
Create Volume
Description: Create Volume
User: root
Started: Fri Mar 09 12:07:22 PDT 2001
Finished: Fri Mar 09 12:07:24 PDT 2001
```

```
State: Successful
Executed Commands:
```

```
/usr/sbin/volassist
-g rootdg make vol04 4m layout=striped stripeunit=128 ncolumn=2
```

The following example shows a command log entry for a failed volume creation:

```
Create Volume FAILED!
Description: Create
VolumeUser: root
Started: Fri Mar 09 12:07:50 PDT 2001
Finished: Fri Mar 09 12:07:51 PDT 2001
State: Failed
```

```
Executed Commands:
/usr/sbin/volassist
-g rootdg make vol05 8g layout=striped stripeunit=128 ncolumn=2
```

```
Failed Command: /usr/sbin/volassist
-g rootdg make vol05 8g layout=striped stripeunit=128 ncolumn=2
```

```
Error Message: lsmsa:volassist: ERROR: Cannot allocate space
for 16777216 block volume
```

B.1.2 Overview of the Access Log File

You can monitor access to the Storage Administrator by reviewing the contents of the access log file. By default, the access log file is `/var/lsmsa/logs/access`.

The following example shows an access log file entry:

```
Mon Apr 02 12:07:22 PDT 2001: user rsn login succeeded
Mon Apr 02 12:22:24 PDT 2001: user jehg login failed with error
*User password invalid*
```

Entries for failed access might be logged multiple times due to a security requirement.

B.1.3 Overview of the Server Log File

The server log file tracks LSM startup information and server errors. By default, the server log file is `/var/lsmsa/logs/server.log`.

The following example shows server log file entries:

```
Starting Compaq Storage Administrator RMI Registry
Starting Compaq Storage Administrator Command Server
Starting Compaq Storage Administrator Server
Fri Mar 16 11:22:21 PST 2001
```

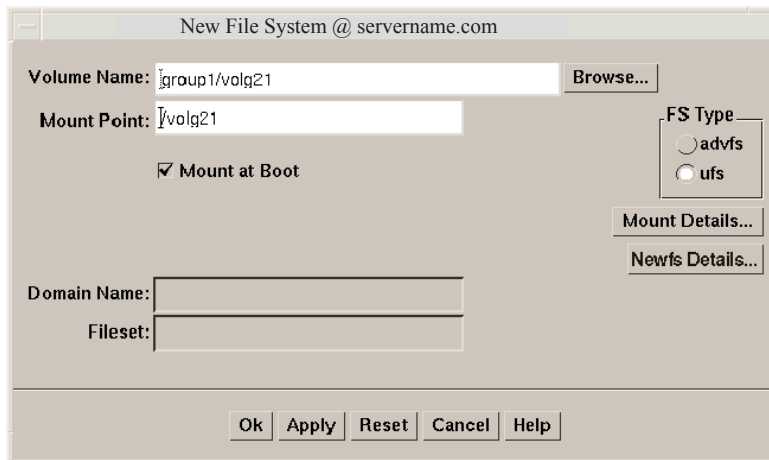
```
security enabled
rebinding ....
rebound

//servername:2410/vrts.remote.vrtsServer
```

B.2 Working with Dialog Boxes

The Storage Administrator displays dialog boxes in which you provide information as shown in Figure B–1. Dialog boxes can contain selectable buttons or fields in which you enter information. Some dialog box fields contain default values that you can change. Items that are not applicable are grayed out.

Figure B–1: Typical Storage Administrator Dialog Box



To use a dialog box, select the appropriate items or enter the appropriate information in a field, then click on one of the following buttons to initiate or cancel the task:

- **Ok** — Performs the current task and closes the dialog box.
- **Apply** — Performs the current task and continues to display the dialog box.
- **Cancel** — Closes the dialog box and cancels the current task. If you have already chosen **Apply**, use this button to close the dialog box. Doing so does not cancel the **Apply** request.
- **Reset** — Clears the information in dialog box fields.
- **Help** — Displays the Help menu.

B.2.1 Specifying Objects in Dialog Boxes

Most Storage Administrator dialog boxes contain one or more object name fields. If you select an object before you select the task, the resulting dialog box usually includes the selected object name. If the object name field is empty, you can specify an object using one of the following methods:

- Enter the object name.

In some cases, you can specify multiple objects (separated by a space) in a single field.

- Click on Browse next to the object name field, and then select the object from the resulting browse dialog box. Most browse dialog boxes display an Object Tree and Object Table. To select an object in a browse dialog box, click on an object group in the Object Tree, then click on the object in the Object Table.

Note

When you select an object for an action, the object name appears in the dialog box. This does not mean that the action you have chosen for that object is valid. If you try to complete an invalid operation on an object, an error message is displayed.

B.2.2 Specifying Object Sizes in Dialog Boxes

The following table lists the object sizes that you can enter to specify for an input field or a display size:

For	Enter
Sectors	s
Kilobytes	k
Megabytes	m
Gigabytes	g

By default, sectors are used for input fields if you do not specify an input size or if you did not change the default value by customizing the GUI (Section B.3.4). By default, sizes are displayed in kilobytes unless you specify otherwise.

B.3 Viewing Objects and Object Properties

There are several windows and dialog boxes that you can use to display information about and perform LSM operations.

B.3.1 Overview of the Main Window

The Object Tree and the Object Table track your LSM configuration. The Storage Administrator constantly monitors objects on the system and makes appropriate changes to the displays. You can view objects in the Object Tree and Object Table in the following ways:

- Click on the plus sign (+) or minus sign (–) next to an object to expand or collapse its hierarchy.
- Click on the object type in the Object Tree. All objects that belong to the selected object appear in the Object Table.

For example, to display all volumes in the `rootdg` disk group, expand the Disk Groups node (by clicking on the plus sign), expand the `rootdg` node, and click on the Volumes group under `rootdg`. Only volumes in the `rootdg` disk group appear in the Object Table.

- To display the components of an object in the Object Table, double-click on the object. All objects that belong to that object appear in the Object Table. If the object does not contain other objects, the Properties dialog box appears.

For example, to display the volumes in a disk group listed in the Object Table, double-click on the disk group name, then double-click on Volumes. All volumes in the disk group appear in the Object Table.

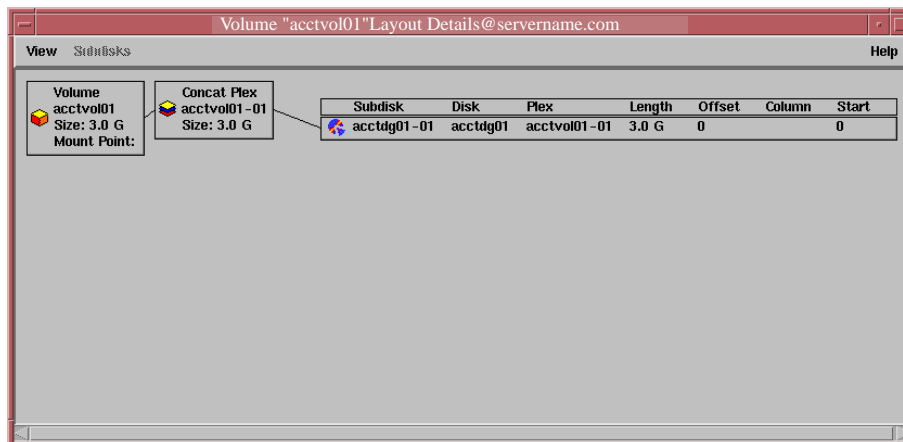
- It might be useful to view a copy of the Object Table to look at different objects; for example, disks and volumes. From the Window menu choose Copy Object Table. A dynamic copy of the Object Table appears in a separate window. The contents of the new Object Table window continue to update.

B.3.2 Viewing Volume Layout Details

To display the layout details for a volume, highlight the volume in the Main window Object Table and choose Show Layout from the Volumes menu.

The Volume Layout Details window displays a graphical view of the selected volume's layout, components, and properties, as shown in Figure B–2.

Figure B–2: Volume Layout Details Window



You can select objects or perform tasks on objects in the window. The Volume Layout Details window is not dynamic, so the objects displayed in the window are not automatically updated when the volume properties change. The View menu changes the display of the Volume Layout Details window.

- To refresh the display, choose Update from the View menu.
- To change the volume displayed, choose Open from the View menu and specify another volume in the Open Volume dialog box.
- To hide the detailed information within each object, choose Compress Display from the View menu. Click on an object to show its details in the compressed display.
- To highlight objects that are related to or part of a specific object, choose Projection on Selection from the View menu, then click on an object.
- To highlight any subdisks on the same disk as a specific subdisk, choose Subdisk Projection from the View menu, then click on a subdisk.

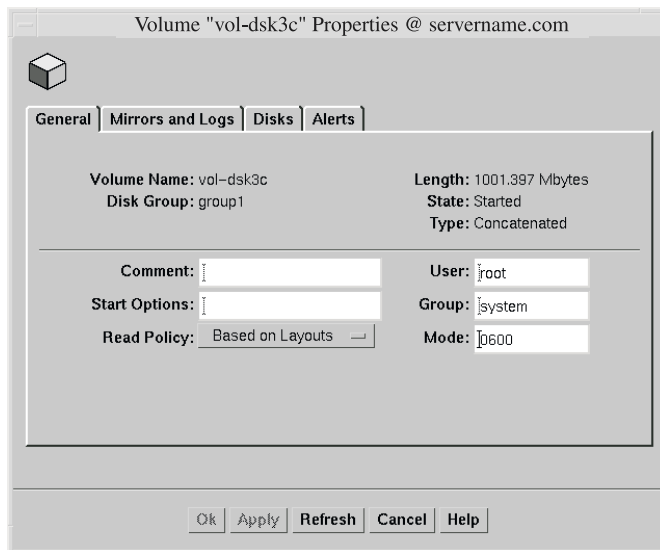
Right-click on an object to display its context-sensitive pop-up menu.

B.3.3 Overview of Object Properties Dialog Boxes

To view the properties of an object, click on the object in the Object Table and then choose Properties from the object's menu. If the object contains no other objects, double-click on the object to display its Properties dialog box.

The Object Properties dialog box displays detailed information specific to the selected object as shown in Figure B–3.

Figure B–3: Volume Properties Dialog Box



You can change some properties through this box. A set of tabbed pages provides information about the object and related objects. The tab labels and page contents vary, depending on the type of object selected. Click on Help for a detailed description of the Properties dialog box fields.

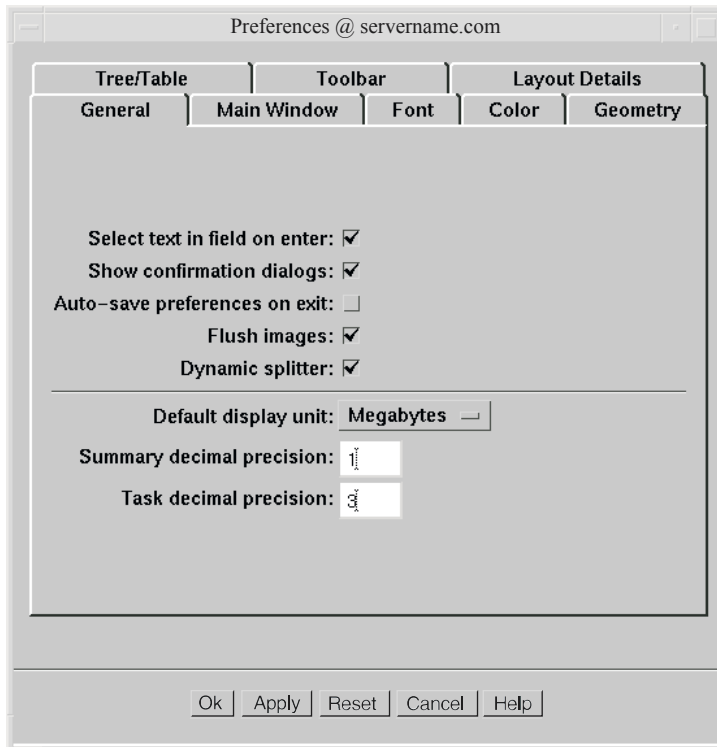
- To change items in the Properties dialog box, make the changes, then click on Ok. This changes the settings for all properties tabs in the Properties dialog box.
- To update the contents of the Properties dialog box to reflect current properties for the object, click on Refresh. If you select a different object while a Properties dialog box is open, the contents of the dialog box does not change to reflect the new object selected. You must choose Properties again and open another dialog box.

B.3.4 Overview of the User Preferences Dialog Box

You can change the way items appear in the Storage Administrator Main window and other windows. The Preferences dialog box (Figure B–4) contains a set of tabbed pages that display preference options for a particular aspect of Storage Administrator. You can customize settings for a single Storage Administrator session or save the settings for future sessions.

To display the Preferences dialog box, click on Prefs on the toolbar.

Figure B-4: Preferences Dialog Box



To change preference settings, make the appropriate selections in the dialog box, then click on **Ok**. This changes the settings for all tabs in the Preferences dialog box. To reset the values for all preferences to the previous settings, click on **Reset** before you click on **Ok**.

When you change preference settings, an asterisk appears on the tab for the page that contains changes. The asterisk disappears when you click on **Ok**, **Apply**, or **Reset**. When you click on **Apply** or **Reset**, an asterisk also appears in the **Help** bar status area in the Storage Administrator Main Window.

Unless you save your preferences, changes apply only to the current session. To save your settings, do one of the following:

- From the **Options** menu, choose **Save Preferences**
- Click on **Save** in the toolbar
- Click the asterisk in the **Help** bar status area

To reload your previously saved preferences, choose **Load Preferences** from the **Options** menu.

The Storage Administrator saves user preferences in the `user's_home_directory/.lsmsa/SAPreference.prf` file on the system where the client is running. If the autosave preference is set, the Storage Administrator saves all preference settings when you exit the Storage Administrator session.

B.3.4.1 General Preferences

The General tab sets the preferences for:

- **Select Text in Field on Enter**
Sets user input to replace mode. This highlights existing text in a field and replaces that text with the new text.
- **Show Confirmation Dialogs**
Shows or hides confirmation dialog boxes for tasks that might have serious consequences (such as data loss). Confirmation dialogs require you to confirm that a task be performed. Confirmation dialogs typically appear for tasks that remove objects. If you hide confirmation dialogs, most tasks are performed immediately and without any confirmation.
- **Auto-Save Preferences on Exit**
Saves all current user preferences when you exit the Storage Administrator.
- **Flush Images**
Draws images slightly slower than usual to prevent the X server from growing. This is recommended if you plan to run the Storage Administrator for long periods of time.
- **Dynamic Splitter**
Refreshes the contents of the window panes while the splitter is being moved to resize the panes.
- **Default Display Unit**
Sets the default size unit for areas that display object sizes. If Best Choice is set, the Storage Administrator uses an appropriate size unit.
- **Summary Decimal Precision**
Sets the decimal point precision for object sizes displayed in the Object Table and other areas that display summaries.
- **Task Decimal Precision**
Sets the decimal point precision for object sizes displayed in task-related dialog boxes and areas that display numerical information.

B.3.4.2 Main Window Preferences

The Main Window tab sets the preferences for:

- **Show Status Bar**
Shows or hides the status bar (at the bottom of the Main window). The Status bar displays alert icons when failures or errors occur.
- **Show Command Launcher**
Shows or hides the Command Launcher. The Command Launcher displays a list of selectable tasks. You can show or hide the Command Launcher by checking or unchecking the Command Launcher box on the Window menu of the Main window.
- **Dock Command Launcher**
Attaches or detaches the Command Launcher and the Main window.
- **Docked Command Launcher Height**
Sets the height of the Command Launcher portion of the Main window.

B.3.4.3 Font Preferences

The Font tab sets the font size, family, and style for:

- **User Font**
Sets the font for user input and objects displayed in the Object Tree and Object Table.
- **System Font**
Sets the font for the Storage Administrator labels, menus, and buttons.
- **Object Table Heading Font**
Sets the font for Object Table headings.
- **Object Table Heading Highlight Font**
Sets the font for the highlighted Object Table headings for sorting purposes.
- **Toolbar Font**
Sets the font for the toolbar buttons.
- **Graphical Display Font**
Sets the font for objects in the Volume Layout Details window.

B.3.4.4 Color Preferences

The Color tab sets color preferences. Change colors by clicking on a color in the color wheel or by sliding the Red, Green, Blue, and Brightness sliders.

Colors can be set for:

- **Background Color**
Sets the background color for all the Storage Administrator windows.
- **Foreground Color**
Sets the color for foreground text in the Storage Administrator windows.
- **Tree/Table Color**
Sets the background color for the Object Tree and Object Table.
- **Connecting Line Color**
Sets the color for the lines that connect items in the Object Tree.
- **Selection Color**
Sets the color for selected items.
- **Selection Foreground Color**
Sets the color for foreground text in selected items.
- **Link Color**
Sets the color for links (such as the links to tasks in the Command Launcher).
- **Projection Color**
Sets the color for the lines that show object relationships in the Volume Layout Details window.

B.3.4.5 Geometry Preferences

The Geometry tab sets the width and height (in pixels) for:

- Main window
- Object Search window
- Alert Monitor window
- Task Request Monitor window
- Volume Layout Details window
- Command Launcher window
- Object Table Copy window

If you resize one of these windows with the splitters, the new size is displayed in the Geometry preference for that window the next time you open the Preferences dialog box.

B.3.4.6 Tree/Table Preferences

The Tree/Table tab sets Object Tree and Object Table preferences for:

- **Display Full Path**
Displays path information in the Object Tree and Object Table.
- **Auto Scroll Table**
When an object is added or changed, scrolls through the objects until the new or changed object is visible in the Object Table.
- **Splitter Position**
Moves the splitter to adjust the relative sizes of the Object Tree and Object Table panes.
- **Selector Tree/Table Width**
Sets the width (in pixels) of the Object Tree and Object Table for Browse dialog boxes that contain an Object Tree and an Object Table.
- **Selector Table Width**
Sets the width (in pixels) of the Object Table for Browse dialog boxes that contain only an Object Table.
- **Visible Selector Rows**
Sets the number of rows displayed in the Object Tree and Object Table in Browse dialog boxes.

B.3.4.7 Toolbar Preferences

The Toolbar tab sets preferences for:

- **Show Toolbar**
Shows or hides the toolbar.
- **Position**
Places the docked toolbar at the top, bottom, or side of the Main window.
- **Presentation**
Displays graphics, labels, or both on the buttons in the toolbar.

B.3.4.8 Layout Details Preferences

The Layout Details tab sets Volume Layout Details window preferences for:

- **Compress Display**
Compresses the graphical display of objects so that details are hidden.
- **Projection on Selection**

When an object is selected, highlights objects that are related to or part of that object.

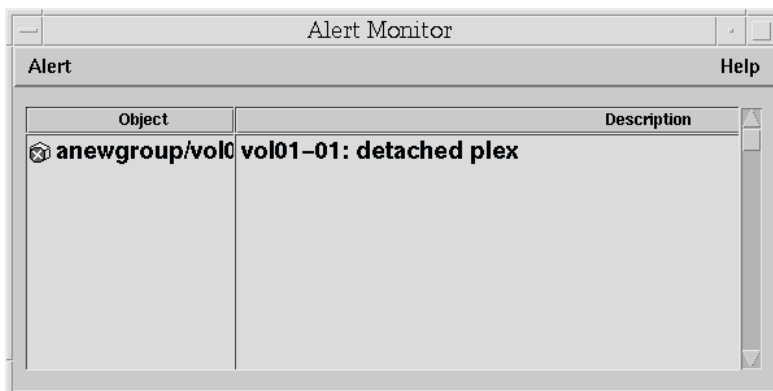
- Subdisk Projection

When a subdisk is selected, highlights other subdisks on the same disk.

B.3.5 Overview of the Alert Monitor Window

The Alert Monitor window (Figure B–5) displays information about failed objects or objects that experienced other errors. Each object is displayed with a description of the failure or error. When an object fails and an alert occurs, the Alert icon appears on the Status bar of the Main window and also overlays the object’s icon in the Object Table.

Figure B–5: Alert Monitor Window



To display the Alert Monitor window either:

- Click on Alert in the toolbar
- Choose Alerts from the Window menu
- Click on the Alert icon on the Status bar

To view the properties of an object with an alert, select the object and choose Object Properties from the Alert menu. You can also access the object Properties dialog box by right-clicking and choosing Properties from the pop-up menu or by double-clicking on the object.

B.3.6 Overview of the Object Table Copy Window

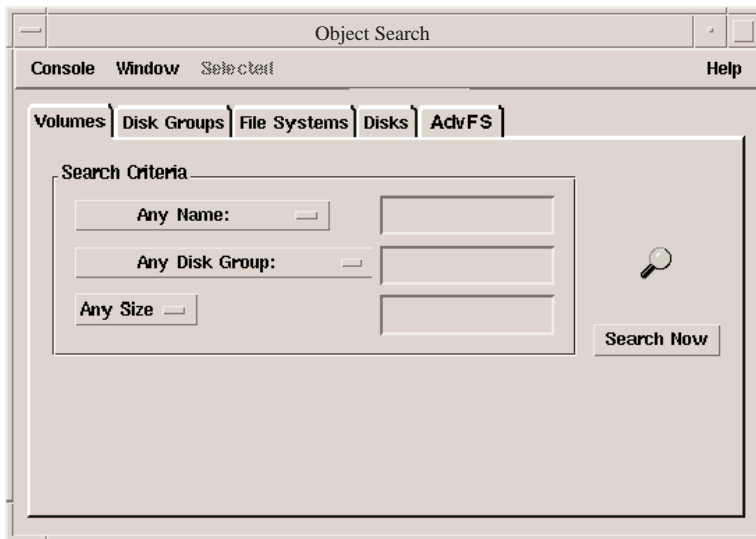
A copy of the Object Table lets you view different parts of your system at the same time. The windows are dynamic, so updates to the system are reflected in all windows. To display the window copy, click on Table in the toolbar or, from the Window menu, choose Copy Object Table.

B.3.7 Overview of the Object Search Window

The Object Search window searches the system for objects that match the specified search criteria. The Object Search window (Figure B–6) contains a set of tabbed pages that display search options for a particular type of object. You can select the type of objects to search for by clicking on the appropriate tab label. The search takes place only on objects of the type you have selected.

To display the Object Search window, click on Search in the toolbar or, from the Window menu, choose Search.

Figure B–6: Object Search Window



Specify the search criteria by selecting a qualifier from the drop-down menu and then entering a value to search for or exclude. For example, in the Volumes tab, you can select the Name Does Not Contain qualifier and enter the string “swap.” This excludes all volume names that contain the word “swap” from the search results. If you enter criteria in more than one field, the search results reflect only the items that match all criteria (Boolean AND).

The table in the bottom half of the Object Search window displays objects and their properties that match the search criteria. (If you do not see the search results, drag the bottom edge of the window to enlarge the display.) Objects displayed in the Object Search window are monitored and removed from the window if they no longer meet the current search criteria.

The Object Search window has menus similar to those in the Main window menu. The Window menu opens other windows or a copy of the current

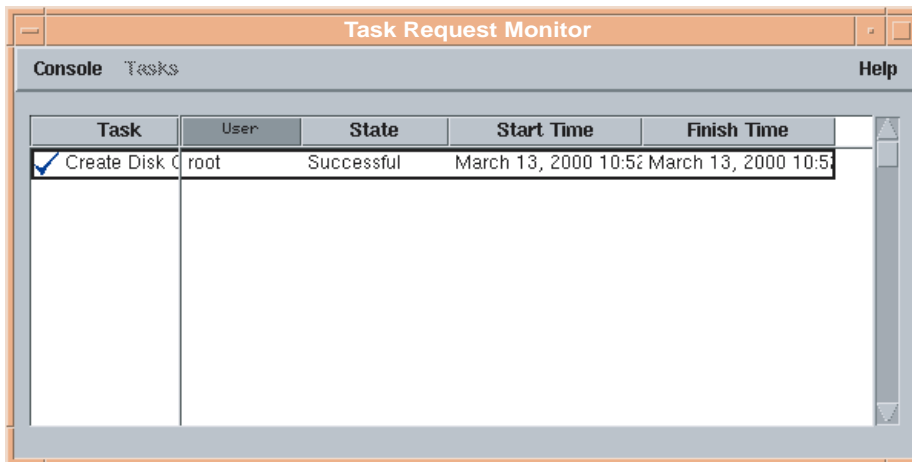
search-results table. The context-sensitive Selected menu accesses tasks or properties for an object selected in the table. You can right-click on an object to access a context-sensitive pop-up menu. To close the Object Search window, choose Close from the Console menu.

B.3.8 Overview of the Task Request Monitor Window

The Task Request Monitor window (Figure B–7) shows LSM and other tasks that the Storage Administrator performed in the current session (and any other sessions running on the system).

To display the Task Request Monitor window, click on Task in the Storage Administrator Main Window, or from the Window menu, choose Tasks.

Figure B–7: Task Request Monitor Window



Task	User	State	Start Time	Finish Time
<input checked="" type="checkbox"/> Create Disk C	root	Successful	March 13, 2000 10:52	March 13, 2000 10:52

Each task is listed with properties such as the user who performed the task, the task status, and the start and finish times.

You can view the low-level commands used to perform a task and see any relevant error messages for failed tasks by displaying the Properties window. To display the Task Properties window, select a task and from the Tasks menu, select Properties. You can copy commands from the Executed Commands field of the Tasks Properties window to the command line or to a script file.

To remove finished tasks and to close the window, from the Console menu choose Remove Finished Tasks.

B.4 Storage Administrator GUI Shortcuts

Shortcuts let you more efficiently perform operations with the GUI.

B.4.1 Sorting Objects

To sort the objects in a table column, click on the column heading. To reverse the order of the objects, click on the column heading again. The sort order cannot be saved with other user preferences.

You can sort entries in the Object Table, the Command Launcher, the Object Search window, and the Task Request Monitor window.

B.4.2 Clearing Alerts

To acknowledge and clear an Alert icon displayed on the Status bar, choose Clear Alert Status from the Options menu.

B.4.3 Using Keyboard Shortcuts

You can use the keyboard shortcuts in Table B–1 and Table B–2 instead of menu commands.

The shortcuts in Table B–1 work from anywhere in the Storage Administrator window.

Table B–1: Keyboard Shortcuts

Keystrokes	Action
Ctrl Shift V	Create a volume
Ctrl G	Create a disk group
Ctrl F	Create a file system
Ctrl Z	Resize the selected object
Ctrl N	Rename the selected object
Ctrl Shift R	Remove the selected object
Ctrl P	Show the properties of the selected object
Ctrl L	Show the layout of the selected volume (graphical view)

The shortcuts in Table B–2 work only in the Main window.

Table B–2: Main Window Keyboard Shortcuts

Keystrokes	Action
Ctrl R	Open the Task Request Monitor window
Ctrl A	Open the Alert Monitor window
Ctrl S	Open the Object Search window
Ctrl C	Close the window

B.4.4 Docking the Toolbar and Command Launcher

To separate the toolbar from the Main window, place the pointer over the toolbar handle (the thin bar next to the toolbar) and drag the toolbar outside the window. You can also use the toolbar handle to move the toolbar to the bottom, side, or top of the Main window.

You can separate or attach the Command Launcher and the Main window. By default, they are attached and the Command Launcher is not shown.

- To separate the Command Launcher from the Main window, choose Preferences from the Options menu. In the Preferences dialog box, choose the Main Window tab. Click on Show Command Launcher and then click on Dock Command Launcher.
- To attach the Command Launcher and the Main window, choose Preferences from the Options menu. In the Preferences dialog box, choose the Main Window tab. Click on Dock Command Launcher.

Glossary

The following are LSM terms and definitions.

C

concatenated plex

A plex that uses subdisks on one or more disks to create a virtual contiguous region of storage space that is accessed linearly. If LSM reaches the end of a subdisk while writing data, it continues to write data to the next subdisk, which can physically exist on the same disk or a different disk. This layout lets you use space on several regions of the same disk, or regions of several disks, to create a single big pool of storage.

See also *RAID 5 plex*, *striped plex*

clone disk group

A disk group created by the `volclonedg` command that is an exact duplicate of another LSM disk group on the system or cluster and whose disks were cloned in hardware before its creation.

configuration database

A small database that contains all volume, plex, subdisk, and disk media records for a disk group. This database is stored in the private regions of some or all disks in the disk group, possibly with two copies on each disk. Because the database pertains to a disk group, record associations cannot span disk groups. Thus, you cannot define a subdisk on a disk in one disk group and associate it with a volume in another disk group.

The configuration database in the `rootdg` disk group contains records for all other disk groups.

See also *description set*, *private region*, *root disk group (rootdg)*

D

description set

A set of files saved by using the `volsave` command that can be used to restore an LSM configuration. By default, an LSM description set is saved in a time-stamped directory under the `/usr/var/lsm/db` directory.

Dirty Region Log (DRL)

The log for a mirrored volume. Typically, the DRL for a mirrored volume is a separate plex containing one subdisk, but the DRL can also be a subdisk

associated with a data plex. A DRL maintains a bitmapped representation of the regions of the volume, and marks as dirty all regions being written to. A DRL reduces the time required to restore synchronization of data for all the plexes in the volume when the system restarts after a failure. A volume can have more than one DRL for redundancy. A DRL does not provide any benefit in the event of a disk failure affecting the volume.

disk

Disks exist as two entities:

- A physical disk on which all data is ultimately stored and which exhibits all the behaviors of the underlying technology.
- An LSM representation of disks that, while mapping one-to-one with the physical disks, is just a presentation of units from which allocations of storage are made.

The difference is that a physical disk presents the image of a device with a definable geometry with a definable number of cylinders, heads, and so on, whereas an LSM disk is simply a unit of allocation with a name and a size.

Disks used by LSM usually contain two special regions: a private region and a public region. Typically, each region is formed from a complete partition of the disk, resulting in a `sliced` disk; however, the private and public regions can be allocated from the same partition, resulting in a `simple` disk. A disk used by LSM can also be a `nopriv` disk, which has only a public region and no private region. LSM `nopriv` disks are created as the result of encapsulating a disk or disk partition.

See also *disk group*, *nopriv disk*, *simple disk*, *sliced disk*, *subdisk*, *volume*

disk access name

See *disk access record*

disk access record

A configuration record identified by a disk access (DA) name that defines the path to a disk. Disk access records most often include a unit number. LSM uses the disk access records stored in a system to find all disks attached to the system. Disk access records do not identify particular physical disks.

Through the use of disk IDs, LSM lets you move disks between controllers or to different locations on a controller. When you move a disk, a different disk record name is used to access the disk, although the disk media record continues to track the actual physical disk.

On some systems, LSM builds a list of disk access records automatically, based on the list of devices attached to the system. On these systems, you do not need to define disk access records explicitly. On other systems, you must define disk access records with the `/sbin/voldisk define` command.

Specialty disks, such as RAM disks or floppy disks, are likely to require explicit `/sbin/voldisk define` commands.

See also *disk ID*, *disk media record*, *volboot file*

disk group

A group of disks that share a common configuration database. Each disk group has an administrator-assigned name that you use to reference that disk group. Each disk group has an internally defined, unique disk group ID, which differentiates two disk groups with the same administrator-assigned name.

Disk groups provide a method to partition the configuration database, so that the database size is not too large and so that database modifications do not affect too many drives. They also allow LSM to operate with groups of physical disk media that can be moved between systems.

Disks and disk groups have a circular relationship: disk groups are formed from disks, and disk group configuration database copies are stored on some or all disks in a disk group. All disks in a disk group are stamped with a disk group ID, which is a unique identifier for naming disk groups.

See also *clone disk group*, *configuration database*, *disk group ID*, *root disk group (rootdg)*

disk group ID

A 64-byte, universally unique identifier assigned to a disk group, when the disk group is created with the `/sbin/voldg init` command. This identifier is in addition to the disk group name, which you assigned. The disk group ID differentiates disk groups that have the same administrator-assigned name.

disk header

A block stored in a private region of a disk that defines several properties of the disk, such as:

- Location and size of the private region
- Location and size of the public region
- Unique disk ID for the disk
- Disk group ID and disk group name (if the disk is currently associated with a disk group)
- Host ID for a host that has exclusive use of the disk

See also *disk ID*, *disk group ID*, *host ID*, *private region*

disk ID

A 64-byte, universally unique identifier assigned to a physical disk, when its private region is initialized with the `/sbin/voldisk init` command.

The disk ID is stored in the disk media record, so that the physical disk can be related to the disk media record at system startup.

See also *disk media record*

disk media name

See *disk media record*

disk media record

A reference to a physical disk or possibly a disk partition, identified by a disk media (DM) name. This record can be thought of as a physical disk identifier for a physical disk or partition. Disk media records are configuration records that you can use to reference a particular disk — independent of its location on the system's various disk controllers.

Operations are provided to set or remove the disk ID stored in a disk media record. Such operations have the effect of removing or replacing disks; any associated subdisks are removed or replaced along with the disk.

See also *configuration database, disk access record, disk ID*

F

Fast Plex Attach

A feature used with mirrored volumes that reduces the time required to resynchronize plexes. Used specifically when detaching one plex of a mirrored volume to create a temporary volume for backups, with the intention of later reattaching the plex to the original (primary) volume.

See also *primary volume, secondary volume, migrant plex*

Fast Plex Attach log

See *FPA log*

FPA log

The log plex for a mirrored volume using the Fast Plex Attach feature, or the log subdisk for a migrant plex detached from a primary volume to create a secondary volume for backups. The FPA log plex for the primary volume tracks only those regions of the volume that change while a migrant plex is detached (as part of a secondary volume). The FPA log subdisk for the secondary volume tracks changes to the secondary volume. When the migrant plex returns to the primary volume, the FPA logs are merged and only the regions marked in the merged log are written to the returning plex, reducing the time required to resynchronize the plex.

H

host ID

A name, usually assigned by the administrator, that identifies a particular system or cluster member. Host IDs are used to assign ownership to particular physical disks. When a disk is part of a disk group that is in active use by a particular host, the disk is stamped with that host ID and only that host can access the disk. Use the `/sbin/voldisk clearimport` command to clear the host ID stored on a disk.

If a disk is a member of a disk group and has a host ID that matches a particular host, then that host will import the disk group as part of system startup.

hot-spare disk, hot-sparing

A hot-spare disk is an LSM disk that you designate to automatically replace a disk that fails while in use by a mirrored or RAID 5 volume. You enable and disable the hot-sparing feature with the `volwatch` command. You can designate a hot-spare disk when you initialize it for LSM use or later, as long as the disk is not being used by other LSM objects (volumes or subdisks).

K

kernel log

A log kept in the private region on the disk that is written by the LSM kernel. The log contains records describing the state of volumes in the disk group. This log provides a mechanism for the kernel to persistently register state changes, so that the `vold` daemon can detect the state changes even in the event of a system failure.

L

log plex

A plex that keeps track of write activity for a mirrored or RAID 5 volume or as part of the Fast Plex Attach feature.

See also *Dirty Region Log (DRL)*, *FPA log*, *RAID 5 log*

M

migrant plex

A plex that, with the support of the Fast Plex Attach feature, is detached from a primary volume and used to create a secondary volume for backups.

See also *Fast Plex Attach*, *primary volume*, *secondary volume*

mirror

See *plex*

mirrored volume

A volume that has two or more concatenated or striped data plexes and, typically, at least one log plex. Contrast with simple volume.

See also *RAID 5 volume*, *redundant volume*

N**nonredundant volume**

A volume that has only one concatenated or striped data plex. If one disk in the volume fails, the entire volume is lost. Contrast with mirrored volume and RAID 5 volume.

See also *simple volume*

nopriv disk

A disk that is configured for use by LSM and has only a public region and no private region. A `nopriv` disk is typically created as a result of encapsulating existing data in a disk or disk partition.

See also *private region*, *public region*, *simple disk*, *sliced disk*

P**plex**

1. A copy of a volume's logical data address space; also known as a mirror. Each plex is, at least conceptually, a copy of the volume that is maintained consistently in the presence of volume I/O and changes to the LSM configuration. Plexes represent the primary means of configuring storage for a volume. Plexes that store data can have a concatenated, striped, or RAID 5 organization (layout).
2. A log that keeps track of write activity for a mirrored or RAID 5 volume or as part of the Fast Plex Attach feature. A log plex consists of only one associated log subdisk and its layout is always concatenated.

See also *concatenated plex*, *log plex*, *migrant plex*, *plex consistency*, *RAID 5 plex*, *striped plex*

plex consistency

A state of equivalence of the contents of all data plexes in an LSM mirrored volume, so that a read request from one plex returns the same data as a read request from the corresponding region in any other plex. If the plexes of a volume contain different data, then the plexes are said to be inconsistent.

Plex inconsistency is a serious compromise of data integrity. This inconsistency is caused by write operations that start around the time of a system failure and are written to one plex but not the other when the system crashes. If the plexes are not first synchronized to contain the same data, plexes are inconsistent after creation of a mirrored volume.

An important role of LSM is to ensure that consistent data is returned to any application that reads a volume. This might require that plex consistency of a volume be “recovered” by copying data between plexes, so that they have the same contents. Alternatively, you can put a volume into a state so that reads from one plex are automatically written back to the other plexes, making the data consistent for that volume offset.

primary volume

The volume from which a plex is detached (the migrant plex) with Fast Plex Attach support to create a temporary secondary volume for backups.

See also *Fast Plex Attach*, *secondary volume*, *migrant plex*

private region

The private region of a disk contains on-disk structures that are used by LSM for internal purposes. Each private region is typically 4096 blocks and begins with a disk header that identifies the disk and its disk group. Private regions can also contain copies of a disk group’s configuration database and copies of the disk group’s kernel log.

See also *disk header*, *kernel log*, *public region*

public region

The public region of a disk is the space reserved for allocating subdisks. Subdisks are defined with offsets that are relative to the beginning of the public region of a disk. Only one contiguous region of a disk can form the public region for a disk.

See also *private region*, *subdisk*

R

RAID 5 log

The log plex for a RAID 5 volume. A RAID 5 log plex stores a copy of the data and parity information for a predefined number of writes. In the event of a system failure, a RAID 5 log reduces the time required to resynchronize data and parity in the volume when the system restarts. When the system restarts after a failure, any write operations that did not complete before the failure are restarted.

See also *RAID 5 plex*, *RAID 5 volume*

RAID 5 plex

A plex that distributes data and parity in equal-size units evenly across each of its associated subdisks. A plex has a characteristic number of stripe columns (represented by the number of associated subdisks) and a characteristic stripe width. The stripe width defines how much data with a particular address is allocated to one of the associated subdisks. The parity data is the result of an XOR operation on the data in each stripe unit. The parity data is written to a different column (presumed to be a different disk) for each stripe, left-shifted by one column, so that no one column contains all the parity for the volume. Therefore, if a disk in a RAID 5 plex fails, the volume is still recoverable by recreating the missing data or parity for each stripe.

See also *concatenated plex*, *RAID 5 log*, *RAID 5 volume*, *striped plex*

RAID 5 volume

A volume that uses a RAID 5 plex and, typically, at least one RAID 5 log plex. A RAID 5 volume has only one RAID 5 plex. Also described as a redundant volume, because a RAID 5 volume provides data redundancy in the form of parity.

See also *mirrored volume*, *RAID 5 log*, *RAID 5 plex*, *redundant volume*, *simple volume*

read policy

A configurable policy for switching between plexes for volume reads (applies only to volumes with striped or concatenated plexes). When a volume has more than one enabled associated plex, LSM distributes reads between the plexes to distribute the I/O load and thus increase total possible bandwidth of reads through the volume.

You set the read policy to be one of the following:

- round-robin read policy

For every other read operation, switches to a different plex from the one used for the previous read operation. Given three plexes, switches between each of the three plexes, sequentially.

- preferred plex read policy

Specifies a particular plex that is used to satisfy read requests. In the event that a read request cannot be satisfied by the preferred plex, this policy changes to the round-robin read policy.

- select read policy

The default policy. Adjusts to use an appropriate read policy based on the set of plexes associated with the volume. If only one enabled read-write striped plex is associated with the volume, then that plex is chosen automatically as the preferred plex; otherwise, the round-robin policy

is used. If a volume has one striped plex and one concatenated plex, preferring the striped plex often yields better throughput.

redundant volume

An LSM volume that provides data redundancy, either through mirroring (as in a mirrored volume, which can use two or more concatenated or striped data plexes, each of which contains a complete set of the volume data), or through striping with parity (as in a RAID 5 volume), and has at least one log plex. Contrast with nonredundant volume.

See also *log plex, mirrored volume, RAID 5 volume*

root disk group (rootdg)

LSM creates and requires one special disk group named `rootdg`. This group is generally the default for most utilities. In addition to defining the regular disk group information, the configuration database for the root disk group contains local information that is specific to a disk group.

See also *configuration database*

S

secondary volume

A temporary volume created for performing backups of another volume, using the Fast Plex Attach feature. The secondary volume is created from a plex (the migrant plex) that was detached from a mirrored volume (the primary volume) with Fast Plex Attach support.

See also *Fast Plex Attach, migrant plex, primary volume, snapshot volume*

simple disk

A disk configured for use by LSM that has a public region and a private region that occupy the same disk partition. A `simple` disk is created by initializing a disk partition (including the `c` partition), instead of the entire disk (without a partition specifier), for LSM use.

See also *nopriv disk, sliced disk, public region, private region*

simple volume

A volume that uses only one concatenated plex. This type of volume provides no data redundancy and no protection from system failure but does permit you to create a volume using space on multiple disks, move data to other LSM disks, and perform online volume management. Without an LSM license, you can create only simple volumes.

See also *concatenated plex, mirrored volume, RAID 5 volume*

sliced disk

A disk configured for use by LSM that has a separate public region (typically the `g` partition of the disk) and a private region (typically the `h` partition).

See also *nopriv disk*, *simple disk*, *public region*, *private region*

snapshot volume

A volume created from a complete, enabled plex of another volume (with the `volassist snapshot` command), used to perform data backups or other operations while leaving the original volume mounted and usable. An LSM snapshot volume is a mounted, fully readable and writable entity that contains a full copy of the data in the original volume, and that is completely separate from the original volume that supplied the data plex for its creation.

See also *secondary volume*

stripe width

A value that represents the amount of data that LSM writes to the subdisks in each column of a striped or RAID 5 data plex. LSM divides the full I/O request into units of the size defined by the stripe width and writes each unit to a different subdisk. The default stripe width for a striped plex is 64K bytes and for a RAID 5 plex is 16K bytes. Stripe widths do not apply to concatenated plexes by definition.

See also *striped plex*, *RAID 5 plex*

striped plex

A plex that places data evenly across each of its associated subdisks. A plex has a characteristic number of stripe columns (represented by the number of associated subdisks) and a characteristic stripe width. The stripe width defines how data with a particular address is allocated to one of the associated subdisks. Given a stripe width of 128 blocks and two stripe columns, the first group of 128 blocks is allocated to the first subdisk, the second group of 128 blocks is allocated to the second subdisk, the third group to the first subdisk, and so on.

See also *concatenated plex*, *RAID 5 plex*, *stripe width*

subdisk

A region of storage allocated on a disk for use by a volume. Subdisks are associated with volumes through plexes. You organize one or more subdisks to form plexes based on a plex layout: concatenated, striped, or RAID 5. Subdisks are defined relative to disk media records.

See also *concatenated plex*, *disk media record*, *RAID 5 plex*, *striped plex*

V

volboot file

The `volboot` file is a special file (usually stored in `/etc/vol/volboot`) that is used to load the `rootdg` disk group at system startup and to define a system's host ID. In addition to a host ID, the `volboot` file contains a list of disk access records. On system startup, this list of disk access records is scanned to find a disk that is a member of the `rootdg` disk group and that is stamped with this system's host ID. When such a disk is found, its configuration database is read and is used to get a more complete list of disk access records that are used in a second stage of loading the root disk group and to locate all other disk groups.

See also *disk access record*

volume

A virtual disk device that applications and file systems can use like a physical disk or disk partition. Volumes present block and raw device interfaces that are compatible in their use with disk partition devices. A volume can span several disk drives, consist of multiple mirrors (plexes), and be moved to different disks. You can change the configuration of a volume without causing disruption to applications or file systems that are using the volume.

See also *plex*

Index

A

AdvFS domain

- backing up multiple volumes, 5–32
- encapsulating, 4–36
- migrating to LSM volume, 4–37
- moving from LSM volume to
 - physical storage, 7–20
- placing under LSM control, 4–36
- removing from LSM control, 5–38
- unencapsulating, 5–38
- unmigrating, 7–20
- using LSM volume for storage, 4–33

AdvFS file system

- growing a volume, 5–40
- shrinking a volume, 5–43

allvol.DF file

- contents, 5–14

automatic data relocation

- (See hot-sparing feature)

B

BCL to DRL conversion, 7–2

block device interface (LSM), 3–29

boot disk

- encapsulating, 3–6
- failed, 6–14
- replacing failed, 6–14
- unencapsulating, 7–16

bootable tape utility, 3–6n

bootdef_dev variable

- resetting, 7–18

C

character device interface (LSM), 3–29

cluster

- adding member with LSM volumes, 7–6
- encapsulating members' swap devices, 3–21
- planning for LSM, 2–10
- shared storage, 2–10
- unencapsulating members' swap devices, 7–20
- using volumes for clusterwide root, 3–13

command line interpreter, 1–20

commands (LSM), 1–21

- troubleshooting, 6–8

concatenated plex, 1–8

- creating, 4–12

concatenated volume

- worksheet, 2–7

configuration database

- backing up, 5–12
- backing up before upgrading LSM, 7–2
- changing attributes, 5–15
- contents of backup, 5–13
- disk requirements for, 1–5
- displaying backups, 5–14
- managing, 5–12
- number of records stored, 1–5
- optimizing after upgrade, 7–5
- overview, 4–2
- restoring from backup, 5–14
- saving, 5–12

connectivity guidelines (LSM),
2–10

D

daemons (LSM), 3–30

data encapsulation, 4–34

device drivers (LSM), 3–30

device interface (LSM), 1–18

device special files (LSM), 3–29

directories (LSM), 3–29

dirty region log

(*See* DRL (dirty region log))

disk, 1–2, 1–3, 1–4, 3–24

(*See also* hot-spare disk; nopriv
disk; simple disk; sliced disk)

adding to disk group, 5–8

checking status, 6–10

displaying information, 5–1

displaying list of, 5–7

encapsulating, 4–35

evacuating, 5–3

failed, 6–12

failing, 6–11

finding unused, 2–15

initializing, 4–1, 4–3

managing, 5–1

moving data off, 5–3

moving volume off, 6–11

placing off line, 5–2

placing on line, 5–2

removing from disk group, 5–8

removing from LSM, 5–6

renaming, 5–1

replacing failed, 6–12

temporary failure, 6–11

troubleshooting, 6–10

disk access name, 1–4

disk configuration GUI, 2–15

disk group, 1–5

(*See also* rootdg disk group)

adding a disk, 4–5, 5–8

cloning, 5–11

cluster considerations, 2–10

converting from Version 4.0, 7–4

creating, 4–4, 4–8

definition, 1–5

deporting, 5–9

deporting before upgrading LSM,
7–3

displaying information, 5–7

forced importing, 6–28

free space, 5–7

importing, 5–10

managing, 5–7

moving between systems, 7–10

planning, 2–10

removing a disk, 5–8

renaming, 5–10

setting number of configuration
copies, 4–8

troubleshooting, 6–27

worksheet, 2–14

worksheet examples, 2–13

disk label

backing up, 4–4

reusing, 6–13

disk media name, 1–5

associating to access name, 7–12

disk partition

encapsulating, 4–35

initializing, 4–3

displaying information

configuration database backups,
5–14

disk, 5–1

disk group, 5–7

plex, 5–45

statistics, 6–2

subdisk, 5–54

system volumes, 3–11

volume, 5–18

volume status, 6–22

domain

(*See* AdvFS domain)

DRL (dirty region log), 1–12

default size, 2–1
minimum size, 2–4
optimizing volume performance,
2–5
planning, 2–4
subdisk sizes, 2–4
using nondefault size, 2–4

dxlsm interface

(See Visual Administrator
interface)

E

/etc/default/volassist file

(See volassist file)

/etc/fdmns directory

(See fdmns directory)

/etc/fstab file

(See fstab file)

/etc/sysconfigtab file

(See sysconfigtab file)

/etc/vol directory

(See vol directory)

event (LSM)

monitoring, 6–1

F

Fast Plex Attach feature

(See FPA (Fast Plex Attach))

fdmns directory, 3–8, 7–17

file system, 4–33

(See also AdvFS file system;
UFS file system)

configuring LSM volume for, 4–33

encapsulating, 4–35

growing a volume, 5–40

shrinking a volume, 5–43

FPA (Fast Plex Attach)

and rootvol, 6–17

applicability, 5–21

disabled by growing a volume,

5–41, 5–42

disabled by shrinking a volume,

5–44n

disabling, 5–40, 5–49, 5–50

enabling, 5–26

overview, 5–21

restrictions, 5–21

FPA log plex, 5–22

adding, 5–48

removing, 5–40

FPA subdisk, 5–22

attaching, 5–26

fstab file, 3–8, 7–17

H

host ID

changing, 7–11

clearing, 6–27

in header file, 5–13

in volboot file, 3–3

mismatch, 6–27

hot-spare disk, 3–23

configuring, 3–24

deconfiguring, 3–24

hot-sparing feature

enabling, 3–23

mail notification, 3–23

mail notification examples, 3–25

moving objects relocated by, 5–5

overview, 3–23

I

I/O statistics, 6–2

interfaces (LSM), 1–19, 1–20,

1–23, A–1, B–1

(See also Storage Administrator;

Visual Administrator

interface; command line

interpreter; voldiskadm menu interface)

K

kernel states, 6–4t

L

license (LSM), 3–2

log plex, 1–12, 1–13

(*See also* DRL (dirty region log);

FPA log plex; RAID 5 log)

adding, 5–47

default size, 2–1, 2–4

definition, 1–12

minimum size, 2–4

LSM

initialization restrictions, 2–16

initializing, 3–1

installing, 3–1

overview, 1–1

upgrading, 3–1

upgrading from Version 4.0, 7–1

verifying initialization, 3–4

lsmsa interface

(*See* Storage Administrator)

M

migrant plex, 5–21

disabling FPA log, and, 5–40

growing a volume, and, 5–41, 5–42

resynchronization, 5–25

shrinking a volume, and, 5–44n

specifying, 5–27

mirror

(*See* plex)

mirrored volume, 1–14

recovering, 6–18

recovering no valid plex, 6–19

recovering one valid plex, 6–20

N

nonredundant volume, 1–14

(*See also* redundant volume)

recovering, 6–18

nopriv disk, 1–4

configuration database and, 1–5

moving between systems, 7–13

O

object hierarchy (LSM), 1–1

object states (LSM), 6–3

objects (LSM), 1–1

P

phantom subdisk, 7–18n

plex, 1–8, 1–9, 1–11, 1–12, 1–13,

5–21, 5–47, 5–48

(*See also* DRL (dirty region log);

FPA (Fast Plex Attach); FPA

plex; RAID 5 log; RAID 5 plex;

concatenated plex; log plex;

migrant plex; striped plex)

adding, 5–46

changing layout, 5–51

definition, 1–7

detaching, 5–49

displaying information, 5–45

dissociating, 5–50

managing, 5–45

reattaching, 5–51

recovering no valid, 6–19

recovering one valid, 6–20

plex resynchronization

disabling for swap volumes, 4–23

migrant plex, and, 5–25

root volume, 3–11n

with DRL, 1–12

with FPA, 1–13

without DRL, 5–49, 5–50

plex states, 6–6t

postinstallation tasks, 3–5
primary swap space
 encapsulating, 3–6
primary volume, 5–21, 5–22
 (*See also* FPA (Fast Plex Attach))
private region, 1–3
 number of records stored, 1–5
 specifying offset, 4–2
public region, 1–3
 specifying offset, 4–2

R

RAID 5 log, 1–13
 planning, 2–5
 recovering, 6–21
 size, 2–5
RAID 5 plex, 1–11
 creating, 4–19
 default stripe width, 1–11
 using different buses, 4–31
RAID 5 volume, 1–17
 recovering, 6–18
 recovering from multiple disk failures, 6–20
 worksheet, 2–9
redundant volume, 1–14, 1–17
 (*See also* nonredundant volume)
root file system
 using LSM for, 3–6
rootdg disk group, 1–5
 moving between systems, 7–11
rootvol volume
 and FPA, 6–17
 backup volume, 5–27n

S

secondary swap space, 4–21

secondary volume, 5–22
 (*See also* FPA (Fast Plex Attach))
simple disk, 1–4, 4–3
simple volume, 1–14
sliced disk, 1–3, 4–3
software subsets (LSM), 3–1
stale subdisk
 recovering, 6–10
startup problems, 6–8
states (LSM), 6–4
Storage Administrator, 1–23, A–1
 access log file, B–2
 adding disk, A–6
 adding disk partition to a disk group, A–17
 adding disk to a disk group, A–16
 adding file system, A–32
 adding fileset to domain, A–49
 adding hot-spare disk, A–8
 adding log, A–30
 adding mirror, A–29
 adding volume to domain, A–48
 AdvFS, A–46
 Alert Monitor window, B–13
 analyzing activity, A–35
 changing fileset quotas, A–52
 checking file system, A–34
 clearing alerts, B–16
 cloning fileset, A–51
 closing, A–6
 command log file, B–1
 copying volume data, A–39
 creating disk group, A–15
 creating domain, A–46
 creating volume, A–27
 customizing, B–7
 defragmenting domain, A–50
 deporting disk group, A–17
 destroying disk group, A–21
 disabling mirror, A–37

- docking the Command Launcher, B-17
- docking the toolbar, B-17
- evacuating disk, A-9
- exiting, A-6
- importing disk group, A-18
- installing, A-2
- joining subdisks, A-22
- keyboard shortcuts, B-16
- main window, A-3, B-5
- managing disk groups, A-15
- managing disks, A-6
- managing subdisks, A-22
- managing volumes, A-26
- mapping volumes, disks, and subdisks, A-35
- mirroring disk, A-10
- mounting file system, A-31
- mounting fileset, A-54
- moving disk group, A-21
- moving subdisk, A-24
- Object Properties dialog box, B-6
- Object Search window, B-14
- Object Table Copy window, B-13
- overview, A-1
- placing disk off line, A-10
- placing disk on line, A-11
- pop-up menus, A-6
- recovering volumes, A-40
- recovering volumes in a disk group, A-20
- recovering volumes on a disk, A-11
- removing disk from disk group, A-12
- removing fileset, A-55
- removing log, A-41
- removing mirror, A-41
- removing subdisk, A-26
- removing volume, A-42
- renaming disk, A-13
- renaming disk group, A-20
- renaming fileset, A-54
- renaming volume, A-43
- repairing mirror, A-38
- replacing disk, A-14
- resizing volume, A-43
- restoring volume from backup, A-41
- right mouse button, A-6
- scanning for new disks, A-15
- selecting objects, A-5
- server log file, B-2
- shortcuts, B-15
- sorting objects, B-16
- specifying object sizes, B-4
- specifying objects, B-4
- splitting subdisk, A-23
- starting, A-2
- starting volume, A-45
- stopping volume, A-45
- Task Request Monitor window, B-15
- tracking activity, B-1
- unmounting file system, A-35
- unmounting fileset, A-53
- using Command Launcher, A-6
- using Console Menu, A-5
- using dialog boxes, B-3
- using Selected Menu, A-5
- using User Preferences dialog box, B-7
- viewing objects, B-5
- Volume Layout Details window, B-5
- stripe width**, 2-5
 - nondefault, 4-16, 4-19
 - RAID 5 plex, 1-11
 - specifying, 4-16, 4-19
 - striped plex, 1-9
- striped plex**, 1-9
 - creating, 4-14, 4-16
 - default stripe width, 1-9
 - using different buses, 4-27
 - using different-size subdisks, 4-24
- striped volume**
 - worksheet, 2-8

subdisk, 5–22, 6–10, 7–18n
 (*See also* FPA subdisk; phantom subdisk; stale subdisk)
 creating, 4–24
 definition, 1–6
 displaying information, 5–54
 dissociating, 5–56
 joining, 5–54
 managing, 5–54
 moving to different disk, 5–55
 RAID 5 log size, 2–5
 removing, 5–56
 sizes for logging, 2–4
 splitting, 5–55
 stale, 6–10

subdisk states, 6–7

swap device
 encapsulating
 cluster member, 3–13, 3–21
 unencapsulating
 cluster member, 7–20

sysconfigtab file, 3–8, 7–17
 troubleshooting, 6–7

system partitions
 encapsulating, 3–6
 unencapsulating, 7–16

system volumes
 adding a mirror, 3–10
 converting to partitions, 7–16
 displaying, 3–11

T

troubleshooting, 6–1

U

UFS file system, 6–22
 (*See also* troubleshooting)
 encapsulating, 4–35
 growing a volume, 5–40

shrinking a volume, 5–43
 unencapsulating, 5–37
 using LSM volume for storage,
 4–34

V

Version 4.0

upgrading, 7–1

Visual Administrator interface,
 1–20

vol directory, 3–29

volassist addfpa command, 5–48

volassist addlog command, 5–47

volassist command, 4–12, 4–14,
 4–16, 4–19

volassist file, 4–11

volassist maxsize command, 5–7,
 6–25

volassist mirror command, 5–46

volassist move command, 5–3,
 5–5, 6–11

volassist snapback command,
 5–26

overview, 5–21

volassist snapfast command, 5–26

overview, 5–21

volassist snapshot command, 5–29

volassist snapstart command,
 5–29

volboot file, 3–5

volclonedg command, 5–11

vold command, 6–8, 7–23

vold daemon, 3–30

checking, 6–8

monitoring, 6–8

restarting, 6–8

voldctl command, 3–5, 4–1, 6–8,
 6–11, 7–23

voldg adddisk command, 5–8

voldg deport command, 5–9

voldg free command, 5-7, 6-25
voldg import command, 5-10
voldg init command, 4-8
voldg list command, 5-15
voldg rmdisk command, 5-2, 5-8
voldisk check command, 6-10
voldisk command, 2-17, 5-1, 5-6
voldisk init command, 6-13
voldisk list command, 5-7, 6-10
voldisk modddb command, 7-5
voldisk offline command, 5-2
voldisk online command, 5-2
voldisk rm command, 5-6
voldiskadd script, 4-5
voldiskadm menu interface, 1-19
voldisksetup command, 4-1, 6-14
voleedit command, 3-24, 5-1, 5-15, 5-37, 5-40, 5-45, 5-56
volencap command, 3-7, 4-35, 4-36
volevac command, 5-3, 5-5
volinfo command, 6-18, 6-22
voliod daemon, 3-30
 checking, 6-9
 monitoring, 6-8
 setting number of, 6-9
voliod utility, 3-5, 3-30
vollogenvt utility, 7-4
volmake command, 4-27, 4-31
volmend command, 6-18, 6-19, 6-20
volmigrate command, 3-14, 4-37
 cluster root and, 3-14
volplex command, 5-49, 5-50, 5-51, 6-21
 attach a plex, 5-31
 dissociate a plex, 5-30
volprint command, 3-11, 5-18, 5-45, 5-47, 5-54, 6-3, 6-10, 6-23, 6-27
volreconfig command, 3-7, 4-35, 4-36
volrecover command, 5-51, 6-18
volrestore command, 5-14
volrootmir command, 3-10, 6-16
volsave command, 5-12, 7-2
volsd command, 5-54, 5-55, 5-56
volsetup command, 3-3
volstat command, 6-2, 6-3
volume, 1-14, 1-17, 2-1, 3-6, 4-9
 (*See also* RAID 5 volume;
 mirrored volume;
 nonredundant volume;
 redundant volume; root
 file system volume; simple
 volume; system volume;
 volume attributes)
 adding cluster member with, 7-6
 backing up, 5-19
 block device interface, 1-18
 changing permissions, 5-45
 character device interface, 1-18
 checking resynchronization, 6-23
 clearing locks, 6-27
 cluster member swap space, 4-21
 creating, 4-9
 creating nondefault, 4-24
 definition, 1-13
 disabled, 6-22
 displaying information, 5-18
 displaying statistics, 6-2
 displaying status, 6-22
 encapsulating data to, 4-34
 growing, 5-40
 insufficient space to create, 6-25
 managing, 5-18
 maximum size, 5-7
 mirroring, 5-46
 moving off a disk, 5-3, 6-11
 planning, 2-1
 primary swap space, 3-6
 reattaching a plex, 5-51
 recovering, 6-17
 relationship to disk groups, 1-13
 removing (destroying), 5-37
 renaming, 5-40

- restoring from backup, 5–35
- shrinking, 5–43
- specifying size, 4–11
- specifying the stripe width, 4–19
- starting, 5–36
- starting disabled, 6–22
- states, 6–4t
- stopping, 5–36
- troubleshooting, 6–17
- usage types, 1–18

- volume attributes**, 2–1, 4–10
 - defaults file, 4–11
 - displaying defaults, 4–9
 - order of precedence, 4–9
- volume command**, 5–36
- volume kernel states**, 6–4t
- volume start command**, 4–27
- volunroot command**, 7–16, 7–22
- volwatch command**, 3–23