Tru64 UNIX

Logical Storage Manager

Part Number: AA-RH9BA-TE

July 1999

Product Version: Tru64 UNIX Version 5.0 or higher

This manual describes how to configure and manage disk storage using the Logical Storage Manager (LSM) software. It includes information on how to plan, set up, monitor, and change an LSM configuration. The manual also provides instructions for handling problems and reference information on LSM commands and user interfaces.

Compaq Computer Corporation
Houston, Texas

Contents

3 Encapsulating Existing Data to LSM Volumes

4 Encapsulating and Mirroring the Boot Disk

5 Configuring Disks and Volumes

6 Managing LSM Objects

7 LSM Preventative Maintenance Procedures

9 Using the Storage Administrator

C Using the Visual Administrator

D The voldiskadm Menu Interface

E LSM Error Messages

Glossary

Index

Examples

Figures

Tables

About This Guide

The Logical Storage Manager (LSM) software provides high data availability, better performance, and greater storage management flexibility through online support for disk storage devices on Alpha systems. This guide explains LSM concepts and how to install and administer the LSM software.

Audience

This guide is intended for system administrators who need to configure and maintain disk storage under the control of the LSM software. This guide assumes that you have a:

- Working knowledge of the operating system
- Basic understanding of system administration
- Basic understanding of disk structures

Related Documents

The following documents provide information related to LSM:

- The *Installation Guide* and *Installation Guide — Advanced Topics* describe how to install the LSM software.
- The *Release Notes* describe supported LSM features and products, and known problems and solutions.
- The *System Configuration and Tuning* guide includes information on planning, configuring and tuning disk storage subsystems.
- The *AdvFS Administration* guide includes information on using the AdvFS and LSM software.
- The *System Administration* guide describes general disk administration.
- The *Cluster Administration* guide describes how to configure LSM in a TruCluster environment.

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

G	Books for general users
S	Books for system and network administrators
P	Books for programmers
D	Books for device driver writers
R	Books for reference page users

Some books in the documentation help meet the needs of several audiences. For example, the information in some system books 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 books in the Tru64 UNIX documentation set.

Conventions

The following conventions are used in this manual:

#	A number sign represents the superuser prompt.
`%` **`cat`**	Boldface type in interactive examples indicates typed user input.
`file`	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.

1

Introduction to the Logical Storage Manager Software

The Logical Storage Manager (LSM) software is an integrated, host-based disk storage management software tool that enables you to configure disks to protect against data loss and improve disk use and performance. You can use the LSM software to perform disk management functions without disrupting users or applications accessing data on those disks.

This chapter introduces LSM features, capabilities, concepts, architecture, and terminology. The `volintro`(8) reference page also provides LSM terms and commands.

1.1 Overview

In general, disk storage management often requires that for each file system or database that you:

- Allocate and reallocate disk space as space requirements change
- Address the space allocated for a particular file system or database
- Access data through an application programming interface

These requirements are more easily accomplished by using the LSM software. Table 1–1 compares these requirements for systems running with and without the LSM software.

Table 1–1: Disk Storage Management With and Without the LSM Software

Requirement	Without the LSM Software	With the LSM Software
Space Allocation	UNIX disks are divided into *partitions*. A partition is defined by its start address on the physical disk and its length. You must partition the disks according to the needs of the users on the system. You cannot move or extend in size a partition once it is in use.	The LSM software obtains space for a file system or raw database by creating an LSM *volume* of the appropriate size. An LSM volume is built from one or more areas of disk space (also called subdisks) located on one or more physical disks. This makes it possible to create LSM volumes by using disk space that is not contiguous with the space already in use, and to create LSM volumes that exceed the size of a physical disk.
Addressing	A UNIX partition is addressed through a physical address, generally referred to as the *device name* or *devname*.	LSM volumes are addressed using a *volume name*. You use a symbolic *disk media name* to refer to a disk that is managed by the LSM software(for example: `disk01`). This makes it easy to change an LSM volume and space allocation when disks are moved, added, or removed in the configuration without affecting applications.
Data Access	Data storage and retrieval on a UNIX partition is achieved through the standard block- and character-device interfaces using the physical device address. Because the partitioning of disks is not easily changed, it is difficult to ensure that data is placed on the available disk drives for optimal access and performance.	LSM volumes are accessed through the standard block- and character-device interfaces using volume names independent of the physical storage addresses used by the volume. Because you can change LSM volume configurations without interrupting user access to the data, you can dynamically change data placement for optimal access and performance.

1.2 LSM Features

Table 1–2 summarizes features of the LSM software.

Table 1–2: LSM Features

Feature	Description
Online storage management	Provides the ability to manage a system's disks as a pool of storage space for creating LSM volumes. By using LSM volumes instead of disk partitions, you can reconfigure LSM volumes to achieve the best performance and availability as your storage needs change without having to stop storage input and output (I/O), shut down the system, or back up and restore data.
Concatenation (disk spanning)	Combines multiple physical disks or portions of disks into a single, larger LSM volume for use by large file systems or databases.
Striping (RAID0)	Improves a system's disk I/O performance by interleaving the data within a volume across several physical disks. Also enables combining multiple physical disks into an LSM volume, similiar to concatentation, with better I/O performance.
Mirroring (RAID1)	Protects against data loss due to hardware malfunction by creating one or more *mirror* (duplicate) images of data on other disks.
Boot disk mirroring	Enables mirroring of critical system disk partitions used for booting and running the system to ensure that no single disk failure leaves the system unusable.
Dirty Region Logging (DRL)	Provides fast resynchronization of a mirrored volume after a system failure, by resynchronizing only the regions that were being updated when the system failed. DRL replaces the Block Change Logging (BCL) in previous LSM versions.
Striping and mirroring (RAID0+1)	Provides improved system performance and high data availability.
RAID5	Provides higher data availability by storing parity information along with striped data, which improves read performance.
Hot-sparing	Automatically reacts to I/O failures on redundant (mirrored or RAID5) objects by relocating affected objects to spare disks or to other free disk space.

Table 1–2: LSM Features (cont.)

Feature	Description
Encapsulation	Enables migration of existing data on disks and disk partitions to LSM volumes.
TruCluster support	Manages storage in a TruCluster environment the same way system storage space is managed. All LSM features are available within a TruCluster environment except for RAID5 and boot disk mirroring.

1.3 LSM Objects

LSM uses hierarchical objects to organize disk usage. Table 1–3 identifies LSM objects and their purposes:

Table 1–3: LSM Objects

Object	Purpose
Volume	A virtual disk device that appears as a disk partition to an application or file system
Plex	An instance of a volume's data
Subdisk	A logical representation of a set of contiguous disk blocks on a disk
Disk group	A collection of disks for use by LSM that share a common configuration
LSM disk	A contiguous area of disk space from which LSM allocates storage

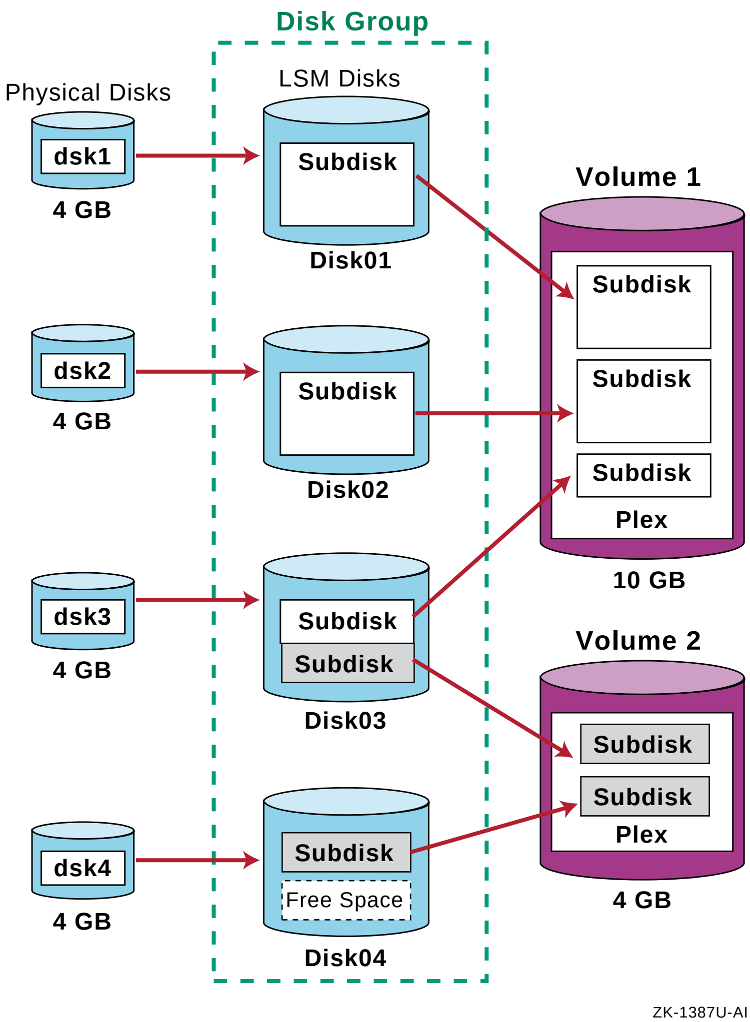
Each object has a dependent relationship on the next-lower object. LSM maintains a configuration database that describes the objects and their relationships.

Figure 1–1 shows a simple LSM configuration consisting of two volumes built from four physical disks. The objects in the configuration are related as follows:

- Physical disks are initialized for use by the LSM software and are assigned to a **disk group** where they become **LSM disks**.
- At least one **subdisk** is mapped to each LSM disk. Each subdisk represents a set of contiguous disk blocks on a physical disk.
- Subdisks are combined to form **plexes**. A plex is one copy or instance of the data.

• **Volumes** are created from plexes, using either a single plex or multiple plexes. Volumes that contain multiple plexes are **mirrored volumes**. Each volume in Figure 1–1 has a single plex, so they are not mirrored.

Figure 1–1: LSM Objects and Their Relationships



1.3.1 Disks

The LSM software can use the following types of storage devices as supported by the operating system:

• Standard Small Computer Systems Interface (SCSI)

• Digital Storage Architecture (DSA)

- Redundant arrays of independent disks (RAID) hardware devices are supported as standard disks, with each RAID device-logical unit viewed as one disk.

Generally, the entire disk is configured for use with the LSM software rather than disk partitions, however, you can use disk partitions. The LSM software logically binds together the partitions from one or more physical disks into a volume that represents the storage to applications and users as a single virtual device. Usually, disk partitions are accessed using a device name ending in a letter from `a-h` (for example, `/dev/disk/dsk7b`).

Figure 1–1 shows that four disks are used with disk access names of `dsk1`, `dsk2`, `dsk3`, and `dsk4`.

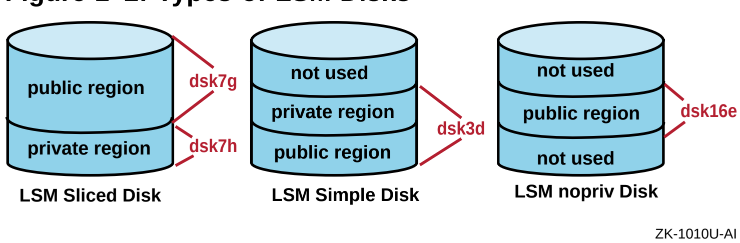
1.3.2 LSM Disks

An LSM disk is created when you initialize a physical disk or disk partitions for use with the LSM software. Two regions are created on each LSM disk:

- A usually small **private region** where the LSM software keeps its internal metadata such as disk header, configuration database, and so on.
- A large **public region** used for storage

You must configure an LSM disk using one of three types of disk layout, as shown in Figure 1–2.

Figure 1–2: Types of LSM Disks



- A **sliced disk** layout is used to configure an entire disk for use with the LSM software. This layout organizes the storage into two regions on separate partitions — a large public region (`dsk7g`) and a small private region (`dsk7h`). This is the most common and recommended layout.
- A **simple disk** layout is used to configure a single disk partition for use with the LSM software. This layout organizes the disk into public and private regions, but both regions are contained within a single disk partition (`dsk3d`).

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

- A **nopriv disk** layout is used to configure disk partitions for use with the LSM software that contain data that you want to preserve, which is accomplished by a process called **encapsulation**. This type of layout has only a public region (`dsk16e`) and no private region.

1.3.3 Disk Groups

A disk group is a named collection of LSM disks that share a common configuration database. The common configuration database contains records describing all the LSM objects in the disk group. The LSM software automatically maintains copies of the disk group's configuration database in the private region of one or more disks in the disk group in case of a disk failure.

When assigned to a disk group, the LSM disk acquires a **disk media name**, which provides a way of specifying the disk independently of its location. This name is either a name that you supply (for example, `payroll2`), or a default name assigned automatically by the LSM software. LSM volumes are created within a disk group and are restricted to the disks within that group.

By default, LSM creates the `rootdg` disk group and, unless otherwise specified, LSM operations are directed to this disk group. You can create disk groups to simplify management and provide data availability. For example:

- On a system with many disks, you can divide disk usage into a few disk groups based on function. This reduces the size of the LSM configuration database for each disk group, and reduces the amount of overhead incurred in configuration changes.
- If a system is unavailable for a prolonged amount of time, you can move the disks in a disk group to another system.

1.3.3.1 Auto-Configured Disks

Automatic configuration provides a convenient way for you to view all the disks on the system to see which disks are currently configured for use with the LSM software and which are not.

When the LSM software starts, it obtains a list of known disk device addresses from the operating system and checks the disk labels to locate all its disk groups, configuration databases, and disks that were configured for use with the LSM software. Disks not initialized for use with the LSM software are automatically configured into the `rootdg` disk group. However the disk is not affected in any way. Automatic configured disks are displayed by utilities as a sliced disk with the status set to `unknown` because they are not currently configured for use with the LSM software.

1.3.4 Subdisks

A subdisk is a set of contiguous blocks on an LSM disk that the LSM software uses to allocate disk space for use in volumes. Free space within an LSM disk's public region can be used to create a subdisk. Subdisks on the same disk cannot overlap each other.

An LSM disk may contain one or more subdisks, as shown in Figure 1–1 where `disk01` and `disk02` each contain a single large subdisk, `disk03` contains two subdisks, and `disk04` contains a single smaller subdisk.

1.3.5 Plexes

A plex is one or more subdisks that are organized into one of the following plex layout types:

- A **concatenated** plex consists of one or more subdisks that create a contiguous address space. Data is allocated sequentially to the subdisks in the plex. This layout is often used for creating large volumes that span multiple disks.
- A **striped** plex contains at least two subdisks, each on a different disk. Data is allocated alternately and evenly to the subdisks in the plex. This layout is useful for balancing the I/O load from applications across multiple disks.
- A **RAID5** plex consists of multiple subdisks on a multiple disks. Data and parity information are allocated alternately and evenly across the subdisks. This plex layout improves I/O performance for read operations, and provides data redundancy by using the parity information to reconstruct data after a disk failure.
- A **log** plex is used with either a mirrored (RAID1) volume's data plexes or a RAID5 plex. A log plex consists of one subdisk that logs a mirrored volume's writes into a DRL or logs a RAID5 volume's writes into a RAID5 log. The recommended configuration is at least one log plex for each LSM mirrored (RAID1) or RAID5 volume.

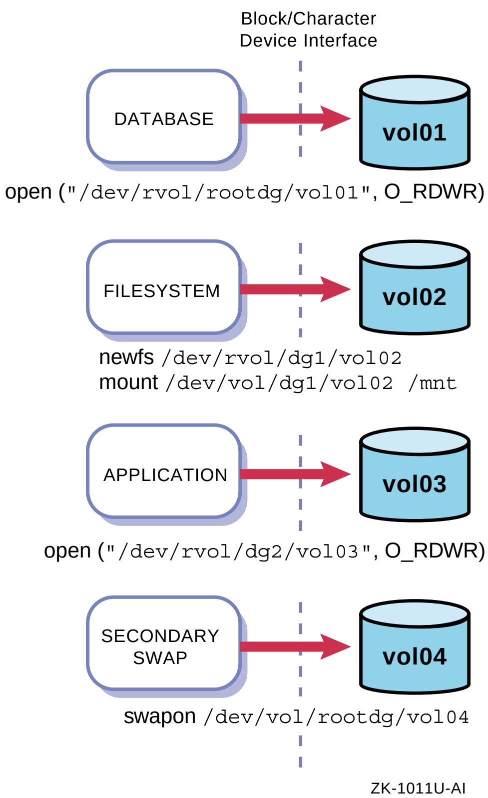
The layout and number of plexes determines how the volume's data is accessed from its underlying storage. For example, a volume with two striped plexes means that the data on a volumes is duplicated or mirrored, and both copies are striped across multiple disks.

1.3.6 Volumes

An LSM volume is a virtual disk. As with all UNIX disks, an LSM volume has a block device interface and a character device interface. A volume's block device is located in the `/dev/vol/`*`diskgroup_name`*`/`*`volume_name`* directory. A volume's character device is in the `/dev/rvol/`*`diskgroup_name`*`/`*`volume_name`* directory. Because these interfaces support the standard UNIX `open`, `close`, `read`, `write`, and `ioctl` calls, an LSM volume can be used by file systems, databases, and applications in the same manner as disk partitions.

For example, Figure 1–3 shows how software that previously used a disk partition can use an LSM volume.

Figure 1–3: Using LSM Volumes Like Other Disk Devices



The storage space on the disk is organized into four LSM volumes. The `vol01` volume is set up for database operations, the `vol02` volume contains a file system that was created and mounted using the `newfs` and `mount` commands, `vol03` is used by another application, and `vol04` is used for secondary swap.

An LSM volume can contain from one to 32 plexes, each containing one or more subdisks. Except for a log plex, each plex contains a copy of the volume's logical data address space.

You configure LSM volumes to have any of the following LSM volume layouts:

- Simple (concatenated)
- Striped (RAID0)
- Mirrored (RAID1)
- Striped and Mirrored (RAID0 plus RAID1)
- RAID5

1.3.6.1 Simple Volumes

A simple volume is useful when there is insufficient contiguous space for a plex on any one disk.

Using an LSM simple volume has minimal I/O performance impact and allows greater flexibility compared to using a disk partition without LSM because you can make on line configuration changes, such as moving the data to a less busy disk, adding a mirror, and so on, without affecting users and applications.

A simple volume contains a single plex in which the data is mapped in a linear manner. You can build the plex from a single subdisk or multiple subdisks (concatenated volume), and the subdisks can reside on the same or different disks (spanned volume). Also, the subdisks do not have to be physically contiguous.

In a simple volume, data is accessed in the first subdisk from beginning to end, then data is accessed in the second subdisk from beginning to end, and so on until the end of the last subdisk.

Caution

Concatenating a plex across multiple disks increases the chance that a disk failure will result in the failure of its volume. Using mirroring or RAID5 (both described later) substantially reduces the chance that a single disk failure will result in a volume failure.

1.3.6.2 Striped Volumes (RAID0)

A striped volume is useful when writing large amounts of data, quickly reading data, or balancing I/O from multi-user applications across multiple disks.

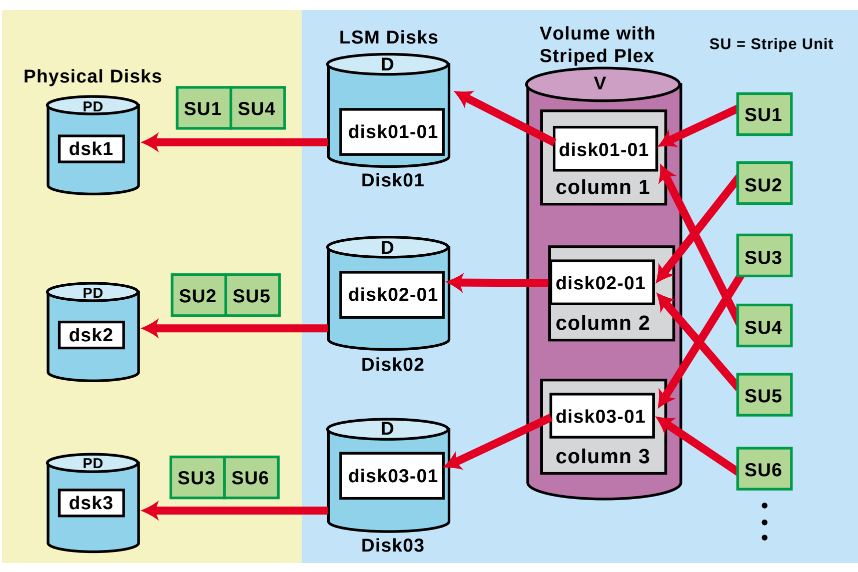
Striping spreads data over two or more disks. A striped plex contains two or more subdisks, spread out over two or more disks. Data is interleaved evenly across the subdisks in a striped plex.

The subdisks are grouped into columns, with each disk limited to one column. Each column contains one or more subdisks. The number and sizes of subdisks per column can vary. You can add subdisks to columns as necessary.

Data is allocated in equal-sized units (called stripe units) that are interleaved between the columns. Each stripe unit is a set of contiguous blocks on a disk. The default stripe unit size is 64KB.

For example, if there are three columns in a striped plex and six stripe units, data is striped over three physical disks, as shown in Figure 1–4.

Figure 1–4: Example of a Striped Volume



In Figure 1–4, the first and fourth stripe (SU1 and SU4) units are allocated in column 1; the second and fifth stripe units (SU2 and SU5) are allocated in column 2; and the third and sixth stripe units (SU3 and SU6) are allocated in column 3.

In Figure 1–4, there are two stripes. The first stripe is SU1 in column 1, SU2 in column 2, and SU3 3 in column 3. The second stripe is SU4 in column 1, SU5 in column 2, and SU6 in column 3.

Striping continues for the length of the columns (if all columns are the same length) or until the end of the shortest column is reached. Any space remaining at the end of subdisks in longer columns becomes unused space.

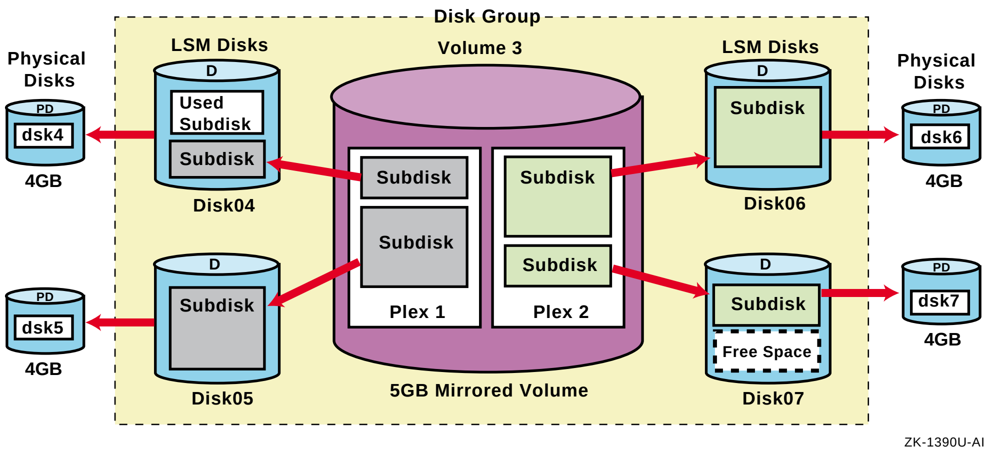
Caution

Striping a volume, or splitting a volume across multiple disks, increases the chance that a disk failure will result in failure of that volume. For example, if five volumes are striped across the same five disks, then failure of any one of the five disks requires that all five volumes be restored from backup. If each volume were on a separate disk, you only need to restore one volume. Using mirroring or RAID5 (both described later) substantially reduces the chance that a single disk failure causes any volumes to fail.

1.3.6.3 Mirrored Volumes (RAID1)

A mirrored volume is useful to reduce the chance that a single disk failure results in volume failure. A mirrored volume uses multiple mirrors (plexes) to duplicate the information in a volume. If a disk fails, the mirrored volume on that disk becomes unavailable, but the system continues to operate using the unaffected mirrored volume. At least two plexes are required for mirroring, as shown in Figure 1–5. Each plex must contain disk space from different disks for the redundancy to be effective.

Figure 1–5: Example of a Mirrored Volume



When striping or spanning across a large number of disks, failure of any one of those disks will generally make the entire plex unusable. The chance of one out of several disks failing makes it worthwhile to consider mirroring to improve the reliability and availability of a striped or spanned volume.

1.3.6.3.1 Dirty-Region Logging

Dirty-region logging (DRL) is an option that provides a fast recovery of mirrored volumes after a system failure. DRL logically divides a mirrored volume into a set of consecutive regions and marks regions that change due to I/O writes to the volume as dirty. When the system restarts after a failure, only those regions of the volume that are marked as dirty in the DRL are recovered.

A write operation to a volume marks a region dirty in the log before the data is written. The dirty bit for a region is not cleared immediately after writing the data to the region. Instead, it remains marked as dirty until the corresponding volume region becomes the least recently used. If a bit for a given region is marked dirty and another write to the same region occurs, the log does not need to be written to before the write operation occurs, thus reducing write overhead operations associated with using a DRL.

LSM maintains a limited number of dirty bits as still dirty even after the I/O was completed on all the volume's plexes. This balances the benefits of not having to log subsequent writes to the same region against how many regions are subsequently recovered if the system fails.

If you do not use DRL and the system fails, the LSM software must copy the full contents of a volume between its mirrors to restore and resynchronize all plexes. Although this process occurs in the background and the volume is still available, it can be a lengthy I/O-intensive procedure and may result in many areas of the volume being unnecessarily recovered.

1.3.6.3.2 Migrating From Block Change Logging to DRL

The DRL feature is a replacement for block change logging (BCL). When you import a disk group from a previous LSM version, either automatically during an upgrade installation or manually, existing mirrored volumes with BCL enabled are automatically reconfigured to DRL if the BCL log subdisk is at least 2 blocks. If the BCL log subdisk is less than 2 blocks, the volume is usable, but logging is disabled after the import.

A DRL must be configured with two or more sectors, preferably an even number because the last sector in a log with an odd number of sectors is not used. The log size is normally proportional to the volume size. If a volume is less than 2 GB, a log of two sectors is sufficient. The log subdisk size should then increase by two sectors for every additional 2 GB of volume size.

It is recommended that you use of the default log length provided by the `volassist` command.

If the BCL cannot be migrated to DRL, logging is disabled on that volume. Logging can be reenabled by removing the invalid logs and enabling DRL appropriately. See Section 5.3.3.3 for information on enabling DRL logging.

Note

A DRL log subdisk must be at least 65 blocks when using LSM in a TruCluster environment.

See the `vollogcnvt`(8) reference page for more information on migrating BCL to DRL.

1.3.6.4 Striped and Mirrored Volumes (RAID0 and RAID1)

A striped and mirrored volume spreads data across multiple disks while providing redundancy of data. Configuring a LSM volume to be both mirrored and striped is a common and effective way to improve both performance and availability for a volume. You create a striped and mirrored volume by configuring each of the volume’s data plexes, or mirrors, to have a stripe layout. You must allocate the striped plex and its mirror on separate disks for striping and mirroring to be effective.

1.3.6.5 RAID5 Volumes

Although both mirroring (RAID1) and RAID5 provide redundancy of data, the approaches differ. Mirroring provides data redundancy by maintaining multiple copies of a volume's data. Data written to a mirrored volume is duplicated in all copies. If a portion of a mirrored volume fails, the system continues to use the other copies of the data.

RAID5 provides data redundancy through the use of parity. While data is written to a RAID5 volume, a parity value is also calculated by performing an exclusive OR (XOR) procedure on data. The resulting parity is written to the volume. If a portion of a RAID5 volume fails, the data that was on that portion of the failed volume is recreated from the remaining data and the parity.

Note

Mirroring of RAID5 volumes is not currently supported.

A RAID5 volume in a TruCluster environment is not currently supported.

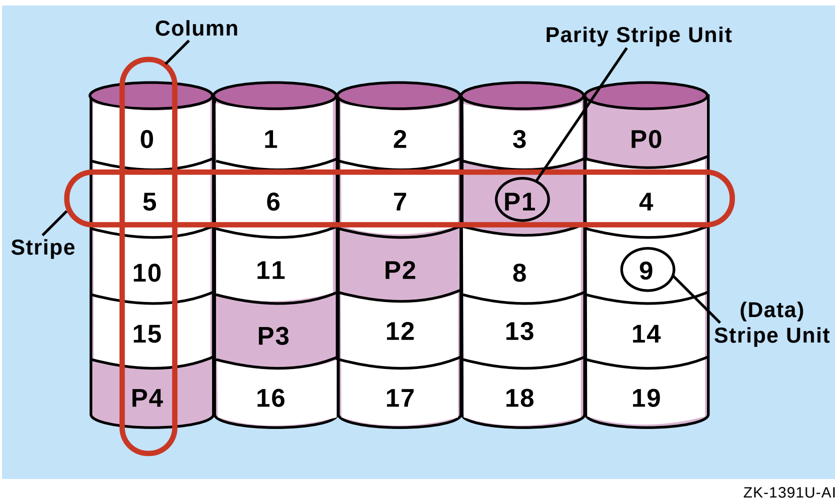
1.3.6.5.1 Left-Symmetric Layout

There are several layouts for data and parity that can be used for a RAID5 volume. The layout selected for the LSM implementation of a RAID5 volume is the left-symmetric layout. The left-symmetric layout provides optimal performance for both random I/O and large sequential I/O.

The left-symmetric layout stripes both data and parity across columns, placing the parity in a different column for every stripe of data. The first parity stripe unit is located in the right column of the first stripe. Each successive parity stripe unit is located in the next stripe, left-shifted one column from the previous parity stripe unit location. If there are more stripes than columns, the parity stripe unit placement begins in the right column again. Data is organized starting to the right of the parity stripe unit.

Figure 1–6 illustrates a left-symmetric parity layout consisting of five disks (one per column).

Figure 1–6: Left-Symmetric Layout



In Figure 1–6, data organization for the first stripe begins at P0 and continues to stripe units 0 through 3. Data organization for the second stripe begins at P1, then continues to stripe unit 4, and on to stripe units 5 through 7. Data organization proceeds in this manner for the remaining stripes.

Each parity stripe unit contains the result of an exclusive OR (XOR) procedure performed on the data in the data stripe units within the same stripe. If data on a disk corresponding to one column is inaccessible due to hardware or software failure, data is restored by performing an XOR procedure on the contents of the remaining columns' data stripe units against their respective parity stripe units (for each stripe).

For example, if the disk corresponding to the leftmost column in Figure 1–6 fails, then the volume is placed in a degraded mode. While in degraded mode, the data from the failed column is re-created by performing the XOR procedure on stripe units 1 through 3 against parity stripe unit P0 to recreate stripe unit 0, then performing the XOR procedure on stripe units 4, 6, and 7 against parity stripe unit P1 to re-create stripe unit 5, and so on.

Note

Failure of multiple columns in a plex with a RAID5 layout detaches the volume. This means that the volume cannot satisfy read or write requests. Once the failed columns are recovered, you might have to recover the user data from backups.

1.3.6.5.2 RAID5 Logging

If a disk and the system fail, it is possible for data not involved in active writes to be lost or corrupt. If this double failure occurs, it is unknown if the data or parity were actually written. Therefore, the recovery of the corrupted disk may be corrupted itself.

You can use RAID5 logging to prevent corruption of recovery data. A RAID5 log of the new data and parity is made on a persistent device, such as a disk device or nonvolatile RAM. The new data and parity are written to the disks.

You can associate a log with a RAID5 volume by attaching it as additional, non-RAID5 layout plexes. More than one log plex can exist per RAID5 volume, in which case the RAID5 logs are mirrored.

Note

The recommended procedure it to always use a log with RAID5 volumes. By default, `volassist` creates a log when you create a RAID5 volume.

1.4 Administering LSM Objects

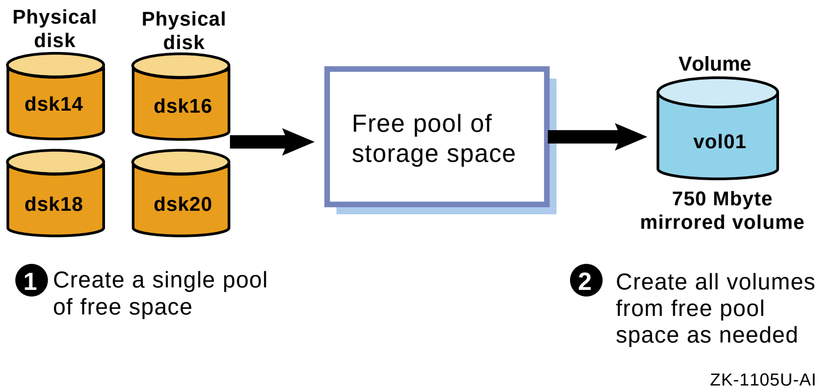
There are two methods to administering LSM objects:

- The top-down approach, where you use commands that automatically configure and create relationships between the LSM objects.
- The bottom-up approach, where you use commands to manually configure and create relationships between the LSM objects.

1.4.1 Top-down Approach

The top-down approach is the recommended method for most users because it allows the LSM software to manage the free disk space and control the relationship of LSM objects. As shown in Figure 1–7, the top-down approach involves a two-step process. First, LSM disks are placed into a disk group and managed as a single, large pool of free storage space. Then, as storage space is needed, you request disk space and LSM allocates the space from this free pool. Based on your specifications (for example, striped and mirrored volumes), the LSM software automatically allocates the storage from different LSM disks to satisfy the volume configuration requirements.

Figure 1–7: Top-Down Storage Allocation with the LSM software

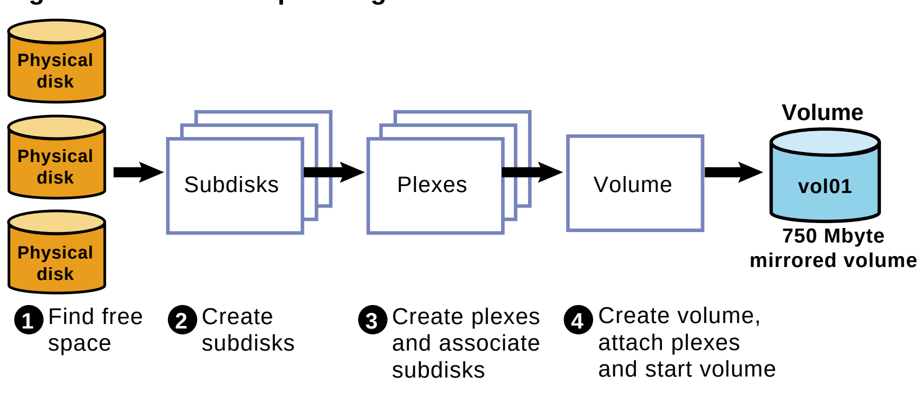


1.4.2 Bottom-Up Approach

The bottom-up approach is the recommended method for advanced users who want to manage the free disk space themselves, or who need more control over the relationship of LSM objects. This approach requires detailed knowledge of LSM technology and commands — especially when creating a mirrored, striped, or RAID5 volume in which the volume's subdisks are defined and configured on different disks.

As shown in Figure 1–8, the bottom-up approach requires considerably more user interaction and detailed knowledge of the LSM software, because you must explicitly perform each of the actions that are done automatically with the top-down approach — that is, find free space, create subdisks, create and associate plexes, create the volume, attach plexes, and start the volume.

Figure 1–8: Bottom-Up Storage Allocation with the LSM software



1.4.3 LSM Administrative Commands and Interfaces

You administer LSM objects using:

- A set of commands ranging from common commands (top-down) that require minimal user input to advanced commands (bottom-up) that require numerous parameter and option values. You must log on as root to use LSM commands.
- A menu-based, interactive interface called `voldiskadm`. This interface provides default values and online help.
- A java-based graphical user interface (GUI) called Storage Administrator. The Storage Administrator allows you to manage local or remote systems on which LSM is running.
- A bit-mapped GUI called Visual Administrator. The Visual Administrator allows you to view and manage disks and volumes, and perform limited file system administration. This GUI requires a bit-mapped display, the Basic X Environment software subset, and the LSM software license.

You can use the LSM interfaces interchangeably. That is, objects created by one interface are compatible with objects created by other LSM interfaces.

1.4.3.1 LSM Commands

LSM commands are divided into two main categories:

- Common commands that support the top-down approach for storage space management and common operations.
- Advanced commands that support the bottom-up approach for storage space management and specialized operations.

Table 1–4 lists the common LSM commands and their functions.

Table 1–4: Common LSM Commands

Command	Function
volsetup	Initialize LSM and rootdg (used once after a new installation)
voldiskadd	Interactively add a disk or list of disks for use with LSM
voldiskadm	Run the interactive menu interface to do common disk-related operations
voldisksetup	Add a disk or a list of disks for use with LSM (with -i option)
voldisk	Display LSM disk information (and other functions)
voldg	Display disk groups (and other functions)
volassist	Create, mirror, back up, and move volumes

Table 1–4: Common LSM Commands (cont.)

Command	Function
volprint	Display LSM configuration information
volmirror	Mirror volumes on a disk
volrestore	Backup or restore the LSM configuration database
volwatch	Monitor LSM for failure events and perform hot sparing
volevac	Evacuate all volumes from a disk
volrecover	Resynchronize plexes after a crash or disk failure
volencap	Create a script to encapsulate partitions that contain existing data
volreconfig	Run a script to perform encapsulation
volrootmir	Mirror the root and swap volumes
lsmsa	Start the Storage Administrator GUI

Table 1–5 lists the advanced LSM commands and their functions.

Table 1–5: Advanced LSM Commands

Command	Function
volinstall	Customize the LSM environment
voldisk	Manage LSM disks
voldg	Manage LSM disk groups
volmake	Create LSM objects (volume, plex, subdisk)
volsd	Perform LSM operations on subdisks
volplex	Perform LSM operations on plexes
volume	Perform LSM operations on volumes
voledit	Create, modify, and remove LSM records
volstat	Display LSM statistics
voldctl	Control daemon operations
volmend	Mend simple problems in configuration records
volnotify	Display LSM configuration events
voltrace	Trace I/O operations on volumes

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

1.4.3.2 The `voldiskadm` Menu Interface

The `voldiskadm` menu interface allows you to perform basic disk and volume procedures.

You start the `voldiskadm` menu interface by entering the following command:

voldiskadm

Example 1–1 shows the `voldiskadm` main menu. To perform a procedure, choose an operation from the main menu and the `voldiskadm` command prompts you for information. The `voldiskadm` command provides default values when possible. You can press Return to use the default value or enter a new value. Enter `?` at any time to view online help.

Example 1–1: Main Menu for the LSM voldiskadm Interface

```
Logical Storage Manager Support Operations
Menu: VolumeManager/Disk

 1      Add or initialize one or more disks
 2      Encapsulate one or more disks
 3      Remove a disk
 4      Remove a disk for replacement
 5      Replace a failed or removed disk
 6      Mirror volumes on a disk
 7      Move volumes from a disk
 8      Enable access to (import) a disk group
 9      Remove access to (deport) a disk group
 10     Enable (online) a disk device
 11     Disable (offline) a disk device
 12     Mark a disk as a spare for a disk group
 13     Turn off the spare flag on a disk
 14     Recover plexes and volumes after disk replacement
 list   List disk information

 ?      Display help about menu
 ??     Display help about the menuing system
 q      Exit from menus

Select an operation to perform:
```

See the Appendix D and the `voldiskadm`(8) reference page for more information on `voldiskadm` interface.

1.4.3.3 The Storage Administrator

The Storage Administrator is a java-based GUI that displays a hierarchical view of LSM objects and their relationships.

You use the Storage Administrator to view and manage LSM objects on a local or remote (client) system. The Storage Administrator consists of a server (daemon) and a client. The server daemon runs on a system on which LSM is initialized and running. The client runs on a system that supports the Java run time environment.

The Storage Administrator provides dialog boxes in which you enter information to create or manage LSM objects. Completing a dialog box may be the equivalent of entering several commands.

See Chapter 9 for more information on the Storage Administrator.

1.4.3.4 The Visual Administrator

The Visual Administrator is a GUI that you can use to manage LSM disks, volumes, and some basic file-system objects. The Visual Administrator displays windows in which LSM objects are represented as icons. You use menus and mouse point-and-click and drag-and-drop techniques to select and manage LSM objects. The Visual Administrator uses LSM commands to carry out operations.

See Appendix B for more information on the Visual Administrator.

1.5 LSM Architecture and Components

The LSM device driver is between the file systems and the disk device drivers. An LSM-built kernel includes volume device drivers that provide a level of abstraction between the disks and the file systems or third-party databases. The file systems and databases are placed on LSM volumes and perform I/O requests to an LSM volume in the same way that they perform I/O requests to any other disk driver.

Once an LSM volume is defined and configured, the file systems and databases issue I/O requests directly to the LSM volume, not to the device drivers.

Figure 1–9 shows the relationships between applications, the kernel, file systems, device drivers, and the LSM configuration databases.

Figure 1–9: LSM Architecture

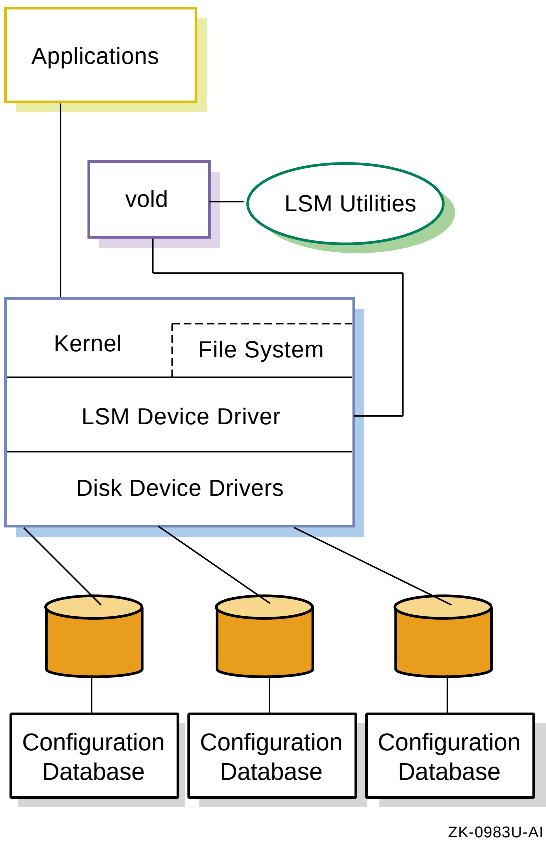


Table 1–6 describes LSM components and their functions.

Table 1–6: LSM Components

Component	Function
Volume configuration daemon (`vold`)	A daemon that: - Provides the interface between the LSM software and the kernel - Takes requests for configuration changes from other utilities, makes the changes, and communicates them to the kernel - Initializes the LSM software when the system boots
Volume device driver (`volconfig`)	An interface that LSM drivers use for loading or changing the kernel's LSM configuration.

Table 1–6: LSM Components (cont.)

Volume extended I/O daemons (voliod)	Internal kernel threads that the LSM driver uses for LSM error handling and recovery, and additional I/O threads to improve performance when writing to a mirror or a DRL.
/etc/vol/volboot file	A configuration file that the LSM software uses when it starts to locate the LSM configuration databases.
Configuration database	A database that contains records describing all the LSM objects and layouts (volumes, plexes, subdisks, disk media names, and disk access names) in a disk group. Typically, the LSM software maintains multiple copies of the configuration database in a disk group in case of a disk failure.

2

Setting Up the LSM Software

This chapter describes how to setup up the LSM software, which includes:

- Installing or upgrading the LSM software subsets.
- Initializing the LSM software, which prepares the system and disks for use with the LSM software.

2.1 Installing or Upgrading the LSM Software

The way you install the LSM software depends on if you are:

- Installing the LSM software for the first time during a full installation. See the *Installation Guide* for more information.
- Installing the LSM software for the first time after a full installation. See the *Installation Guide* for more information.
- Performing a full installation on a system that was previously using the LSM software. See Section 2.1.1 for more information.
- Performing an upgrade installation on a system with the LSM software. See Section 2.1.2 for more information.

Note

The LSM Versions 5.0 software and higher have an on-disk LSM internal metadata format that is not compatible with the previous versions of the LSM software. That is, LSM Version 5.0 or higher cannot use the metadata format from previous LSM versions, nor can a previous LSM version use the metadata format from LSM Version 5.0 or higher.

If the LSM software detects an older metadata format within a disk's private region during a disk group import, LSM automatically converts the old format to the new format. Once converted to the new format, you can no longer use a disk group with previous versions of LSM.

2.1.1 Performing a Full Installation on an System With LSM

If the system is running the LSM software, follow these steps before performing a full installation:

1. Check for a previous version of the LSM software by entering the following command:

   \# **voldisk list rz2 | grep version**

   Output similar to the following is displayed:

   ```
   version: 1.1
   ```

2. Optionally, prevent non-`rootdg` disk groups from automatically being converted to the new LSM internal metadata format during the full installation process.

   For example, to prevent disk groups called `dg1` and `dg2` from being converted and used with the new version of the LSM software, enter:

   \# **voldg deport dg1 dg2**

3. Determine the previous hostid configured with the LSM software by entering the following command:

   \# **/sbin/voldctl list**

   Output similar to the following is displayed:

   ```
   Volboot file
   version: 3/1
   seqno:   0.2
   hostid:  rio.dec.com
   entries:
   disk rz2 type=sliced
   disk rz3 type=sliced
   disk rz8 type=sliced
   disk rz9 type=sliced
   ```

   In this output the hostid is `rio.dec.com`

4. Save the current LSM configuration by entering the following command:

   \# **volsave**

   Output similar to the following is displayed:

   ```
   LSM configuration being saved to /usr/var/lsm/db/LSM.date.rio
   LSM Configuration saved successfully to /usr/var/lsm/db/LSM.date.rio
   ```

5. Confirm that the LSM configuration was saved by entering the following command:

   \# **ls /usr/var/lsm/db/LSM.*date*.rio**

   Output similar to the following is displayed:

   ```
   header        rootdg.d      volboot       voldisk.list
   ```

6. Save the LSM configuration to tape or other removable media.

The LSM software is reinitialized if during the full installation you selected to install the system's `root`, `usr`, `var`, and `swap` partitions directly onto LSM volumes .

If the system's `root`, `usr`, `var`, and `swap` partitions are not installed directly onto LSM volumes during the full installation, select and configure the LSM subsets during the operating system installation process. See the *Installation Guide* for more information on installing the operating system software.

After installing the operating system, follow these steps to reinitialize the LSM software to use previous disk group configurations:

1. Either:

    a. Restore the `/etc/vol/volboot` file by entering the following command:

    ```
    # cp /backup/usr/var/lsm/db/LSM.date.hostname/ \
    volboot /etc/vol/volboot
    ```

    b. Or, create a new `/etc/vol/volboot` file using the hostid obtained in the previous Step 3. For example, to create a new `/etc/vol/volboot` file for a system with a hostid of `rio`, enter:

    ```
    # voldctl init rio.dec.com
    ```

2. Reinitialize the LSM special device files and start the LSM daemons and volumes by entering the following command:

    ```
    # volsetup
    ```

Warning

Do not use the force option with the `volsetup` command. Doing so destroys the previous LSM configuration for the `rootdg` disk group.

2.1.2 Performing an Upgrade Installation on a System with LSM

If the LSM software was initialized on a system before an upgrade installation, be sure to select the LSM subsets during the upgrade installation process. If one of the file systems was configured on an LSM volume, you must start the LSM software and its volumes before proceeding with the upgrade installation after booting the system to single-user mode.

See the *Installation Guide* for more information on installing the LSM software.

2.2 Initializing the LSM Software For the First Time

You must initialize the LSM software if during a full installation you did not install the system's file systems into LSM volumes, or if you did not perform an upgrade installation on a system that was previously running the LSM software.

Use one of the following methods to initialize the LSM software for the first time:

- While installing the operating system software, select the option to install file systems directly to LSM volumes. See the *Installation Guide* for more information on initializing the LSM software while install the operating system software.
- Use the `volsetup` command. This is the simplest method to initialize the LSM software. The `volsetup` command automatically provides a default configuration that is suitable for most environments. See Section 2.2.2 for more information on initializing the LSM software using the `volsetup` command.
- Use a series of LSM commands. Although this method is more complicated, it allows you to have more control over your LSM configuration. See Section 2.2.3 for more information on initializing the LSM software using commands.

Initializing the LSM software:

- Modifies the `/etc/inittab` file to include LSM entries that automatically starts the LSM software when the system boots.
- Allows you to create, initialize, and add disks to the `rootdg` disk group. You must configure at least one disk or partition in the `rootdg` disk group. You do not have to use the `rootdg` disk group, however it must exist before you can create other disk groups.
- Verifies the disk labels for the disks in the `rootdg` disk group. If any disks were previously used, and the `fstype` field for any partition is anything other than `unused`, you must reinitialize the disk label.
- Creates the `/etc/vol/volboot` file, which contains:
  - The host ID that the LSM software uses to establish ownership of physical disks. The host ID ensures that two or more nonclustered systems can access disks on a shared SCSI bus without interfering with each other.

– An optional list of the disks in the rootdg disk group that contain the LSM configuration database.

Do not manually edit or delete the /etc/vol/volboot file. Use the voldctl command to update the /etc/vol/volboot file.

- Starts the vold daemon. The vold daemon receives requests from other utilities for configuration changes, modifies configuration information stored on disk, and communicates the changes to the kernel.
- Sets the number of configuration databases. The size of the configuration database depends on the number of LSM objects (volume, plex, subdisk, and disk). Each LSM object created in the disk group requires one record in the configuration database. Approximately two records can fit in one sector (512 bytes). Disks that are added to the rootdg disk group need one record for each disk in all other disk groups. Certain LSM configuration changes (for example, moving a subdisk) involve creating and later deleting two to four temporary records in the configuration database to perform the operation.
- Sets the number of log regions and the size of the private region. The default private region size is 4096 blocks (512 bytes).
- Starts the LSM software.

2.2.1 Before You Initialize the LSM Software

Before you initialize the LSM software, you should:

- Verify that the LSM subsets are installed by entering the following command:

  # **setld -i | grep LSM**

  Output similar to the following is displayed:

  ```
  OSFLSMBASE500  installed Logical Storage Manager (System Administration)
  OSFLSMBIN500   installed Logical Storage Manager Kernel Modules (Kernel
                 Build Environment)
  OSFLSMX11500   installed Logical Storage Manager GUI (System Administration)
  ```

  If the LSM subsets do not display with a status of installed, use the setld command to install them. See the *Installation Guide* for more information on installing the LSM subsets.

- Verify that the LSM drivers are configured into the kernel by entering the following command:

  # **devswmgr -getnum driver=LSM**

  Output similar to the following is displayed:

  ```
  Device switch reservation list
  ```

```
                                (*=entry in use)
          driver name             instance   major
  ------------------------------   --------   -----
                             LSM          4      43
                             LSM          3      42
                             LSM          2      41*
                             LSM          1      40*
```

If the LSM drivers do not display, you must rebuild the kernel using the `doconfig` command. See the *Installation Guide* for more information on rebuilding the kernel.

- Identify the disks that you want to use with the LSM software by entering the following command:

  \# **file /dev/rdisk/dsk*c**

  Note the device names in the output.

- Choose a disk to use for the `rootdg` disk group. Enter the `disklabel`, `swapon`, `mount`, or `showfdmn` command (as shown in the output that follows) to identify the disk partitions already in use. Also, check the configuration for any other third-party software that uses raw partition:

  \# **disklabel dsk0**

  Output similar to the following is displayed:

```
# /dev/rdisk/dsk0a:
8 partitions:
#        size   offset    fstype   [fsize bsize   cpg]
  a:   262144        0     AdvFS                          # (Cyl.    0 - 328*)
  b:   262144   262144      swap                          # (Cyl.  328*- 657*)
  c:  2050860        0    unused        0     0           # (Cyl.    0 - 2569)
  d:   508857   524288    unused        0     0           # (Cyl.  657*- 1294*)
  e:   508857  1033145    unused        0     0           # (Cyl. 1294*- 1932*)
  f:   508858  1542002    unused        0     0           # (Cyl. 1932*- 2569)
  g:  1526572   524288     AdvFS                          # (Cyl.  657*- 2569)
  h:        0        0    unused        0     0           # (Cyl.    0 - -1)
```

  \# **swapon -s**

  Output similar to the following is displayed:

```
Swap partition /dev/disk/dsk6b (default swap):
...
```

  \# **mount**

  Output similar to the following is displayed:

```
root_domain#root on / type advfs (rw)
usr_domain#usr on /usr type advfs (rw)
var_domain#var on /var type advfs (rw)
staff1#staff1 on /share/demo2/usr/staff1 type advfs (rw)
...
```

  \# **showfdmn root_domain usr_domain var_domain staff1 | **
  grep dev

Output similar to the following is displayed:

```
1L     262144       39712     85%     on    256    256   /dev/disk/dsk1a
1L    2050848      582240     72%     on    256    256   /dev/disk/dsk2c
2     1191936      444624     63%     on    256    256   /dev/disk/dsk23d
1L    2050848      883136     57%     on    256    256   /dev/disk/dsk5c
1L    4110480     1800880     56%     on    256    256   /dev/disk/dsk7c
2     4110480     1804384     56%     on    256    256   /dev/disk/dsk20c
3     4110480     1794688     56%     on    256    256   /dev/disk/dsk14c
```

If you don't know the domain names, enter:

ls -lagR /etc/fdmns

- If you are in single-user mode, set the host name for your system before you initialize the LSM software by entering the following command:

  # **/sbin/hostname -s *name***

- Verify that the PATH environment variable includes the /usr/sbin and/sbin directories. This simplifies the use of LSM commands, which are located in both of these directories.

2.2.2 Initializing the LSM Software Using The `volsetup` Command

The volsetup command automatically initializes the LSM software by:

- Modifying disk labels
- Initializing disks for use with the LSM software
- Creating the rootdg disk group
- Configuring disks into the rootdg disk group

You enter the volsetup command only once. To add more disks, you can use the voldiskadd command, as described in Section 6.2.1.

Initialize the LSM software by entering the following command:

volsetup *disk_name*

If you omit the name of a disk, the volsetup command prompts you for it.

For example, to initialize the LSM software using a disk called dsk4 to create the rootdg disk group, enter:

volsetup dsk4

Note

When you initialize the LSM software, do not specify the boot disk with the volsetup command. After you initialize the LSM software, you can encapsulate the boot disk to add partitions on the boot disk to the rootdg disk group. See Chapter 4 for more information on encapsulating the boot disk.

If the volsetup command displays an error message or if the initialization fails, you may need to modify the disk label and reinitialize the disk. See the volsetup(8) reference page for more information.

2.2.3 Initializing the LSM Software Using Commands

Using the volsetup command to initialize the LSM software for the first time is the most common and easiest way to set up the LSM software as described in Section 2.2.2. However, if you require more control over how the LSM software is set up, you can use a series of commands instead of the volsetup command.

Follow these steps to use a series of LSM commands to initialize the LSM software:

1. Adds entries to the /etc/inittab file that automatically start LSM when the system boots by entering the following command:

   # **volinstall**

   If the volinstall command fails, then:

   - Verify that the/etc/inittab file was modified to include LSM entries by entering the following command:

     # **grep LSM inittab**

     Output similar to the following is display:

     ```
     lsmr:s:sysinit:/sbin/lsmbstartup -b /dev/console 2>&1 ##LSM
     lsm:23:wait:/sbin/lsmbstartup -n /dev/console 2>&1 ##LSM
     vol:23:wait:/sbin/vol-reconfig -n /dev/console 2>&1 ##LSM
     ```

   - Verify that the LSM special files were created by entering the following command:

     # **ls -l /dev/vol***

     Output similar to the following is display:

     ```
     crw-r--r--   1 root   system  41,  0 Mar  4 09:35 /dev/volconfig
     crw-r--r--   1 root   system  41,  3 Mar  4 09:35 /dev/volinfo
     crw-r--r--   1 root   system  41,  2 Mar  4 09:35 /dev/voliod
     crw-r--r--   1 root   system  41,  1 Mar  4 09:35 /dev/voltrace
     ```

2. Start the vold daemon in the disabled mode by entering the following command:

   # **vold -m disable**

3. Create and initialize the /etc/vol/volboot file by entering the following command:

   # **voldctl init**

4. Create the rootdg disk group by entering the following command:

   # **voldg init rootdg**

Warning

Enter the voldg init command only once to create a disk group. If you use the voldg init command a second time, you will destroy configuration information.

5. Verify that the rootdg disk group was created by entering the following command:

   # **volprint**

   Output similar to the following is displayed:

```
Disk group: rootdg

TY NAME    ASSOC      KSTATE  LENGTH    PLOFFS   STATE    TUTIL0  PUTIL0

dg rootdg  rootdg     -       -         -        -        -       -
dm dsk4    dsk4       -       1854536   -        -        -       -
```

6. Verify that the disk label for disks to be used with the LSM software have a fstype status of unused by entering the following command:

   # **disklabel *disk_name***

   For example, to display the disk label for a disk called dsk10, enter:

   # **disklabel dsk10**

   Output similar to the following is displayed:

```
# /dev/rdisk/dsk10c:
type: SCSI
disk: RZ1BB-CS
label:
flags: dynamic_geometry
bytes/sector: 512
sectors/track: 86  tracks/cylinder: 16
sectors/cylinder: 1376
cylinders: 3045
sectors/unit: 4110480
rpm: 7228
interleave: 1
trackskew: 40
cylinderskew: 80
headswitch: 0           # milliseconds
track-to-track seek: 0  # milliseconds
drivedata: 0
```

```
8 partitions:
#        size   offset   fstype [fsize bsize cpg] # NOTE: values not exact
  a:   131072        0   unused      0     0      # (Cyl.    0 - 95*)
  b:   262144   131072   unused      0     0      # (Cyl.   95*- 285*)
  c:  4110480        0   unused      0     0      # (Cyl.    0 - 2987*)
  d:        0        0   unused      0     0      # (Cyl.    0 - -1)
  e:        0        0   unused      0     0      # (Cyl.    0 - -1)
  f:        0        0   unused      0     0      # (Cyl.    0 - -1)
  g:  1858632   393216   unused      0     0      # (Cyl.  285*- 1636*)
  h:  1858632  2251848   unused      0     0      # (Cyl. 1636*- 2987*)
```

If the disk is no longer in use, but the `fstype` field for any partition is anything other than `unused`, you must initialize the disk label. For example, to initialize the disk label for a disk called `dsk2`, enter:

\# **disklabel -wr dsk2**

If you receive an error message that the disk does not start at block zero, enter the following commands:

\# **disklabel -z *disk_name***

\# **disklabel -wr *disk_name***

7. Repartition and initialize the LSM private region on the disk by entering the following command:

   \# **voldisksetup -i *disk_name***

   For example, to repartition and initialize the LSM private region on a disk called `dsk9`, enter:

   \# **voldisksetup -i dsk9**

   Display the results by entering the following command:

   \# **disklabel dsk9 | grep LSM**

   Output similar to the following is displayed:

```
g:    4106384          0   LSMpubl          # (Cyl.    0 - 2984*)
h:       4096    4106384   LSMpriv          # (Cyl. 2984*- 2987*)
```

   LSM automatically maintains the number of active configuration databases and the location of the databases for a disk group. LSM dynamically evaluates and, if needed, activates or deactivates a configuration database within a disk's private region when a disk is added, removed, or fails. Therefore, it is not necessary to explicitly specify the location and number of configurations on a disk.

   See the `voldisksetup`(4) and `voldisk`(4) reference pages for more information on disk initialization options.

8. Add a disk to the `rootdg` disk group by entering the following command:

   \# **voldg adddisk *disk_name***

   For example, to add a disk called `dsk9` to the `rootdg` disk group, enter:

voldg adddisk dsk9

9. Enable the `vold` daemon by entering the following command:

   # **voldctl enable**

10. Set the number of LSM I/O daemons, which is either two or the number of central processing units (CPUs) on the system, whichever is greater. For example, on a single CPU system, enter:

    # **voliod set 2**

    On a four CPU system, enter:

    # **voliod set 4**

    You only need to set the LSM I/O daemons the first time you initialize the LSM software. The correct number of I/O daemons is correctly set when the system boots.

2.2.4 Verifying that the LSM Software was Initialized

Follow these steps to verify that the LSM software is initialized:

1. Verify that the disk was added to the `rootdg` disk group by entering the following command:

   # **volprint**

   Output similar to the following is displayed:

```
Disk group: rootdg

TY NAME    ASSOC      KSTATE  LENGTH   PLOFFS   STATE    TUTIL0  PUTIL0

dg rootdg  rootdg     -       -         -        -        -       -
dm dsk4    dsk4       -       1854536   -        -        -       -
```

2. Verify that the `vold` daemon is enabled by entering the following command:

   # **voldctl mode**

   Output similar to the following is displayed:

   `mode: enabled`

3. Verify that two or more `voliod` daemons are running by entering the following command:

   # **voliod**

   Output similar to the following is displayed:

   `2 volume I/O daemons are running`

4. Verify that the `/etc/inittab` file was modified to include LSM entries by entering the following command:

grep LSM /etc/inittab

Output similar to the following is displayed:

```
lsmr:s:sysinit:/sbin/lsmbstartup -b /dev/console 2>&1 ##LSM
lsm:23:wait:/sbin/lsmbstartup -n /dev/console 2>&1 ##LSM
vol:23:wait:/sbin/vol-reconfig -n /dev/console 2>&1 ##LSM
```

5. Verify that the `/etc/vol/volboot` file was created by entering the following command:

   # **/sbin/voldctl list**

   Output similar to the following is displayed:

```
Volboot file
version: 3/1
seqno:   0.4
hostid:  test.abc.xyz.com
entries:
```

2.3 Using the LSM Software

After the LSM software is set up, you can:

- Migrate, or encapsulate, existing file systems or data into LSM volumes. Encapsulating file systems or data allows you to configure existing file systems or data into LSM volumes, without physically moving the data. See Chapter 3 for more information.
- Migrate, or encapsulate, the partitions on the boot disk into LSM volumes. Encapsulating the boot disk allows you to configure the partitions on the boot disk into LSM volumes, without physically moving the data. See Chapter 4 for more information.
- Configure new disks for use with the LSM software and create LSM volumes. See Chapter 5 for more information.

Note

LSM does not support encapsulation of data on ULTRIX Disk Shadowing (UDS) volumes or ULTRIX Striping Driver stripe volumes.

3

Encapsulating Existing Data to LSM Volumes

After the LSM software is set up, as described in Chapter 2, you may want to migrate, or encapsulate, existing file systems or data into LSM volumes. If you do not want to do this, see Chapter 5 for information on how to configure new disks for use with the LSM software and how to create LSM volumes.

LSM provides commands that you use to configure existing file systems and data into LSM volumes, without physically moving the data. This process is called encapsulation. LSM supports data encapsulation of:

- The boot disk
- Disks and disk partitions
- Advanced File System (AdvFS) storage domains

During the encapsulation process, the LSM software transforms disks, partitions, or AdvFS file domains into LSM volumes. You supply a device name, a device's partition, or an AdvFS domain name that the LSM software uses to identify how the device is used, and LSM creates an LSM volume to cover those areas on the disk.

This chapter describes how to encapsulate disks, partitions, and AdvFS file domains. See Chapter 4 for information on encapsulating the boot disk.

3.1 Encapsulation Requirements

The following conditions must be met before you can encapsulate data:

- The `rootdg` disk group must exist and be enabled, unless you are encapsulating the boot disk. See Chapter 4 for information on encapsulating the boot disk.

  When encapsulating a disk partition, an LSM `nopriv` disk and a volume that maps to that disk partition is created in the `rootdg` disk group.

- Back up all user data before beginning the encapsulation process. In the event that a failure occurs during the encapsulation process, you can restore the saved data to return the disks to their original states.

- The data within the disk partition must be unmounted or off line for the encapsulation to proceed. If you cannot unmount or take the data off line, then the encapsulation process must reboot the system.

3.2 Encapsulating Disks and Disk Partitions

The encapsulation process changes disks and disk partitions into LSM disks.

The partitions on a disk are mapped by a partition table using the `disklabel` command. The partition and disk label have the following characteristics:

- The disk label maps a physical disk to a maximum of eight partitions that are labeled from `a` to `h`.
- Partition `c` covers the whole disk.
- Partitions can overlap each other, however no set of overlapping partitions can be in use at the same time.
- Each partition has an `fstype` attribute, which identifies the file system type associated with the partition. If a partition is:
  - Not in use, the `fstype` is `unused`.
  - In use, the `fstype` displays values such as `4.2BSD`, `AdvFS`, `swap`, `LSMsimp`, `LSMpubl`, `LSMpriv`, or `LSMnopriv`.
- The disk label resides at block 0 of a physical device. For some devices, the default disk label is defined in `/etc/disktab` file.

Each available partition has a special device file in the `/dev` directory. Users and applications access storage through these special device files.

The LSM encapsulation process uses information in the disk label and the `/etc/fstab` file to find out if a partition is in use, for example if it contains a UFS file system. If the partition does not have information in a disk label or `/etc/fstab` file, the encapsulation process uses the partition name.

Follow these steps to encapsulate a disk or disk partitions:

1. Create the LSM script that contain the LSM commands that creates LSM volumes for a disk or disk partitions by entering the following command:

   \# **volencap {*disk_name* | *partition_name*}**

   For example, to create an encapsulation script for a disk called `dsk3`, enter:

   \# **volencap dsk3**

To create an encapsulation script for partitions called `dsk4g` and `dsk5h`, enter:

```
# volencap dsk4g dsk5h
```

2. Complete the encapsulation process by entering the following command:

```
# volreconfig
```

If any partition or disk that was encapsulated is still in use, the `volreconfig` command prompts you to reboot the system.

After the encapsulation, LSM converts each disk or partition to a `nopriv` LSM disk. LSM then creates a subdisk that maps the entire disk or partition. LSM uses the subdisk to create a plex and, in turn, uses the plex to create an LSM volume. 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.

3.3 Encapsulating AdvFS Domains

If an AdvFS domain consists of one disk partition, you can encapsulate it for use with the LSM software using the procedure described in Section 3.2. If the AdvFS domain consists of multiple disk partitions, it may be more convenient to encapsulate the AdvFS domain instead of the individual disk partitions. When encapsulating the AdvFS domain, each physical device in the domain is encapsulated into an LSM volume by changing the links in the domain tree to point to the LSM volumes.

LSM volumes encapsulated from domain physical devices must reflect the exact data at the exact Logical Block Number (LBN) location as the physical device. The entire LBN range of the LSM `nopriv` disk is defined as one LSM subdisk. A plex is created with this subdisk and an LSM volume is created with the plex.

No mount point changes are necessary during encapsulation, because the mounted filesets are abstractions to the domain. The domain can be activated normally after the encapsulation process completes. Once the domain is activated, the filesets remain unchanged and the encapsulation is transparent to AdvFS domain users.

Follow these steps to encapsulate an AdvFS domain:

1. Make sure that the AdvFS domain is not in use and all filesets are unmounted.
2. Create the LSM script that contain the LSM commands that creates LSM volumes for an AdvFS domain by entering the following command:

volencap ***domain_name***

For example, to create an encapsulation script for an AdvFS domain called dom1, enter:

volencap dom1

3. Complete the encapsulation process by entering the following command:

   # **volreconfig**

   If the AdvFS file domain is mounted, the volreconfig command prompts you to reboot the system.

The /etc/fdmns directory is updated on successful creation of LSM volumes.

3.4 Preserving Block 0

Block 0 on a disk device contains the disk label and is read-only by default. UFS, AdvFS, and swap do not use block 0 when putting data on device partitions. To preserve the Logical Block Number (LBN) mapping for a disk partition that includes block 0 on the disk, the LSM nopriv disk must also start at LBN block 0. As long as this disk is used for UFS, AdvFS, and swap volumes, this does not present a problem.

However, if the LSM nopriv disk is later reused for applications that write to block 0, then a write failure occurs. To help avoid such failures, you should remove this nopriv disk from the LSM disk group and readd the disk as a sliced disk if the volume is no longer needed.

The voldiskadd and voldisksetup commands automatically map out block 0 when adding sliced or simple disks for use with the LSM software. It is recommended that you use these commands to add disks to LSM. See Chapter 5 for more information on adding disk for use with the LSM software.

3.5 Unencapsulating LSM Volumes

When you unencapsulate an LSM volume, the LSM nopriv disk is removed, and the disk label's fstype filed is marked unused. A disk label type of unused does not check for overlaps and can cause a user to inadvertently overwrite the disk without warning. Use the disklabel command to set the fstype field to the appropriate file system type. For example, a disk partition used for AdvFS should be labeled AdvFS. A disk partition used for used for UFS should be labeled 4.2BSD. See the disklabel(8) reference page for information on setting the fstype field.

3.5.1 Unencapsulating UNIX Partitions

Follow these steps to unencapsulate LSM volumes and return to using disk partitions:

1. Disable the volume by entering the following command:

   # **volume [-g *disk_group*] stop *volume_name***

   For example, to disable a volume called `vol-dsk4g` in the `staffdg` disk group, enter:

   # **volume -g staffdg stop vol-dsk4g**

2. Remove the volume recursively by deleting the volume, plexes, and subdisks by entering the following command:

   # **voledit [-g *disk_group*] -rf rm *volume_name***

   For example, to recursively remove a volume called `vol-dsk4g` from the `staffdg` disk group enter:

   # **voledit -g staffdg -rf rm vol-dsk4g**

3. Remove the disk media records from LSM control by entering the following command:

   # **voldg rmdisk *disk_name***

   For example, to remove the disk media record for a disk called `dsk4g`, enter:

   # **voldg rmdisk dsk4g**

4. Remove the disk access records from LSM control by entering the following command:

   # **voldg rm *disk_name***

   For example, to remove the disk access record for a disk called `dsk4g`, enter:

   # **voldisk rm dsk4g**

5. Use the `disklabel` command to set the `fstype` field to the appropriate file system type. See the `disklabel`(8) reference page for information on setting the `fstype` field.

6. Edit the `/etc/fstab` file to mount the file system that is on the partition.

3.5.2 Unencapsulating AdvFS Domains

Follow these steps to unencapsulate LSM volumes and return to using AdvFS domains. The following example moves the volume called vol-dsk4e, in the staffdg disk group on the dsk4e partition to the AdvFS domain called dom1.

1. Make sure that the volume is not mounted or in use, then disable the volume by entering the following command:

   ```
   # volume [-g disk_group] stop volume_name
   ```

   For example, to disable a volume called vol-dsk4e in the staffdg disk group, enter:

   ```
   # volume -g staffdg stop vol-dsk4e
   ```

2. Remove the volume recursively by deleting the volume, plexes, and subdisks by entering the following command:

   ```
   # voledit [-g disk_group] -rf rm volume_name
   ```

   For example, to recursively remove a volume called vol-dsk4e from the staffdg disk group enter:

   ```
   # voledit -g staffdg -rf rm vol-dsk4e
   ```

3. Remove the disk media records from LSM control by entering the following command:

   ```
   # voldg rmdisk disk_name
   ```

   For example, to remove the disk media record for a disk called dsk4e, enter:

   ```
   # voldg rmdisk dsk4e
   ```

4. Remove the disk access records from LSM control by entering the following command:

   ```
   # voldg rm disk_name
   ```

   For example, to remove the disk access record for a disk called dsk4e, enter:

   ```
   # voldisk rm dsk4e
   ```

5. Change the domain disk link by entering the following commands:

   ```
   # cd /etc/fdmns/domain_name
   # rm volume_name
   # ln -s /dev/disk/disk_name disk_name
   ```

6. Use the disklabel command to set the fstype field to the appropriate file system type. See the disklabel(8) reference page for information on setting the fstype field.

4

Encapsulating and Mirroring the Boot Disk

This chapter describes how to encapsulate and create mirrored volumes for all the partitions on the boot disk including the root file system, swap, `usr`, and `var` partitions.

Encapsulating and creating mirrored volumes for the partitions on the boot disk provides complete redundancy and recovery capability in the event of a boot disk failure. For example, if the system experiences failure of the primary boot disk, you can boot the system using the mirrored volumes.

4.1 Boot Disk Encapsulating and Mirroring Overview

Encapsulating the boot disk creates the necessary scripts that convert all the partitions on the boot disk to LSM volumes.

Mirroring the boot disk automatically creates mirrored volumes for:

- The root file system. This partition is called the root volume or `rootvol`. During startup, the system sees the `rootvol` volume as a regular partition, and accesses it using standard partition naming.
- The primary swap partition if it is on the boot disk. This partition is called the swap volume or `swapvol`. Other volumes that you use for swap (secondary swap volumes) are treated as ordinary LSM volumes.
- The `usr` partition if it is on the boot disk. This partition is called the `usr` volume or `usrvol`.
- The `var` partition if it is on the boot disk. This partition is called the `var` volume or `varvol`.

4.1.1 Root and Swap Usage Types

A volume that contains a file system has a usage type of `fsgen`. A volume that does not contain a file system has a usage type of `gen`. Because the root file system and swap partitions need special handling during system startup, LSM automatically sets up and uses the following special usage:

- `root`

This type is intended for volumes used as root devices. Because the root volume contains a file system, the root usage type resembles the `fsgen` usage type. The root usage type restricts the configuration of the volume such that all plexes of the volume are accessible as a root device through normal disk drivers.

- `swap`

  This type is intended for volumes used as the primary swap device. Unlike the root device, a swap device does not contain a file system; therefore, it resembles the `gen` usage type.

4.1.2 Boot Disk Encapsulation and Mirror Restrictions

LSM setups and configures the boot disk for use with the LSM software . Because the boot disk needs to be handled specially, you should be aware of the following restrictions:

- The root volume, swap volume, `usr` volume, `var` volume, and volumes created for any secondary swap volumes must be in the `rootdg` disk group.
- The `rootvol` volume cannot be striped.
- The `rootvol` and `swapvol` volumes must not span or contain a plex with multiple noncontiguous subdisks.
- The `rootvol` and `swapvol` volumes cannot be RAID5 volumes.
- Do not configure the `swapvol` volume with logging. There is no need for logging and resynchronization of mirrored swap volumes because swap data is not reused across system reboots.
- The names `rootvol` and `swapvol` are automatically assigned to the root volume and swap volume. Do not change these names.

  If `rootvol` and `swapvol` exist, then the encapsulation procedure creates a `rootvol` and `swapvol` with a `-02` suffix. You cannot use the suffix to differentiate between booted and mirrored plexes because LSM assigns plex suffixes in ascending order beginning with `-01` and reuses available numbers. Therefore, note the name of the plex from which the system boots in the event that you need to disassociate the plex.
- The root and swap volumes have the following specific minor device numbers, which should not be changed:
  - `rootvol` is minor device 0
  - `swapvol` is minor device 1

LSM will configure the boot disk for mirroring; however, be aware that the `rootvol` and `swapvol` volumes differ from other volumes in that they have the following LSM volume restrictions:

- You must have two unused partitions, which cannot be the `a` or `c` partitions.
- You must encapsulate the root file system and primary swap partitions at the same time.

4.2 Encapsulating the Boot Disk

To encapsulate the boot disk, you can:

- Select the option to encapsulate the boot disk directly into an LSM volume during the full installation process. See the *Installation Guide* for more information.
- Manually encapsulate the boot disk into LSM. See Section 4.2.1 for more information.

As part of the encapsulation process the system must be rebooted, and the following files are changed:

- The `/etc/fstab` (UFS root file system) is changed to use LSM volumes instead of disk partitions.
- The `/etc/sysconfigtab` is changed to update the `swapdevice` entry.
- If you are using AdvFS, the `/etc/fdmns/*` directory is updated to change domain directories that have disk partitions associated with the root disk.

4.2.1 Manually Encapsulating the Boot Disk

You can manually encapsulate the boot disk if you did not encapsulate it during the full installation. The steps to encapsulate the boot disk are the same whether you are using the UFS or the AdvFS. You should encapsulate the entire boot disk.

Follow these steps to encapsulate the entire boot disk:

1. Enter the `disklabel` command to verify that there is at least one free partition on the boot disk. The LSM encapsulation process requires one free partition-table entry to store LSM disk label tags. Note that the encapsulation procedure requires only the partition-table entry; it does not need the disk space associated with the partition. LSM uses space from the swap partition to create an LSM private region for the boot disk. After the encapsulation process completes, the swap partition is smaller by the size of the private region (by default 4096 sectors).
2. Enter the `volencap` command to create the LSM command scripts that convert all the partitions on the boot disk (including the root file system, `swap`, `usr` and `var` partitions) to LSM volumes. Specify the

boot device with the `volencap` command instead of individual partitions so that the entire boot disk is encapsulated. For example:

```
# volencap dsk1
```

Output similar to the following is displayed:

```
Setting up encapsulation for dsk1.

    - Creating simple disk dsk1h for config area (privlen=4096)
      Warning: space taken from -> dsk1b dsk1h
    - Creating nopriv disk dsk1a for rootvol
    - Creating nopriv disk dsk1b for swapvol
    - Creating nopriv disk dsk1g

The following disks are queued up for encapsulation or use by LSM:
dsk1h dsk1a dsk1b dsk1g

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

3. Enter the `volreconfig` command to encapsulate the boot disk, which also shuts down the system. For example:

```
# volreconfig
```

Output similar to the following is displayed:

```
The system will need to be rebooted in order to continue with
LSM volume encapsulation of:
 dsk1h dsk1a dsk1b dsk1g

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]
```

4. Boot the system.

4.2.2 Mirroring the Boot Disk

After you encapsulate the boot disk, you can mirror it. Mirroring, which is typically used to mirror user data, cannot access some of the data that is required for a system to boot, such as the boot track. Boot disk mirroring is restricted such that the plexes on the mirrored disk must be accessible as root and swap partitions.

Mirror the entire original boot disk onto another disk. Having one contiguous mirror that contains the volumes for boot disk partitions makes it easier to convert from volumes back to partitions. Choose a target disk for the mirror that:

- Is as large as the total amount of space in use on the boot disk.
- Is uninitialized (not under LSM control) .
- Has a disk label with all the partitions marked `unused`. Enter the `disklabel` command to display, and if necessary, reinitialize the disk label. See the `disklabel`(8) reference page for more information on the `disklabel` command.

If you are mirroring only the root file system and swap partitions, the target mirror disk must be at least as large as the sum of the sizes of the root file system and swap partitions on the original boot disk, plus the length of the private region.

You can use a disk that is the same as the original. For example you can use an RZ1BB to mirror another RZ1BB, or you can use a disk with a physical geometry different from the original disk. For example, an RZ1CD can mirror an RZ1BB, because the RZ1CD is larger than the RZ1BB.

Enter the `volrootmir -a` command to create mirrored volumes for all the partitions on the boot disk, including the root file system, swap, `usr` and `var` partitions. For example, to create mirrored volumes for all the partitions on the boot disk onto a disk called `dsk2`, enter:

```
# volrootmir -a dsk2
```

4.3 Unencapsulating the Boot Disk

You can unencapsulate the boot disk to revert volumes on the boot disk (`rootvol`, `swapvol`, `usrvol`, and `varvol`) back to partitions.

Unencapsulating the boot disk requires that you reboot the system using the disk that was last used for the `rootvol` and `swapvol` volumes (which might require that you first change the default boot device on the system console) and changes the following files:

- The `/etc/fstab` (UFS root filesystem) is changed to use disk partitions instead of LSM volumes.
- The `/etc/sysconfigtab` is changed to update the `swapdevice` entry.
- If you are using AdvFS, the `/etc/fdmns/*` directory is updated to change domain directories that have disk partitions associated with the root disk.

Note

To unencapsulate the `rootvol` and `swapvol` volumes, you must be sure that they have only one plex, and therefore are not mirrored.

Follow these steps to unencapsulate the boot disk volumes:

1. Remove the secondary plexes for volumes related to the boot disk. For example, to remove the secondary plexes for the `rootvol`, `swapvol`, `usrvol` and `varvol` volumes, enter:

```
# volplex -o rm dis rootvol-02
# volplex -o rm dis swapvol-02
# volplex -o rm dis usrvol-02
# volplex -o rm dis varvol-02
```

2. Revert all the LSM volumes back to partitions on the boot disk by entering the following command:

   ```
   # volunroot -a
   ```

3. Reboot the system using the disk that was last used for the `rootvol` volume.

4.4 Solving Boot Disk Encapsulation Problems

There are two recovery procedures if problems occur during the boot disk encapsulation process:

- If problems arise during the encapsulation procedure, you might need to manually reset the changes that were made by the encapsulation procedure.
- If booting to multiuser mode is impossible after boot disk encapsulation has succeeded, you can allow booting from the physical disk partition.

4.4.1 Resetting Changes Made During the Encapsulation Procedure

If problems occur during the encapsulation procedure, the encapsulation procedure tries to back out all changes made, and restore the use of partitions for the root file system. Under some circumstances, you might need to manually reset the changes made as a result of encapsulating the boot disk.

Follow these steps to manually reset the changes made during the encapsulation procedure:

1. Halt the machine.
2. At the console prompt, boot the machine in interactive mode. For example:

   ```
   >>> boot -fl i
   ```

3. When prompted for the kernel file name, enter:

   ```
   /vmunix lsm_rootdev_is_volume=2
   ```

4. When the system enters single-user mode, mount the root file system partition. For example:

   ```
   # mount -u /
   ```

5. If the root file system is:

- UFS, edit the /etc/fstab file as follows:
  - Change the device-special file from /dev/vol/rootdg/rootvol to the a partition of the boot disk.
  - Change the /dev/vol/rootdg/usrvol to the g partition of the boot disk.
  - If the var file system is on a separate partition, change the fstab from /dev/vol/rootdg/varvol to the h partition of the boot disk.
- AdvFS, enter:

  ```
  # cd /etc/fdmns/root_domain
  # rm rootvol
  # ln -s /dev/dskxa dskxa
  ```

  For the usr file system, enter:

  ```
  # cd /etc/fdmns/usr_domain
  # rm usrvol
  # ln -s /dev/dskxg dskxg
  ```

  If the var file system is separate from usr file system, enter:

  ```
  # cd /etc/fdmns/var_domain
  # rm varvol
  # ln -s /dev/dskxh dskxh
  ```

6. Edit the /etc/sysconfigtab file and change the LSM entry from:

   ```
   lsm_rootdev_is_volume = 1
   ```

   to:

   ```
   lsm_rootdev_is_volume = 0
   ```

7. Configure the swap partition to no longer use the LSM volumes by updating the vm:swapdevice entry in the sysconfigtab file to not reference the LSM volumes.

   See the *System Administration* and the swapon(8) reference page for more information.

8. Remove files related to the conversion. For example:

   ```
   # rm -rf /etc/vol/reconfig.d/disk.d/*
   ```

9. Update the disk label to remove the LSMnopriv fstypes that were set up by the volencap command for the swap, usr, and var partitions. For example:

   If the swap partition is on the b partition, enter:

   ```
   # disklabel -sF dskxb swap
   ```

   If the usr file system is on the g partition, enter:

disklabel -sF dskxg (AdvFS/ufs)

If the var file system is on the h partition, enter:

disklabel -sF dskxh (AdvFS/ufs)

10. Reboot the system on the same boot disk. The system reboots using disk partitions.

4.4.2 Booting From a Disk Partition

If you problems occur while booting to multiuser mode, follow these steps to boot from the physical disk partition:

1. Use the instructions for unencapsulating the boot disk in Section 4.3.
2. After the system reboots, enter the volmend command to set the good plex in the rootvol volume to ACTIVE.
3. Undo the changes that the encapsulation procedure made as described in Section 4.4.1.
4. Reboot the system.

5

Configuring Disks and Volumes

This chapter describes how to use LSM commands to create LSM volumes. Creating LSM volumes typically involves:

- Checking that there is free disk space for the LSM volume and adding more disks or disk groups if necessary
- Creating LSM volumes
- Configuring LSM volumes for use

The tasks described in this chapter can also be accomplished by using:

- The Storage Administrator GUI. See Chapter 9 for more information on the Storage Administrator.
- The `voldiskadm` menu interface. See Appendix D for more information on the `voldiskadm` menu interface.
- The Visual Administrator GUI. See Appendix B for more information on the Visual Administrator

For more information on an LSM command, see the reference page that corresponds to its name. For example, for more information on the `volassist` command, enter:

```
# man volassist
```

5.1 Checking for Free Disk Space

Before you create a volume, check to see if any of the system's disks were initialized for use with the LSM software, and verify that there is enough free disk space within a disk group to create the volume.

5.1.1 Checking for Initialized Disks

To display a list of initialized disks, enter:

```
# voldisk list
```

Output similar to the following is displayed:

```
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       rootdg      online
dsk7         sliced    -          -           online
dsk8         sliced    dsk8       dg1         online
dsk9         sliced    dsk9       dg1         online
dsk10        sliced    -          -           unknown
dsk11        sliced    -          -           unknown
```

A value of `online` in the `STATUS` column indicates that a disk was initialized for use with the LSM software.

In the previous example:

- The disks called `dsk2` through `dsk6` were initialized for use with the LSM software and were added into the `rootdg` disk group.
- The disk called `dsk7` was initialized for use with the LSM software because its status is `online`, but it is not currently configured within any disk group.
- The disks called `dsk8` and `dsk9` were initialized for use with the LSM software and were added into the `dg1` disk group.
- The disks called `dsk0`, `dsk1`, `dsk10`, and `dsk11` were not initialized for use with the LSM software because their status is `unknown`.

5.1.2 Checking for Space in a Disk Group

To display how much free disk space is available in disk groups, enter:

 # voldg free

Output similar to the following is displayed:

```
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    -
rootdg       dsk6         dsk6         dsk6         0          4106368    -
dg1          dsk8         dsk8         dsk8         0          4106368    -
dg1          dsk9         dsk9         dsk9         0          4106368    -
```

The value in the `LENGTH` column displays the amount of free space on a disk.

To display detailed disk space information in a specific disk group, enter:

 # volassist [-g *disk_group*] help space

For example, to display detailed disk space information about the `rootdg` disk group, enter:

volassist help space

Output similar to the following is displayed:

```
Disk: dsk2 len=4106368 used=2097217 free=2009151 (48.93%)
Attributes:
  dm:dsk2 device:dsk2 da:dsk2
Free regions:
2097233,2009151

Disk: dsk3 len=4106368 used=2097152 free=2009216 (48.93%)
  Attributes:   dm:dsk3 device:dsk3 da:dsk3
Free regions:
  2097168,2009216

Disk: dsk4 len=4106368 used=0 free=4106368 (100.00%)
Attributes:
  dm:dsk4 device:dsk4 da:dsk4
Free regions:
  16,4106368

Disk: dsk5 len=4106368 used=0 free=4106368 (100.00%)
Attributes:
  dm:dsk5 device:dsk5 da:dsk5
Free regions:
  16,4106368

Disk: dsk6 len=4106368 used=0 free=4106368 (100.00%)
Attributes:
  dm:dsk6 device:dsk6 da:dsk6
Free regions:
  16,4106368

Disk sets:   da:dsk2 space=4106368 used=2097217 free=2009151 (48.93%)
da:dsk3 space=4106368 used=2097152 free=2009216 (48.93%)
da:dsk4 space=4106368 used=0 free=4106368 (100.00%)
da:dsk5 space=4106368 used=0 free=4106368 (100.00%)
da:dsk6 space=4106368 used=0 free=4106368 (100.00%)
device:dsk2 space=4106368 used=2097217 free=2009151 (48.93%)
device:dsk3 space=4106368 used=2097152 free=2009216 (48.93%)
device:dsk4 space=4106368 used=0 free=4106368 (100.00%)
device:dsk5 space=4106368 used=0 free=4106368 (100.00%)
device:dsk6 space=4106368 used=0 free=4106368 (100.00%)
dm:dsk2 space=4106368 used=2097217 free=2009151 (48.93%)
dm:dsk3 space=4106368 used=2097152 free=2009216 (48.93%)
dm:dsk4 space=4106368 used=0 free=4106368 (100.00%)
dm:dsk5 space=4106368 used=0 free=4106368 (100.00%)
dm:dsk6 space=4106368 used=0 free=4106368 (100.00%)
```

To display detailed information about the disks in the `dg1` disk group, enter:

volassist -g dg1 help space

Output similar to the following is displayed:

```
Disk: dsk8 len=4106368 used=0 free=4106368 (100.00%)
Attributes:
  dm:dsk8 device:dsk8 da:dsk8
Free regions:
```

```
  16,4106368

Disk: dsk9 len=4106368 used=0 free=4106368 (100.00%)
Attributes:
  dm:dsk9 device:dsk9 da:dsk9
Free regions:
  16,4106368

Disk sets:
da:dsk8 space=4106368 used=0 free=4106368 (100.00%)
da:dsk9 space=4106368 used=0 free=4106368 (100.00%)
device:dsk8 space=4106368 used=0 free=4106368 (100.00%)
device:dsk9 space=4106368 used=0 free=4106368 (100.00%)
dm:dsk8 space=4106368 used=0 free=4106368 (100.00%)
dm:dsk9 space=4106368 used=0 free=4106368 (100.00%)
```

After you determine whether or not there is available disk space to create a volume, it may be necessary to configure more disks for use with the LSM software before you can create a volume. See Section 5.2 if you need to configure more disks for use with the LSM software. See Section 5.3 if there is sufficient disk space to create a volume.

5.2 Configuring a Disk for LSM Use

If there is not enough disk space to create a volume, you must configure more disks for use with the LSM software, which involves:

1. Initializing a disk for use with the LSM software, which does the following:
   - Destroys existing data on a disk.
   - Updates the disk label.
   - Uses the default values described in Table 5–1 to configure a disk for use with the LSM software.
2. Adding the initialized disk to a disk group. You must place an initialized disk into a disk group for the LSM software to use it. You can add disks to the default disk group (`rootdg`), which by default is created during the LSM installation and always exists on a system running the LSM software, or you can create additional disk groups to organize your disks into logical sets. Each disk group that you create must:
   - Contain at least one disk that is `online` and does not belong to another disk group
   - Be assigned a unique name

Table 5–1 shows the default values for the options that are used when you initialize disks for use with the LSM software. These options specify the size and layout of the disk's private region, which contains the disk's identification information, an area for the disk group's configuration database, and other information used internally by the LSM software. The default values for these options are sufficient for most environments, and changing them is usually not necessary.

Table 5–1: Disk Options Default Values

Option	Specifies	Default Value
`privlen=length`	The length of the private area (used for LSM private data) to create on the disk.	4096 sectors
`publen=length`	The length of the public area to create on the disk.	The size of the disk minus the private area on the disk
`noconfig`	Whether or not to disable the setup of kernel logs and configuration databases on the disk. The size of the private area is not changed, but it will not contain the normal private data.	Disabled
`config`	Whether or not to enable the setup of kernel logs and configuration databases on the disk.	Enabled
`nconfig=number`	The number of configuration copies and log copies to be initialized on the disk.	1
`configlen=length`	The length in sections of each configuration copy.	The default value is calculated based on the value of the `nconfig` attribute
`loglen=length`	The length of each log copy.	The default value is calculated based on the values of the `nconfig` and `nlog` attributes

The following sections describe how to configure new disks for use with the LSM software by using either the `voldiskadd` interactive utility or the individual LSM commands.

See Chapter 9 for information on how to configure new disks using the Storage Administrator. See Appendix C for information on how to configure new disks using the `voldiskadm` menu interface.

5.2.1 Configuring a Disk Using the `voldiskadd` Command

You use the `voldiskadd` command to initialize an entire disk for use with the LSM software. The `voldiskadd` command prompts you for information about the disk, uses default information described in Table 5–1 to initialize the disk, and places the disk in a disk group that you specify. If the disk group does not exist, it is created.

If you do not want to use the default information described in Table 5–1 to initialize the disk, initialize the disk using the individual LSM commands described in Section 5.2.2.

To configure a disk for use with the LSM software using the `voldiskadd` command, enter:

```
# voldiskadd disk_name
```

For example, to configure a disk called `dsk9` as an LSM sliced disk, enter:

```
# voldiskadd dsk9
```

If you omit the device name on the command line, `voldiskadd` prompts you for it.

Output similar to the following is displayed. Notice in this output that the disk will be a member of the `dg1` disk group, which is created as a result of this procedure.

```
  Add or initialize disks

Menu: VolumeManager/Disk/AddDisks

  Here is the disk selected.

  dsk9

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

  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)

  The default disk name that will be assigned is:

  dg101

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

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

  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)

  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)

  Initializing device dsk9.

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

Goodbye.
```

Once a disk is configured for use with the LSM software, you can create volumes as described in Section 5.3.

5.2.2 Configuring a Disk LSM Using Individual Commands

To configure a disk for use with the LSM software by using individual commands, you enter:

- The `voldisksetup` command to initialize disks
- The `voldg` command to either add the initialized disk to an existing disk group or to create a new disk group

5.2.2.1 Initializing a Disk Using the `voldisksetup` Command

The `voldisksetup` command performs two functions:

- Updates the partition table in the disk's disk label. The disk must already have a disk label before using the `voldisksetup` command.
- Initializes the disk's LSM private region that contains the disk's identification information, an area for the disk group's configuration database, and other important information used by the LSM software.

To initialize a disk, enter:

\# **voldisksetup -i {*diskname* | *partition*} [*options*]**

By specifying a disk name with the `voldisksetup` command, the entire disk is initialized for use with the LSM software as a `sliced` disk.

Alternatively, specifying a disk partition initializes that partition for use with the LSM software as a `simple` disk. For ease of management and greater flexibility, configure the entire disk for use with the LSM software as a `sliced` disk whenever possible.

Table 5–1 lists the options for which you can change values when using the `voldisksetup` command; however, it is usually not necessary to change these values.

Follow these steps to configure an entire disk for use with the LSM software as a sliced disk:

1. Identify that the disk is not initialized for use with the LSM software by entering the following command:

   # **voldisk list**

   Output similar to the following is displayed:

   ```
   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       rootdg     online
   dsk7        sliced   -          -          online
   dsk8        sliced   dsk8       dg1        online
   dsk9        sliced   dsk9       dg1        online
   dsk10       sliced   -          -          unknown
   dsk11       sliced   -          -          unknown
   dsk12       sliced   -          -          unknown
   dsk13       sliced   -          -          unknown
   ```

   Disks not initialized for use with the LSM software display `unknown` in the `STATUS` column.

2. Once an uninitialized disk is identified, enter the `disklabel` command to verify that the disk is not being used. For example to verify that a disk called `dsk10` is not being used, enter:

   # **disklabel dsk10**

   Output similar to the following is displayed:

   ```
   # /dev/rdisk/dsk10c:
   type: SCSI
   disk: RZ1BB-CS
   label:
   flags: dynamic_geometry
   bytes/sector: 512
   sectors/track: 86
   tracks/cylinder: 16
   sectors/cylinder: 1376
   cylinders: 3045
   sectors/unit: 4110480
   rpm: 7228
   interleave: 1
   ```

```
trackskew: 40
cylinderskew: 80
headswitch: 0           # milliseconds
track-to-track seek: 0  # milliseconds
drivedata: 0

8 partitions:
#        size   offset   fstype [fsize bsize   cpg]  # NOTE: values not exact
a:   131072        0   unused      0     0         # (Cyl.    0 - 95*)
b:   262144   131072   unused      0     0         # (Cyl.   95*- 285*)
c:  4110480        0   unused      0     0         # (Cyl.    0 - 2987*)
d:        0        0   unused      0     0         # (Cyl.    0 - -1)
e:        0        0   unused      0     0         # (Cyl.    0 - -1)
f:        0        0   unused      0     0         # (Cyl.    0 - -1)
g:  1858632   393216   unused      0     0         # (Cyl.  285*- 1636*)
h:  1858632  2251848   unused      0     0         # (Cyl. 1636*- 2987*)
```

All the disk partition's `fstype` field should be listed as `unused`.

Note

Not all software that uses a disk partition updates the `fstype` field to something other than `unused`. Be sure to verify that the disk is really unused.

3. Initialize the disk for use with the LSM software by entering the following command:

   # **voldisksetup -i *disk_name***

   For example, to initialize a disk called `dsk10` for use with the LSM software, enter:

   # **voldisksetup -i dsk10**

4. Display the results.

   - Use the `disklabel` command to display how the disk label was updated. For example, to display the disk label for a disk called `dsk10`, enter:

     # **disklabel dsk10**

     Output similar to the following is displayed:

```
# /dev/rdisk/dsk10c:
type: SCSI
disk: RZ1BB-CS
label:
flags: dynamic_geometry
bytes/sector: 512
sectors/track: 86  tracks/cylinder: 16
sectors/cylinder: 1376
cylinders: 3045
sectors/unit: 4110480
rpm: 7228
interleave: 1
trackskew: 40
cylinderskew: 80
```

```
headswitch: 0           # milliseconds
track-to-track seek: 0  # milliseconds
drivedata: 0

8 partitions:
#        size   offset   fstype [fsize bsize cpg]  # NOTE: values not exact
  a:   131072        0   unused      0     0       # (Cyl.    0 - 95*)
  b:   262144   131072   unused      0     0       # (Cyl.   95*- 285*)
  c:  4110480        0   unused      0     0       # (Cyl.    0 - 2987*)
  d:        0        0   unused      0     0       # (Cyl.    0 - -1)
  e:        0        0   unused      0     0       # (Cyl.    0 - -1)
  f:        0        0   unused      0     0       # (Cyl.    0 - -1)
  g:  4106384        0  LSMpubl                    # (Cyl.    0 - 2984*)
  h:     4096  4106384  LSMpriv                    # (Cyl. 2984*- 2987*)
```

- Use the `voldisk list` *disk_name* command to display the disk values used within the disk's private region. For example, to display the disk values for a disk called `dsk10`, enter:

  \# **voldisk list dsk10**

```
Device:    dsk10
devicetag: dsk10
type:      sliced
hostid:
disk:      name= id=929462025.1171.wdt2
group:     name= id=
flags:     online ready autoimport
pubpaths:  block=/dev/disk/dsk10g char=/dev/rdisk/dsk10g
privpaths: block=/dev/disk/dsk10h char=/dev/rdisk/dsk10h
version:   2.1
iosize:    min=512 (bytes) max=32768 (blocks)
public:    slice=6 offset=16 len=4106368
private:   slice=7 offset=0 len=4096
update:    time=929462026 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 disabled
  config   priv    249-  3010[  2762]: copy=01 offset=000231 disabled
  log      priv   3011-  3463[   453]: copy=01 offset=000000 disabled
```

- Use the `voldisk list` command to verify that the status of the disk is `online`, but not part of a disk group. For example:

  \# **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         rootdg       online
dsk7         sliced    -            -            online
dsk8         sliced    dsk8         dg1          online
dsk9         sliced    dsk9         dg1          online
dsk10        sliced    -            -            online
dsk11        sliced    -            -            unknown
dsk12        sliced    -            -            unknown
dsk13        sliced    -            -            unknown
```

After the disk is initialized, you can add it to an existing disk group or you can create a new disk group. The following sections describe how to add an initialized disk to an existing disk group or how to create a new disk group.

5.2.2.2 Adding a Disk To a Disk Group

After a disk is initialized for use with the LSM software, you can add it into an existing disk group.

Follow these steps to add a disk to an existing disk group:

1. Identify initialized disks that do not belong to a disk group by entering the following command:

   ```
   # voldisk list
   ```

   Output similar to the following is displayed:

   ```
   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       rootdg      online
   dsk7        sliced    -          -           online
   dsk8        sliced    dsk8       dg1         online
   dsk9        sliced    dsk9       dg1         online
   dsk10       sliced    -          -           online
   dsk11       sliced    -          -           unknown
   dsk12       sliced    -          -           unknown
   dsk13       sliced    -          -           unknown
   ```

   Initialized disks that do not belong to a disk group have a STATUS of online and a blank GROUP entry, which is represented by a dash.

   In the previous output, disks called dsk7 and dsk10 are initialized and not part of a disk group because their status is online and the GROUP column is blank.

2. Display the disk groups by entering the following command:

   ```
   # voldg list
   ```

   Output similar to the following is displayed:

   ```
   NAME        STATE     ID
   rootdg      enabled   927328730.1026.wdt2
   dg1         enabled   929455995.1168.wdt2
   ```

   This output shows that the system has two disk groups: rootdg and dg1.

3. Add an initialized disk to a disk group by entering the following command:

   ```
   # voldg adddisk [-g disk_group] diskname
   ```

For example, to add the LSM sliced disk called `dsk7` to the `rootdg` disk group, enter:

```
# voldg adddisk dsk7
```

To add the LSM sliced disk called `dsk10` to a disk group called `dg1`, enter:

```
# voldg -g dg1 adddisk dsk10
```

After disks are added to a disk group, you can create volumes as described in Section 5.3.

5.2.2.3 Creating A Disk Group

While placing all the disks into the default disk group, `rootdg`, provides the greatest flexibility for creating and reconfiguring volumes, you may want to group disks together to create other disk groups.

You can use an initialized disk that is not in a disk group to create a disk group. The disks configured into a disk group provide the disk space that is used for creating volumes. LSM volumes can only use disks that are in the same disk group. Therefore, carefully decide how to group disks into disk groups.

Follow these steps to create a disk group:

1. Identify initialized disks that are not in a disk group by entering the following command:

   ```
   # voldisk list
   ```

   Output similar to the following is displayed:

   ```
   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       rootdg     online
   dsk7        sliced   -          -          online
   dsk8        sliced   dsk8       dg1        online
   dsk9        sliced   dsk9       dg1        online
   dsk10       sliced   -          -          online
   dsk11       sliced   -          -          unknown
   dsk12       sliced   -          -          unknown
   dsk13       sliced   -          -          unknown
   ```

   Initialized disks that not in a disk group have a `STATUS` of `online` and a blank `GROUP` entry, which is represented by a dash.

2. Create a disk group by entering the following command:

   ```
   # voldg init disk_group disk_name
   ```

For example, to create a disk group called dg2 using a disk called dsk10, enter:

```
# voldg init dg2 dsk10
```

3. By default, LSM maintains up to four copies of the LSM configuration database on different disks within the disk group. When a disk is added, removed, or fails, the LSM software automatically evaluates, and if necessary, changes the number of copies and location of the configuration databases for that disk group.

   To display the current number and locations of a disk group's configuration database, enter the following command:

   ```
   # voldg list disk_group
   ```

   For example, to display the current number and locations of configuration databases for a disk group called dg2, enter:

   ```
   # voldg list dg2
   ```

   Output similar to the following is displayed:

   ```
   Group:     dg2
   dgid:      929473041.1178.wdt2
   import-id: 0.1177
   flags:
   copies:    nconfig=default nlog=default
   config:    seqno=0.1027 permlen=2993 free=2991 templen=2 loglen=453
   config disk dsk10 copy 1 len=2993 state=clean online
   log disk dsk10 copy 1 len=453
   ```

4. Display the current LSM configuration for a disk group by entering the following command:

   ```
   # volprint [-g disk_group] -ht
   ```

   For example to display the current LSM configuration for a disk group called dg2, enter:

   ```
   # volprint -g dg2 -ht
   ```

   Output similar to the following is displayed:

   ```
   DG NAME         NCONFIG      NLOG     MINORS   GROUP-ID
   DM NAME         DEVICE       TYPE     PRIVLEN  PUBLEN   STATE
   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

   dg dg2          default      default  5000     929473041.1178.wdt2
   dm dsk10        dsk10        sliced   4096     4106368  -
   ```

Once a disk is initialized for use with the LSM software and in a disk group, you can create volumes as described in Section 5.3.

5.3 Creating A Volume

After disks are initialized and added into disk groups, you can create LSM volumes. Creating an LSM volume includes:

- Selecting the type of data layout for the volume. Data layout types are simple, concatenated, striped, mirrored, mirrored/striped, or RAID5.
- Deciding which volume usage type to use, for example the `fsgen` or `gen` type. The LSM volume should use the `fsgen` type if the volume will contain a file system; otherwise, the volume should use the `gen` usage type.

  If you encapsulated and mirrored the boot disk as described in Chapter 4, then the root volume has a usage type of `root` and the swap volumes has a usage type of `swap`. See Chapter 4 for more information on encapsulating and mirroring the root and swap
- Locating storage space to create the volume.
- Creating and associating a volume object with one or more plex objects.
- Associating subdisks to each of the volume's plexes.

You can create an LSM volume by using:

- The `volassist` command

  The `volassist` command provides an easy method for creating and changing volume configurations. The `volassist` command:

  – Finds space for and creates volumes

  – Use a set of default values for options, which you can change, to create a volume. To view the default values for options, enter:

    ```
    # volassist help showattrs
    ```

    Output similar to the following is displayed:

    ```
    #Attributes:
    layout=nomirror,nostripe,span,nocontig,raid5log,noregionlog,diskalign,nostorage
    mirrors=2 columns=0 nlogs=1 regionlogs=1 raid5logs=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=
    ```

  – Adds mirrors and logs to existing volumes

  – Provides for the migration of data from specified disks

  – Provides facilities for the online backup of existing volumes

- A series of individual LSM commands. Using individual commands to create volumes is for system administrators who require greater flexibility in defining an LSM volume configuration.
- The Storage Administrator
- The `voldiskadm` menu interface

The following sections describe how to create LSM volumes by using either the `volassist` command or the individual LSM commands.

See Chapter 9 for information on how to create volumes using the Storage Administrator.

5.3.1 Creating Simple and Concatenated Volumes

An LSM volume that maps the volume blocks directly to the disk blocks without mirroring, striping, or disk concatenation is often referred to as a simple volume.

Using a simple volume has minimal I/O performance impact and allows greater flexibility compared to using a disk partition without LSM because you can easily change the configuration online, such as moving the data to a less busy disk, adding a mirror, and so on.

A concatenated LSM volume is a volume that combines one or more sections of disk space. Usually these multiple disk sections, or subdisks, span multiple, different disks, but this is not required.

A concatenated volume can be used to combine several smaller disks to form a single, larger LSM volume.

5.3.1.1 Using the `volassist` Command

You can use the `volassist` command to create a simple volume on a disk by specifying the disk to be used and a volume size that is less than or equal to the available storage space on that disk. Specifying a volume size that exceeds an individual disk size will create a concatenated volume.

If you do not specify a disk name or multiple disks, the `volassist` command selects the disk location of the volume. However, a concatenated volume may be created that spans multiple disks if the volume will not fit on one disk.

Follow these steps to create a simple volume called `v1` in the `rootdg` disk group on a disk called `dsk2`:

1. Display the disk space on a disk by entering the following command:

   ```
   # volassist help space | grep disk_name
   ```

For example, to check the space on a disk called dsk2, enter:

```
# volassist help space | grep dsk2
```

Output similar to the following is displayed:

```
Disk: dsk2 len=4106368 used=0 free=4106368 (100.00%)
  dm:dsk2 device:dsk2 da:dsk2
  da:dsk2 space=4106368 used=0 free=4106368 (100.00%)
  device:dsk2 space=4106368 used=0 free=4106368 (100.00%)
  dm:dsk2 space=4106368 used=0 free=4106368 (100.00%)
```

2. Create the volume by entering the following command:

   ```
   # volassist [-g group_name] make volume_name\
    length [disk_name]
   ```

   For example, to create a simple volume called v1 in the rootdg disk group on a disk called dsk2, enter:

   ```
   # volassist make v1 4106368s dsk2
   ```

3. Display the results by entering the following command:

   ```
   # volprint -ht volume_name
   ```

   For example, to display information about a volume called v1, enter:

   ```
   # volprint -ht v1
   ```

   Output similar to the following is displayed:

```
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  v1           fsgen        ENABLED  ACTIVE   4106368  SELECT    -
pl v1-01        v1           ENABLED  ACTIVE   4106368  CONCAT    -        RW
sd dsk2-01      v1-01        dsk2     0        4106368  0         dsk2     ENA
```

Follow these steps create a concatenated volume:

1. Display the disk space in a disk group by entering the following command:

   ```
   # volassist [-g disk_group] help space
   ```

   For example, to display the disk space in a disk group called dg1, enter:

   ```
   # volassist -g dg1 help space
   ```

   Output similar to the following is displayed:

```
Disk: dsk8 len=4106368 used=0 free=4106368 (100.00%)
Attributes:
  dm:dsk8 device:dsk8 da:dsk8
Free regions:
  16,4106368

Disk: dsk9 len=4106368 used=0 free=4106368 (100.00%)
Attributes:
```

```
  dm:dsk9 device:dsk9 da:dsk9
Free regions:
  16,4106368

Disk sets:
  da:dsk8 space=4106368 used=0 free=4106368 (100.00%)
  da:dsk9 space=4106368 used=0 free=4106368 (100.00%)
  device:dsk8 space=4106368 used=0 free=4106368 (100.00%)
  device:dsk9 space=4106368 used=0 free=4106368 (100.00%)
  dm:dsk8 space=4106368 used=0 free=4106368 (100.00%)
  dm:dsk9 space=4106368 used=0 free=4106368 (100.00%)
```

2. Create the concatenated volume by entering the following command:

   **# volassist [-g *group_name*] -U *usage_type* make \
   volume_name *length* [*disk_names*]**

   For example, to create a 3 GB, concatenated volume called `v2` in the disk group called `dg1` on disks called `dsk8` and `dsk9`, enter:

   # **volassist -g dg1 -U gen make v2 3g dsk8 dsk9**

3. Display the results by entering the following command:

   # **volprint -ht *volume_name***

   For example, to display information about a volume called `v2`, enter:

   # **volprint -ht v2**

   Output similar to the following is displayed:

```
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  v2           gen          ENABLED  ACTIVE   6291456  SELECT    -
pl v2-01        v2           ENABLED  ACTIVE   6291456  CONCAT    -        RW
sd dsk8-01      v2-
01        dsk8     0        2185088  0        dsk8     ENA
sd dsk9-01      v2-01        dsk9     0        4106368  2185088   dsk9     ENA
```

5.3.1.2 Using Individual Commands

To create a simple volume called `v1` in the `rootdg` disk group on a disk called `dsk2` using the `volmake` and `volume` commands, enter:

volmake sd dsk2-01 dsk2,0,4106368
volmake plex v1-01 sd=dsk2-01
volmake -U fsgen vol v1 plex=v1-01
volume start v1

Display the results by entering the following command:

volprint -ht *volume_name*

For example, to display information about a volume called `v1`, enter:

volprint -ht v1

Output similar to the following is displayed:

```
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  v1           fsgen        ENABLED  ACTIVE   4106368  SELECT    -
pl v1-01        v1           ENABLED  ACTIVE   4106368  CONCAT    -        RW
sd dsk2-01      v1-01        dsk2     0        4106368  0         dsk2     ENA
```

5.3.2 Creating A Striped Volume

Using LSM striped volumes (RAID0) is a common and effective way to dramatically increasing I/O performance. The actual performance gained by striping depends on numerous factors such as:

- The number of disks within the stripe set
- The location of the disks
- How users and applications perform I/O
- The stripe width

The I/O performance can improve and scale linearly by the same number of disks used within a stripe-set. For example, striping a volume's data across two disks can potentially improve both read and write performance for that volume by a factor of 2, and striping data across four disks can potentially improve performance up to a factor of 4.

LSM striped volumes can also improve performance by eliminating one bus or controller from becoming the bottleneck for the volume's I/O. By using multiple disks that reside on multiple buses to form the stripe set, a greater I/O throughput can be achieved for a single volume than would be otherwise possible if the volume's data resided on the same I/O bus. Therefore, understanding the system's hardware I/O topology when selecting which disks to use when configuring a striped volume will help to significantly improve I/O performance and avoid bottlenecks.

The default stripe width of 64KB usually works best for most I/O workloads, such as file systems and databases that generate multiple I/O to the same volume. For highly specialized environments where very large, raw I/O will always be performed to a volume one I/O at a time (for example multiple I/O is never issued to the same volume at the same time), a different stripe width may provide better performance, which enables the larger data transfer to be split up and performed in parallel. The best stripe width size to use for single, large I/O environments depends on:

- Whether the I/O size varies

- The number of disks within the stripe set
- The hardware configuration, such as whether multiple buses are used
- The hardware performance, such as average disk seek and transfer times

It is best to experiment with different stripe widths sizes to determine the size that works best for these specialized I/O environments.

Using LSM's online support can help when configuring and deconfiguring different plexes with different stripe width sizes for comparing what works best for your actual I/O workload.

5.3.2.1 Using the `volassist` Command

Follow these steps to create a striped volume using the `volassist` command:

1. Determine which disks are configured for use with the LSM software by entering the following command:

   \# **volprint -g *disk_group* -dt**

   For example, to display the LSM disks in the `rootdg` disk group, enter:

   \# **volprint -g rootdg -dt**

   Output similar to the following is displayed:

```
DM NAME         DEVICE       TYPE     PRIVLEN  PUBLEN   STATE
dm dsk2         dsk2         sliced   4096     4106368  -
dm dsk3         dsk3         sliced   4096     4106368  -
dm dsk4         dsk4         sliced   4096     4106368  -
dm dsk5         dsk5         sliced   4096     4106368  -
dm dsk6         dsk6         sliced   4096     4106368  -
dm dsk7         dsk7         sliced   4096     4106368  -
```

2. Create the striped volume by entering the following command:

   \# **volassist [-g *disk_group*] make *volume_name* *length* **
   nstripe=*n* [*options*]

   Where *n* is the number of columns to be configured in the stripe set.

   For example, to create the striped volume called `v_stripe` with the default stripe width on disks `dsk2` through `dsk7`, enter:

   \# **volassist make v_stripe 6g nstripe=6 dsk2 dsk4 dsk6 **
   dsk3 dsk5 dsk7

3. Display the results by entering the following command:

   \# **volprint -ht *volume_name***

   For example, to display information about a volume called `v_stripe`, enter:

volprint -ht v_stripe

Output similar to the following is displayed:

```
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  v_stripe     fsgen        ENABLED  ACTIVE   12582912 SELECT    v_stripe-01
pl v_stripe-01  v_stripe     ENABLED  ACTIVE   12582912 STRIPE    6/128    RW
sd dsk2-01      v_stripe-01  dsk2     0        2097152  0/0       dsk2     ENA
sd dsk3-01      v_stripe-01  dsk3     0        2097152  1/0       dsk3     ENA
sd dsk4-01      v_stripe-01  dsk4     0        2097152  2/0       dsk4     ENA
sd dsk5-01      v_stripe-01  dsk5     0        2097152  3/0       dsk5     ENA
sd dsk6-01      v_stripe-01  dsk6     0        2097152  4/0       dsk6     ENA
sd dsk7-01      v_stripe-01  dsk7     0        2097152  5/0       dsk7     ENA
```

5.3.2.2 Using Individual Command

For control over how the volume is configured, use the `volmake` and `volume` commands. Using these commands to create a stripe volume provides you with greater control in specifying which disks will be used for which stripe column. In this way, you can obtain the best performance by configuring the striped plex so the stripe columns alternate or rotate across different hardware buses.

Follow these steps to use the `volmake` and `volume` commands to create an LSM stripe volume:

1. Determine which hardware bus on which each LSM disk resides on by entering the following command:

   # **file /dev/rdisk/*disk_name***

   For example, to determine the hardware bus for disks called `dsk2c`, `dsk3c`, `dsk4c`, `dsk5c`, `dsk6c`, `dsk7c`, enter:

   # **file /dev/rdisk/dsk2c /dev/rdisk/dsk3c \
   /dev/rdisk/dsk4c /dev/rdisk/dsk5c \
   /dev/rdisk/dsk6c /dev/rdisk/dsk7c**

   Output similar to the following is displayed:

```
/dev/rdisk/dsk2c:  character special (19/70) SCSI #1 RZ1BB-CS ( disk #3 (SCSI ID #4)
(SCSI LUN #0)
/dev/rdisk/dsk3c:  character special (19/86) SCSI #1 RZ1BB-CS ( disk #4 (SCSI ID #6)
(SCSI LUN #0)
/dev/rdisk/dsk4c:  character special (19/102) SCSI #2 RZ1BB-CS ( disk #5 (SCSI ID #1)
(SCSI LUN #0)
/dev/rdisk/dsk5c:  character special (19/118) SCSI #2 RZ1BB-CS ( disk #6 (SCSI ID #5)
(SCSI LUN #0)
/dev/rdisk/dsk6c:  character special (19/134) SCSI #3 RZ1BB-CS ( disk #7 (SCSI ID #0)
(SCSI LUN #0)
/dev/rdisk/dsk7c:  character special (19/150) SCSI #3 RZ1BB-CS ( disk #0 (SCSI ID #2)
(SCSI LUN #0)
```

2. Create the subdisks by entering the following commands:

   # **volmake sd *sub_disk_name disk_name, length***

For example, to create subdisks called dsk2-01, dsk3-01, dsk4-01, dsk5-01, dsk6-01, dsk7-01, enter:

```
# volmake sd dsk2-01 dsk2,0,2097152
# volmake sd dsk3-01 dsk3,0,2097152
# volmake sd dsk4-01 dsk4,0,2097152
# volmake sd dsk5-01 dsk5,0,2097152
# volmake sd dsk6-01 dsk6,0,2097152
# volmake sd dsk7-01 dsk7,0,2097152
```

3. Create a striped plex by entering the following command:

```
# volmake plex plex_name layout=stripe st_width=64k \
 sd=sub_disk_names
```

For example, to create a plex called v_stripe-01 using subdisks called dsk2-01, dsk3-01, dsk4-01, dsk5-01, dsk6-01, dsk7-01, enter:

```
# volmake plex v_stripe-01 layout=stripe st_width=64k \
 sd=dsk2-01,dsk4-01,dsk6-01,dsk3-01,dsk5-01,dsk7-01
```

Notice the order of the subdisks specified when creating the plex will rotate the stripe columns across different hardware buses.

4. Create the volume using the striped plex by entering the following command:

```
# volmake -U usage_type vol volume_name plex=plex_name
```

For example, to use a plex called v_stripe-01 to create a volume called v_stripe, enter:

```
# volmake -U gen vol v_stripe plex=v_stripe-01
```

5. Start the volume by entering the following command:

```
# volume start volume_name
```

For example, to start a volume called v_stripe, enter:

```
# volume start v_stripe
```

6. Display the results by entering the following command:

```
# volprint -ht volume_name
```

For example, to display information about a volume called v_stripe, enter:

```
# volprint -ht v_stripe
```

Output similar to the following is displayed:

```
#  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  v_stripe     gen          ENABLED  ACTIVE   12582912 ROUND     -
pl v_stripe-01  v_stripe     ENABLED  ACTIVE   12582912 STRIPE    6/128    RW
sd dsk2-01      v_stripe-01  dsk2     0        2097152  0/0       dsk2     ENA
sd dsk4-01      v_stripe-01  dsk4     0        2097152  1/0       dsk4     ENA
sd dsk6-01      v_stripe-01  dsk6     0        2097152  2/0       dsk6     ENA
sd dsk3-01      v_stripe-01  dsk3     0        2097152  3/0       dsk3     ENA
sd dsk5-01      v_stripe-01  dsk5     0        2097152  4/0       dsk5     ENA
sd dsk7-01      v_stripe-01  dsk7     0        2097152  5/0       dsk7     ENA
```

5.3.3 Creating a Mirrored Volume

Using LSM mirrored volumes (RAID1) is a common and effective way to improve data availability. If one disk fails on a mirrored volume, the data can still be accessed from the other copy, or plex. By mirroring data using disks connected to different controllers or buses, you can improve data availability even further because the data is still accessible if a controller, cable, or storage cabinet fails. Therefore, it is helpful to understand a system's I/O hardware topology; that is, knowing which disk reside on which I/O bus.

Besides improving data availability, mirroring significantly improves read performance because multiple reads to the same volume are simultaneously done by using the multiple copies of data. For example, read performance can potentially improve by a factor of two on a mirrored volume with two plexes because twice as many reads are performed done at the same time.

Writes to the volume result in multiple, simultaneous write requests to each plex, so the time it takes to write to a volume may be slightly longer because of slight performance deviations between individual disks. For example, an individual write might take an additional 5 percent on average to complete because the volume write must wait for both writes to complete on both plexes (disks).

You can improve overall I/O performance with mirroring because the larger performance gains for read often more than offset the slight degradation for writes. Comparing the number of read operations to the number of write operations on a volume using the `volstat` command can help give you better insight into whether mirroring can also help improve overall performance as well as provide higher data availability.

Because the LSM software allows you to change a volume (add or remove a mirror) on line, you can measure the overall performance implications on the actual I/O workload without stopping or disrupting service to a volume.

Mirrored volumes created with the `volassist` command will have dirty region logging (DRL) enabled by default. A DRL is used with mirrored volumes to track used (or dirty) regions within the mirrored volume. While DRL may add slight overhead to writes to the mirrored volume, the DRL significantly reduces the amount of time that it takes to resynchronize a mirrored volume when the system boots after a failure because only the dirty regions within the volume are resynchronized rather than the entire volume.

While using a DRL with a mirrored volume is not required and has no affect on data integrity, a DRL dramatically reduces the amount of time it takes to resynchronize a mirrored volume. It is recommended that you configure a mirrored volume with a DRL, which is the default.

Note

In a TruCluster environment, the resynchronization overhead and time is significantly high. You should always configure a mirrored volume with a DRL in a TruCluster environment.

5.3.3.1 Using the `volassist` Command

Follow these steps to create a mirrored volume:

1. Determine which disks are configured for use with the LSM software by entering the following command:

   ```
   # volprint -g disk_group -dt
   ```

   For example, to display the LSM disks in the `dg1` disk group, enter:

   ```
   # volprint -g dg1 -dt
   ```

   Output similar to the following is displayed:

   ```
   TY NAME         ASSOC        KSTATE    LENGTH    PLOFFS    STATE     TUTIL0   PUTIL0
   dm dsk8         dsk8         -         4106368   -         -         -        -
   dm dsk9         dsk9         -         4106368   -         -         -        -
   dm dsk10        dsk10        -         4106368   -         -         -        -
   ```

   By default, `volassist` creates a DRL, so additional storage space is needed when creating the volume. Also, the plex layout is concatenated by default.

2. Create a mirrored volume by entering the following command:

   ```
   # volassist [-g disk_group] make volume_name \
   length nmirror=2 [disk_names]
   ```

   For example, to create a mirrored volume in the `dg1` disk group using disks called `dsk8`, `dsk9`, and `dsk10`, enter:

   ```
   # volassist -g dg1 make v_mirr01 4106368s nmirror=2 \
   dsk8 dsk9 dsk10
   ```

   If you do not specify a disk name, or if you specify more than three disks, the `volassist` command selects the disk location of the volume.

3. Display the results by entering the following command:

   # **volprint -ht *volume_name***

   For example, to display information about a volume called v_mirr01, enter:

   # **volprint -ht v_mirr01**

   Output similar to the following is displayed:

```
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  v_mirr01     fsgen        ENABLED  ACTIVE   4106368  SELECT    -
pl v_mirr01-01  v_mirr01     ENABLED  ACTIVE   4106368  CONCAT    -        RW
sd dsk8-01      v_mirr01-01  dsk8     0        4106368  0         dsk8     ENA
pl v_mirr01-02  v_mirr01     ENABLED  ACTIVE   4106368  CONCAT    -        RW
sd dsk9-01      v_mirr01-02  dsk9     0        4106368  0         dsk9     ENA
pl v_mirr01-03  v_mirr01     ENABLED  ACTIVE   LOGONLY  CONCAT    -        RW
sd dsk10-01     v_mirr01-03  dsk10    0        130      LOG       dsk10    ENA
```

Notice in this output that the total volume size is 4106368 sectors, and two data plexes called v_mirr01-01 and v_mirr01-02 of the same size were created, which use disks called dsk8 and dsk9 respectively. Also notice the DRL plex is called v_mirr01-03 and uses a disk called dsk10.

To maintain greater control over which disk will contain the volume's data and which disk is used for the volume's DRL, you may want to first create the volume without a DRL, then add the log separately. For example, to create the same volume as the previous example, but explicitly specifying the disks for data and the disk for a DRL, enter the following commands:

```
# volassist -g dg1 make v_mirr01 4106368s nmirror=2 \
 layout=nolog dsk8 dsk9
```

```
# volassist -g dg1 addlog v_mirr01 dsk10
```

5.3.3.2 Using Individual Commands

For complete control over how the volume is configured, use the volmake and volume commands. For example, to create a volume with two plexes called v_mirr01-01 and v_mirr01-02 that use disks called dsk8 and dsk9 respectively, and create a DRL plex called v_mirr01-03 that uses a disk called dsk10, enter:

```
# volmake -g dg1 sd dsk8-01 dsk8,0,4106368
# volmake -g dg1 sd dsk9-01 dsk9,0,4106368
# volmake -g dg1 sd dsk10-01 dsk10,0,130
# volmake -g dg1 plex v_mirr01-01 sd=dsk8-01
# volmake -g dg1 plex v_mirr01-02 sd=dsk9-01
# volmake -g dg1 plex v_mirr01-03 logsd=dsk10-01
#  volmake -g dg1 -Ufsgen vol v_mirr01 \
```

```
plex=v_mirr01-01,v_mirr01-02,v_mirr01-03
# volume start v_mirr01
```

In this example, notice the mirrored volume's DRL size is 130 blocks and was placed on a different disk than the volume's data.

The following section provides more information on sizing and placing a mirrored volume's DRL.

5.3.3.3 Creating a DRL for a Mirrored Volume

When creating a mirrored volume using the `volassist` command, a DRL is created by default. This section provides additional information on sizing and placing a mirrored volume's DRL for best results.

Follow these guidelines to create a DRL:

- The volume must be mirrored.
- Avoid placing the log on a heavily-used disk.
- Avoid using the same disk for both the volume's data and log.
- Use disks within a storage subsystem configured with a nonvolatile write-back cache, if available.
- At least one log subdisk must exist on the volume. However, only one log subdisk can exist per plex.
- Although you can associate a logging subdisk to a plex that also contains data, it is best to configure a logging subdisk to plexes that do not contain data, for example a separate or log only plex.
- It is possible to mirror log subdisks by having more than one log subdisk (but only one per plex) in the volume. This ensures that logging can continue, even if a disk failure causes one log subdisk to become inaccessible.
- The minimum DRL size for a TruCluster environment is 65 blocks. The `volassist` command creates a DRL sized for a TruCluster environment even on non-TruCluster system to ensure a smooth migration to a TruCluster environment in the future.

  Table 5–2 shows example optimum DRL sizes for TruCluster configurations.

Table 5–2: DRL Sizes for TruCluster Configurations

Volume Size in GB	DRL Size in Blocks
1 or smaller	65
2	132
3	132
4	198
5	198
60	2046
61	2046
62 or larger	2122

See *Cluster Administration* for information about configuring LSM in a TruCluster environment.

- The minimum DRL size for a non-TruCluster environment is 2 blocks.

  For systems not configured as part of a TruCluster environment, you must configure a log subdisk with 2 or more blocks, preferably an even number, because the last block in a log subdisk with an odd number of blocks is not used. The log subdisk size is normally proportional to the volume size. If a volume is less than 2 GB, a log subdisk of 2 blocks is sufficient. Increase the log subdisk by 2 blocks for each additional 2 GB of volume size. To facilitate later migration to a TruCluster environment, you should use the TruCluster DRL sizes in Table 5–2.

  By default, the `volassist` command configures a larger log subdisk so the mirrored volume with the log can be used within a TruCluster.

  Table 5–3 shows example optimum DRL sizes for non-TruCluster systems.

Table 5–3: DRL Sizes for Non-TruCluster Configurations

Volume Size in GB	DRL Size in Blocks
1 or smaller	2
2	4
3	4
4	6
5	6
60	62

Table 5–3: DRL Sizes for Non-TruCluster Configurations (cont.)

Volume Size in GB	DRL Size in Blocks
61	62
62 or larger	64

By default, a log plex is created to contain the log subdisk. Once created, the plex containing a log subdisk is treated as a regular plex. You can remove the log plex and subdisk using the same procedures to remove regular plexes and subdisks.

To use the `volassist` command to create a DRL for a mirrored volume, enter:

```
# volassist [-g disk_group] addlog volume_name \
 [disk_name]
```

For example, to create a DRL for a volume called `volmir`, enter:

```
# volassist addlog volmir
```

To use the `volmake` and `volplex` commands to create a DRL for a volume called `volmir`, enter:

```
# volmake sd dsk10-01 dsk10,0,130
# volmake plex volmir-03 logsd=dsk10-01
# volplex att volmir volmir-03
```

Note

Do not configure a DRL for mirrored volumes that are used for swap.

5.3.4 Creating A Mirrored and Striped Volume

Configuring a LSM volume to be both mirrored and striped is a common and effective way to improve both performance and availability for a volume. This is accomplished with the LSM software by configuring each of the volume's data plexes, or mirrors, to have a stripe layout. Just as when creating either a striped or mirrored volume, understanding the system's I/O hardware topology (for example, which disks are on which buses) is useful for maximizing performance and availability by using disks that reside on different buses.

It may not always be practical to both mirror and stripe across buses (for example, have each disk on its own I/O bus). Mirroring across I/O buses is preferred over striping because this provides both the highest level of availability and ensures all the volume's reads and writes are evenly distributed across the buses for the best performance.

5.3.4.1 Using the `volassist` Command

Follow these steps to create a mirrored and striped volume:

1. Create a mirrored and striped volume by entering the following command:

   **# volassist [-g *disk_group*] make *volume_name* nstripe=*n* **
   nmirror=*m* [*options*]

   In this command, *n* is the number of columns to be used and *m* is the number of plexes.

   For example, to create a 3GB, mirrored and striped volume called `vol4` with the default stripe width using any of the disks in the `rootdg` disk group, enter:

   # **volassist make vol4 3g nmirror=2 nstripe=3**

2. Display the results by entering the following command:

   # **volprint -ht *volume_name***

   For example, to display information about a volume called `v_mirr01`, enter:

   # **volprint -ht vol4**

   Output similar to the following is displayed:

```
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  vol4         fsgen        ENABLED  ACTIVE   6291456  SELECT    -
pl vol4-01      vol4         ENABLED  ACTIVE   6291456  STRIPE    3/128    RW
sd dsk2-01      vol4-01      dsk2     130      2097152  0/0       dsk2     ENA
sd dsk3-01      vol4-01      dsk3     0        2097152  1/0       dsk3     ENA
sd dsk4-01      vol4-01      dsk4     0        2097152  2/0       dsk4     ENA
pl vol4-02      vol4         ENABLED  ACTIVE   6291456  STRIPE    3/128    RW
sd dsk5-01      vol4-02      dsk5     0        2097152  0/0       dsk5     ENA
sd dsk6-01      vol4-02      dsk6     0        2097152  1/0       dsk6     ENA
sd dsk7-01      vol4-02      dsk7     0        2097152  2/0       dsk7     ENA
pl vol4-03      vol4         ENABLED  ACTIVE   LOGONLY  CONCAT    -        RW
sd dsk2-02      vol4-03      dsk2     0        130      LOG       dsk2     ENA
```

The `volassist` command selects which disks to use, which may not be the optimum configuration.

Follow these steps to select the disks are used:

1. Check the I/O hardware topology for the disks to be used. For example:

   # **file /dev/rdisk/dsk6c /dev/rdisk/dsk7c **
   **/dev/rdisk/dsk8c /dev/rdisk/dsk9c **

/dev/rdisk/dsk10c /dev/rdisk/dsk11c /dev/rdisk/dsk12c

Output similar to the following is displayed:

```
/dev/rdisk/dsk6c:  character special (19/134) SCSI #3 RZ1BB-CS ( disk #7 (SCSI ID #0)
(SCSI LUN #0)
/dev/rdisk/dsk7c:  character special (19/150) SCSI #3 RZ1BB-CS ( disk #0 (SCSI ID #2)
(SCSI LUN #0)
/dev/rdisk/dsk8c:  character special (19/166) SCSI #3 RZ1BB-CS ( disk #1 (SCSI ID #4)
(SCSI LUN #0)
/dev/rdisk/dsk9c:  character special (19/182) SCSI #3 RZ1BB-CS ( disk #2 (SCSI ID #6)
(SCSI LUN #0)
/dev/rdisk/dsk10c: character special (19/198) SCSI #4 RZ1BB-CS ( disk #3 (SCSI ID #1)
(SCSI LUN #0)
/dev/rdisk/dsk11c: character special (19/214) SCSI #4 RZ1BB-CS ( disk #4 (SCSI ID #3)
(SCSI LUN #0)
/dev/rdisk/dsk12c: character special (19/230) SCSI #4 RZ1BB-CS ( disk #5 (SCSI ID #5)
(SCSI LUN #0)
```

2. Create a striped volume using disks on the same I/O bus (as shown in the previous output) by entering the following command:

   **# volassist make *volume_name length* **
   nstripe=*number disks*

   For example, to create a 3GB 3–way striped volume called `vol4` using disks called `dsk10`, `dsk11`, and `dsk12`, enter:

   # **volassist make vol4 3g nstripe=3 dsk10 dsk11 dsk12**

3. Add a 3-way striped plex (mirror) so that the volume's data is mirrored across SCSI buses by entering the following command:

   # **volassist mirror *volume_name* nstripe=*number disks***

   For example to mirror a volume called `vol4` using disks called `dsk6`, `dsk7`, and `dsk8`, enter:

   # **volassist mirror vol4 nstripe=3 dsk6 dsk7 dsk8**

4. Add a DRL on a separate disk by entering the following command:

   # **volassist addlog *volume_name disk_name***

   For example, to create a DRL on a disk called `dsk9` for a volume called `vol4`, enter:

   # **volassist addlog vol4 dsk9**

5. Display the results by entering the following command:

   # **volprint -ht *volume_name***

   For example, to display information for a volume called `vol4`, enter:

   # **volprint -ht vol4**

   Output similar to the following is displayed:

```
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  vol4         fsgen        ENABLED  ACTIVE   6291456  SELECT    -
pl vol4-01      vol4         ENABLED  ACTIVE   6291456  STRIPE    3/128    RW
sd dsk10-01     vol4-01      dsk10    0        2097152  0/0       dsk10    ENA
sd dsk11-01     vol4-01      dsk11    0        2097152  1/0       dsk11    ENA
sd dsk12-01     vol4-01      dsk12    0        2097152  2/0       dsk12    ENA
pl vol4-02      vol4         ENABLED  ACTIVE   6291456  STRIPE    3/128    RW
sd dsk6-01      vol4-02      dsk6     0        2097152  0/0       dsk6     ENA
sd dsk7-01      vol4-02      dsk7     0        2097152  1/0       dsk7     ENA
sd dsk8-01      vol4-02      dsk8     0        2097152  2/0       dsk8     ENA
pl vol4-03      vol4         ENABLED  ACTIVE   LOGONLY  CONCAT    -        RW
sd dsk9-01      vol4-03      dsk9     0        130      LOG       dsk9     ENA
```

5.3.4.2 Using Individual Commands

For complete control over how the mirrored and striped volume is configured, use the `volmake` and `volume` commands. Using these commands allow you to specify how disks are used. For example the following commands creates a 3–way stripe, adds a 3–way striped plex (mirror) so that the volume's data is mirrored across SCSI buses, and adds a DRL on a separate disk:

```
# volmake sd dsk6-01 dsk6,0,2097152
# volmake sd dsk7-01 dsk7,0,2097152
# volmake sd dsk8-01 dsk8,0,2097152
# volmake plex vol4-01 layout=stripe st_width=64k \
sd=dsk6-01,dsk7-01,dsk8-01
# volmake sd dsk10-01 dsk10,0,2097152
# volmake sd dsk11-01 dsk11,0,2097152
# volmake sd dsk12-01 dsk12,0,2097152
# volmake plex vol4-02 layout=stripe st_width=64k \
sd=dsk10-01,dsk11-01,dsk12-01
# volmake sd dsk9-01 dsk9,0,2097152
# volmake plex vol4-03 logsd=dsk9-01
# volmake -U fsgen vol vol4 plex=vol4-01,vol4-02,vol4-03
# volume start vol
```

5.3.5 Creating a RAID5 Volume

A RAID5 volume provides an alternative method to mirroring (RAID1) for improving data availability. A RAID5 volume contains a single plex, consisting of multiple subdisks derived from three or more disks. Data is striped across the subdisks, along with parity information that provides data redundancy.

Compared to a mirrored volume, a RAID5 volume requires fewer disks to improve data availability . For example, a 5-way stripe set requires six disks if configured as RAID5, compared to ten disks if it were mirrored and striped. However, there are disadvantages to using RAID5 volumes that might make usin mirrored or mirrored and striped volumes more desirable:

- RAID5 write-performance is often slower because both the data and new parity information are written. A single write to a volume often translates into two reads followed by two writes in order to read, modify, and write the volume's new data and new parity.
- If a disk fails, a write to any one of the volume's disks translates to first reading all the disks before the data and parity are written. A read to a RAID5 volume with a failed disk may require reading all the other disks instead.
- Data availability is not as high as with mirroring. For example, if a second disk fails in RAID5 volume, all the volume's data on those disks is lost because the parity information can only be used to recover data when one disk fails.

Despite the disadvantages, using a RAID5 volume might make sense either for read intensive environments or to improve availability on rarely accessed data.

You must configure a RAID5 log with a RAID5 volume. A RAID5 log is required to recover the volumes data when the system boots after a system failure. A RAID5 log differs from a mirrored volume DRL in that RAID5 logs contain the data that was being written to the volume when the failure occurred. The data in the RAID5 log is required to recover a RAID5 volume running in degraded mode due to a failed disk. Therefore, a RAID5 log is necessary to maintain the volume's data integrity after a failure. A mirrored volume DRL is not needed for data integrity, rather it is used only to accelerate the recovery process. When creating a RAID5 volume with the `volassist` command, a log is created by default.

The stripe width used for a RAID5 volume is typically smaller than the stripe width used for striping (RAID0) to lessen the performance impact of RAID5 writes. Unlike striping (RAID0), splitting up a write across all of the disks within stripe-set improves performance because reading existing data to determine the new parity is not necessary when writing a full, RAID5 row of data. For example, writing 64KB of data to a five column RAID5 stripe with a stripe width of 64KB may involve two parallel reads followed by two parallel writes (for example, reading both the existing data and parity, then writing the new data and new parity).

However, writing the same 64KB of data to a five-column RAID5 stripe with a stripe width of only 16KB could instead allow the 64KB of data to immediately write to disks (for example, five parallel writes to the four data disks and the one parity disk) because the new parity for the RAID5 row is determined from the 64KB of data, thereby making any reads of the old data or parity unnecessary. The RAID5 default stripe width of 16KB and usually works best for most environments.

5.3.5.1 Using the `volassist` Command

Follow these steps to create a RAID5 volume:

1. Create a RAID5 volume by entering the following command:

   # **volassist [-g *disk_group*] make *volume_name length* **
   layout=raid5 [*options*]

   For example, to create a RAID5 volume called `volraid` that is 100 MB, enter:

   **volassist make volraid 100m layout=raid5 nstripe=4 **
   dsk6 dsk7 dsk8 dsk9 dsk10

2. Display the results by entering the following command:

   # **volprint -ht *volume_name***

   For example, to display information for a volume called `volraid`, enter:

   # **volprint -ht volraid**

   Output similar to the following is displayed:

```
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  volraid      raid5        ENABLED  ACTIVE   204864   RAID       -
pl volraid-01   volraid      ENABLED  ACTIVE   204864   RAID       4/32     RW
sd dsk6-01      volraid-01   dsk6     0        68288    0/0        dsk6     ENA
sd dsk7-01      volraid-01   dsk7     0        68288    1/0        dsk7     ENA
sd dsk8-01      volraid-01   dsk8     0        68288    2/0        dsk8     ENA
sd dsk9-01      volraid-01   dsk9     0        68288    3/0        dsk9     ENA
pl volraid-02   volraid      ENABLED  LOG      1280     CONCAT     -        RW
sd dsk10-01     volraid-02   dsk10    0        1280     0          dsk10    ENA
```

5.3.5.2 Using Individual Commands

For complete control over how the volume is configured, use the `volmake` and `volume` commands. Using these commands allow you to specify how disks are used. For example, to create a RAID5 volume called `volraid` that is 100 MB, enter:

```
# volmake sd dsk6-01 dsk6,0,68288
# volmake sd dsk7-01 dsk7,0,68288
# volmake sd dsk8-01 dsk8,0,68288
# volmake sd dsk9-01 dsk9,0,68288
# volmake plex volraid-01 layout=raid5 st_width=16k \
 sd=dsk6-01,dsk7-01,dsk8-01,dsk9-01
# volmake sd dsk10-01 dsk10,0,1280
# volmake plex volraid-02 logsd=dsk10-01
# volmake -U raid5 vol volraid plex=volraid-01,volraid-02
```

```
# volume start volraid
```

5.3.5.3 Adding a RAID5 Log

You should always configure a log with a RAID5 volume. You should add a log if a RAID5 volume does not have a log or if the current log fails due to a disk failure.

To add a RAID5 log to a RAID5 volume using the volassist command, enter:

volassist [-g *disk_group*] addlog *volume_name* [*disk_name*]

For example, to create a log for the RAID5 volume called volraid, enter:

```
# volassist addlog volraid
```

Alternatively, you can use the volmake and volplex commands to add a RAID5 log to a RAID5 volume. For example, to create a log for the RAID5 volume called volraid, enter:

```
# volmake sd dsk10-01 dsk10,0,1280
# volmake plex volraid-02 logsd=dsk10-01
# volplex att volraid volraid-02
```

5.4 Configuring LSM Volumes For Use

Once you create an LSM volume, you can use the LSM volume in the same manner that you would use a disk partition. For example, you can put a file system on it, configure a database to use the volume as a raw device, use the LSM volume for additional system swap, and so on. Because LSM adheres to the same interfaces as any other Unix disk driver, anything that can be configured to use a disk or disk partition can use an LSM volume instead.

The following sections provide examples on how to use an LSM volume for file systems, secondary swap, and a raw device such as third party databases.

5.4.1 Using LSM Volumes with UFS

Follow these steps to create a UFS file system on an LSM volume:

1. Create a volume with a volume usage type of fsgen as described in the previous sections.
2. Specify the LSM volume's character (raw) device name to the newfs command. LSM character special devices are in the /dev/rvol directory, so to configure an UFS file on an LSM volume, enter:

```
# newfs [specific_options] /dev/rvol/disk_group/
volume_name
```

In this command, *disk_group* is the volume's disk group and *volume_name* is the volume's name.

Volume special device files for volumes in the `rootdg` disk group are in the `/dev/rvol` and `/dev/rvol/rootdg` directories, so it's not necessary to specify the name of the disk group for volumes in the `rootdg` disk group. See the `newfs`(8) reference page for more information on the `newfs` options and creating UFS file systems.

For example, to create a UFS file system on the LSM volume called `vol_mirr` in the `rootdg` disk group, enter:

```
# newfs /dev/rvol/vol_mirr
```

3. Use the LSM block special device name to mount the file system. For example to mount the LSM volume called `vol_mirr` on `mnt2`, enter:

```
# mount /dev/vol/vol_mirr /mnt2
```

Once a UFS file system is placed on an LSM volume, the volume's configuration can be changed to `online`. For example, the volume can be mirrored or moved to occupy a different disk using the `volassist mirror` or `volassist move` commands. Also, a volume snapshot can be taken for quick data backup using the `volassist snapshot` command as described in Chapter 6. Note, however that UFS file system can not be dynamically resized. Therefore, do not resize the volume using the `volassist grow` or `volassist shrink` commands. Resizing an LSM volume containing a UFS file system may lead to data loss.

5.4.2 Using LSM Volumes with AdvFS

Using LSM with AdvFS is a common and effective way to manage the system's storage and file systems. You use LSM to manage and provide storage and use AdvFS to store, manage, and provide files and file systems. Using LSM to manage all the system's storage provides the greatest flexibility for spreading the performance and space needs across different hardware, regardless of how that storage is used. For example, whether the volumes are used for AdvFS, UFS, databases, or swap.

Common LSM volume configurations used for AdvFS domains are:

- Mirrored volumes, which maintains data and system availability in the event of an AdvFS domain or a system panic (crash) that is caused by a disk failure.
- Striped volumes, which spreads the file system's I/O, including AdvFS transaction log I/O, across multiple disks for better performance.

- Mirrored/striped volumes, which maximizes both availability and performance.

Use the following guidelines when creating an LSM volume for an AdvFS domain:

- When resizing an AdvFS domain's storage, use AdvFS's addvol and rmvol commands rather than resizing the LSM volume itself. See the AdvFS documentation for more information on these commands.
- The volassist command's default stripe-width of 64KB usually works best with AdvFS.
- When using multiple, striped LSM volumes within the same AdvFS multi-volume domain, configure the same number of disks within the LSM striped volume. For example, if six disks were used to create two, LSM striped volumes in an AdvFS multi-volume domain, configure both volumes as 3-way striped sets instead of as one 2-way striped set and one as a 4-way striped set.

5.4.2.1 Using an LSM Volume Within an AdvFS Domain

To use an LSM volume within an AdvFS file domain, create a volume with a volume usage type of fsgen as described in the previous sections. Once the LSM volume is created, specify the LSM volume's block device name using either the mkfdmn or addvol command. LSM block special devices reside in the /dev/vol directory. For example, to use the mkfdmn command to create an AdvFS file domain using an LSM volume, enter:

```
# mkfdmn [options] /dev/vol/disk_group/volume_name \
domain_name
```

In this command *disk_group* is the name of the volume's disk group and *volume_name* is the name of the volume. Volume special device files for volumes in the rootdg disk group are in the /dev/vol and /dev/vol/rootdg directories, so it is not necessary to specify the *disk_group* name for volumes in the rootdg disk group.

For example, to create an AdvFS domain called dom1 on the LSM volume called vol_mirr1 in the rootdg disk group, enter:

```
# mkfdmn /dev/vol/vol_mirr1 dom1
```

See the mkfdmn(8) reference page for more information on using the mkfdmn options and creating AdvFS domains.

Once the file domain is created, create an AdvFS fileset and mount it in the usual manner. For example:

```
# mkfset dom1 fs1
# mount dom1#fs1 /mnt2
```

5.4.2.2 Adding an LSM Volume into an Existing AdvFS Domain

To add an LSM volume into an existing AdvFS domain, enter:

```
# addvol /dev/vol/disk_group/volume_name domain_name
```

Where *disk_group* is the name of the volume's disk group and *volume_name* is the name of the volume. Volume special device files for volumes in the `rootdg` disk group are in the `/dev/vol` and `/dev/vol/rootdg` directories, so it is not necessary to specify the *disk_group* name for volumes in the `rootdg` disk group.

For example, to add an AdvFS domain called `dom1` on the LSM volume called `vol_mirr2` in the `rootdg` disk group, enter:

```
# addvol /dev/vol/vol_mirr2 dom1
```

5.4.2.3 Removing an LSM Volume from AdvFS Domain

To remove an LSM volume from an AdvFS domain, enter:

```
# rmvol /dev/vol/disk_group/volume_name domain_name
```

In this command *disk_group* is the name of the volume's disk group and *volume_name* is the name of the volume. Volume special device files for volumes in the `rootdg` disk group are in the `/dev/vol` and `/dev/vol/rootdg` directories, so it is not necessary to specify the *disk_group* name for volumes in the `rootdg` disk group.

For example, to remove an LSM volume called `vol_mirr1` in the `rootdg` disk group from an AdvFS domain called `dom1`, enter:

```
# rmvol /dev/vol/vol_mirr1 dom1
```

Output similar to the following is displayed:

```
rmvol: Removing volume '/dev/vol/vol_mirr1' from domain 'dom1'
rmvol: Removed volume '/dev/vol/vol_mirr1' from domain 'dom1'
```

5.4.3 Using LSM Volumes for Secondary Swap Space

The system swap space is a vital system resource. If disk errors occur in the swap space, a system crash is likely to occur. You can use an LSM mirrored volume for the secondary swap space to guard against disk I/O errors in the secondary swap space.

Follow these steps to create an LSM mirrored volume for the secondary swap space:

1. Create an LSM volume in the `rootdg` disk group with a usage type of `gen` and set the volume's start options to `norecov`

2. Add the volume as secondary swap space using the swapon command

If you are adding multiple disks as LSM volumes to secondary swap space, add the disks as several individual LSM volumes rather than striping or concatenating them into a single, larger LSM volume. Adding multiple, individual LSM volumes is preferable because the swapping algorithm automatically distributes its data across multiple disks to improve performance.

Note

Do not configure DRL on swap volumes. Mirror resynchronization is not necessary after a crash for volumes used for swap, and configuring DRL on swap volumes interferes with crash dumps.

The following commands create and add a mirrored volume called swapvol1 with a size of 102400 sectors to the secondary swap space:

```
# volmake sd dsk8-01 dsk8,0,102400
# volmake sd dsk9-01 dsk9,0,102400
# volmake plex vol_swap2-01 sd=dsk8-01
# volmake plex vol_swap2-02 sd=dsk9-01
# volmake -U gen vol vol_swap2 \
plex=vol_swap2-01,vol_swap2-02 \
start_opts=norecov
# volume start vol_swap2
```

To display the results, enter:

```
# volprint -ht vol_swap2
```

Output similar to the following is displayed:

```
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_swap2    gen          ENABLED  ACTIVE   102400   ROUND     -
pl vol_swap2-01 vol_swap2    ENABLED  ACTIVE   102400   CONCAT    -        RW
sd dsk8-01      vol_swap2-01 dsk8     0        102400   0         dsk8     ENA
pl vol_swap2-02 vol_swap2    ENABLED  ACTIVE   102400   CONCAT    -        RW
sd dsk9-01      vol_swap2-02 dsk9     0        102400   0         dsk9     ENA
```

Once the LSM volume is created, you can configure to use it as a swap device like any other disk device. For example, to configure the LSM volume for swap using the swapon command, enter:

```
# swapon /dev/vol/vol_swap2
```

Then add the LSM special device file to the swapdevice kernel attribute value within the vm: section of the /etc/configtab file. For example:

```
vm:
          swapdevice=/dev/disk/dsk1b, /dev/vol/vol_swap2
```

See the *System Administration* and the swapon(8) and the sysconfig(8) reference pages for more information on adding additional swap space.

5.4.4 Using LSM Volumes with Databases and Other Software

Databases and other software that directly use disk partitions to perform raw I/O can also be configured to use LSM volumes.

To do so, create a volume with a usage type of gen, then configure it to be used with the database or other software by using the volume's character special device file located in the /dev/rvol/*disk_group* directory where *disk_group* is the volume's disk group name. Note that volume special device files for volumes in the rootdg disk group are in the /dev/rvol, /dev/rvol/rootdg, /dev/vol and in the /dev/vol/rootdg directories, so it is not necessary to specify the disk group name for volumes in the rootdg disk group.

Often databases or other software that perform raw I/O require the special device file to have certain settings for the access permissions, mode, user, and/or group. The special device file settings for LSM volumes can be specified when the volume is created. For example:

```
# volassist -U gen make vol_db1 32g user=dba group=dba \
mode=0600
```

To display the LSM volumes special device file's access permissions generated by from the above example, enter:

```
# ls -l /dev/*vol/vol_db1
```

Output similar to the following is displayed:

```
crw-------   1 dba      dba       40,  8 Jun 28 16:33 /dev/rvol/vol_db1
brw-------   1 dba      dba       40,  8 Jun 28 16:33 /dev/vol/vol_db1
```

Once the volume is created, do not change these attributes using standard UNIX commands such as the chown, chgrp, or chmod commands. To change the owner, group, or mode of LSM volume special device files, use the LSM voledit command. For example, to change user and group to dba and the mode to 0600 for a volume called vol_db1 in the rootdg disk group, enter:

```
# voledit set user=dba group=dba mode=0600 vol_db1
```

6

Managing LSM Objects

This chapter describes how to manage LSM objects, including disks, disk groups, volumes, plexes, and subdisks using LSM commands. The tasks described in this chapter can also be accomplished by using:

- The Storage Administrator. See Chapter 9 for more information on the Storage Administrator.
- The voldiskadm menu interface. See Appendix D for more information on the voldiskadm menu interface.
- The Visual Administrator. See Appendix B for more information on the Visual Administrator

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
```

6.1 Managing Disk Groups

As discussed in Chapter 5, the voldg command is used for creating new disk groups and adding or removing disks into that disk group. The voldg command is also used to perform other disk group operations such as obtaining a summary of free space within the disk group and importing and deporting disk groups.

See the voldg(8) reference page for more information on other disk group operations

6.1.1 Displaying Free Space Within A Disk Group

To display a summary of the free storage space within a disk group, enter the following command:

```
# voldg [-g disk_group] [-qa] free [disk_name]
```

For example, to display a summary of free storage space within the rootdg disk group, enter:

```
# voldg free
```

Output similar to the following is displayed:

```
GROUP      DISK       DEVICE     TAG        OFFSET    LENGTH    FLAGS
rootdg     dsk2       dsk2       dsk2       204800    3901568   -
rootdg     dsk3       dsk3       dsk3       0         4106368   -
rootdg     dsk4       dsk4       dsk4       0         4106368   -
rootdg     dsk5       dsk5       dsk5       0         4106368   -
rootdg     dsk6       dsk6       dsk6       0         4106368   -
rootdg     dsk7       dsk7       dsk7       0         4106368   -
rootdg     dsk9       dsk9       dsk9       0         4106368   -
rootdg     dsk11      dsk11      dsk11      0         4106368   -
```

6.1.2 Deporting and Importing Disk Groups

After a disk group is created as described in Chapter 5, 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. All the volumes within the disk group should be stopped before deporting the disk group.

Follow these steps to deport a disk group:

1. Stop all volumes within the disk group by entering the following command:

   # **volume [-g *disk_group*] stopall**

   For example, to stop all the volumes in a disk group called `dg1`, enter:

   # **volume -g dg1 stopall**

2. Deport the disk group by entering the following command:

   # **volume deport *disk_group***

   For example, to deport a disk group called `dg1`, enter:

   # **voldg deport dg1**

To reenable access to a disk group, you import the disk group, then restart the volumes.

Follow these steps to import a disk group:

1. Import the disk group by entering the following command:

   # **volume import *disk_group***

   For example, to import a disk group called `dg1`, enter:

   # **voldg import dg1**

2. Start all volumes within the disk group by entering the following command:

   # **volume [-g *disk_group*] startall**

   For example, to start all the volumes in a disk group called `dg1`, enter:

```
# volume -g dg1 startall
```

6.2 Managing Disks Using the voldisk and voldg Commands

Chapter 5 discussed how to add disks to LSM and how to display them. This section covers other disk operations that you can perform using the voldisk and voldg commands including removing disks, configuring disks for hot sparing, renaming disks, evacuating and replacing LSM disks, and reconfiguring the private regions settings on a disk.

When manipulating disks with the LSM software, it is important to understand the difference between a **disk access** name and a **disk media** name.

The name of a disk assigned by the operating system is referred to as the disk access name. For example, the disk access name might be dsk0, dsk1, dsk2 and is the name used when first initializing a disk for use with the LSM software.

When the disk is added into an LSM disk group, the disk is assigned an LSM disk media name. You can use any name you want for the disk media name. Often the same disk access name is used for the disk media, but you can use any string or name such as my_data_disk. The LSM voldiskadd interactive utility uses a disk media naming convention of disk*nn* where *nn* are numbers.

By mapping disk access names to LSM disk media names, the LSM disk group configuration is independent of the system's disk naming. Therefore, changes to disk access names, for example adding or removing hardware, do not affect the LSM software. Also, the mapping of disk access names to LSM disk media names provides flexibility when moving disks between systems where a disk will have a different disk access name on the system to which it is move.

The voldisk command is used to manipulate disks for use with the LSM software that are usually not configured into a disk group, so the system's disk access name is used with this command. You can use the voldg command to manipulate disks that are configured into an LSM disk group, so the disk media is used with this command. You can use the voldisk list command to display both a disk's access and media names.

Many of the operations described in this section can also be done by using the Storage Administrator and the voldiskadm menu interface. See Chapter 9 for information on using the Storage Administrator. See Appendix C for information on using the voldiskadm menu interface.

6.2.1 Adding A Disk

When you add a disk under LSM control, the disk is either initialized or encapsulated. If the disk is not set up, initialize it. If you are placing a disk with partitions that are in use under LSM control, 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 disk group.

See Chapter 3 for more information on encapsulating a disk.

To initialize a disk, enter:

voldisksetup -i *diskname*

For example, to initialize disks called `dsk4` and `dsk5` as LSM sliced disks, enter:

voldisksetup -i dsk4

voldisksetup -i dsk5

Specifying the disk name with no partition letter causes the `voldisksetup -i` command to initialize the disk as sliced.

6.2.2 Displaying Disks

To display a list of disks, enter:

voldisk list

Output similar to the following is displayed:

```
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        rootdg      online
dsk7         sliced    -           -           online
dsk8         sliced    dsk8        dg1         online
dsk9         sliced    dsk9        dg1         online
dsk10        sliced    -           -           unknown
dsk11        sliced    -           -           unknown
```

A value of `online` in the `STATUS` column indicates that a disk was initialized for use with the LSM software.

In the above example:

- The disks called `dsk2` through `dsk6` were initialized for use with the LSM software and were added into the `rootdg` disk group.

- The disk called dsk7 was initialized for use with the LSM software because its status is online, but it is not currently configured within any disk groups.
- The disks called dsk8 and dsk9 were initialized for use with the LSM software and were added into the dg1 disk group.
- The disks called dsk0, dsk1, dsk10, and dsk11 were not initialized for use with the LSM software because their status is unknown.

To display how much free disk space is available to create volumes, enter:

```
# voldg free
```

Output similar to the following is displayed:

```
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   -
rootdg       dsk6         dsk6         dsk6         0         4106368   -
dg1          dsk8         dsk8         dsk8         0         4106368   -
dg1          dsk9         dsk9         dsk9         0         4106368   -
```

The value in the LENGTH column displays the amount of free space on a disk.

The volassist help space command also displays detailed information about LSM disks. To use the volassist help space command to display detailed information about LSM disks, enter:

```
# volassist [-g disk_group] help space
```

For example, to display detailed disk space information in the dg1 disk group, enter:

```
# volassist -g dg1 help space
```

Output similar to the following is displayed:

```
Disk: dsk8 len=4106368 used=0 free=4106368 (100.00%)
Attributes:
  dm:dsk8 device:dsk8 da:dsk8
Free regions:
  16,4106368

Disk: dsk9 len=4106368 used=0 free=4106368 (100.00%)
Attributes:
  dm:dsk9 device:dsk9 da:dsk9
Free regions:
  16,4106368

Disk sets:
da:dsk8 space=4106368 used=0 free=4106368 (100.00%)
da:dsk9 space=4106368 used=0 free=4106368 (100.00%)
device:dsk8 space=4106368 used=0 free=4106368 (100.00%)
device:dsk9 space=4106368 used=0 free=4106368 (100.00%)
```

```
dm:dsk8 space=4106368 used=0 free=4106368 (100.00%)
dm:dsk9 space=4106368 used=0 free=4106368 (100.00%)
```

6.2.3 Renaming a Disk

Because disk access names are defined by the operating system and media names are defined by you, you can only rename disk media names for disks in a disk group.

To rename a disk, enter:

 # voledit rename *old_diskname new_diskname*

Follow these steps to rename `disk03` to `disk01`:

1. List the current disk names. For example:

   # voldisk list

   ```
   DEVICE       TYPE      DISK         GROUP        STATUS
   dsk8         sliced    dsk8         rootdg       online
   dsk9         sliced    disk03       rootdg       online
   dsk9d        simple    -            -            online
   ```

2. Rename the disk called `disk03` to `disk01`. For example:

   # voledit rename disk03 disk01

3. Confirm that the name change was successful. For example:

   # voldisk list

   ```
   DEVICE       TYPE      DISK        GROUP        STATUS
   dsk8         sliced    dsk8        rootdg       online
   dsk9         sliced    disk01      rootdg       online
   dsk9d        simple    -           -            online
   ```

6.2.4 Placing a Disk Off line

You can place a disk in an off line state to prevent LSM from accessing it. You must remove a disk from its disk group before you take it off line. An off line disk remains unavailable until you restore access to the disk by placing it on line.

You place a disk in an off line state to protect it from unintentional use, for example, if attempts to access it may have a negative effect on the system. You cannot take a disk that is in use off line.

To place a disk in an off line state, you must remove the disk from its disk group, and enter:

 # voldisk offline *disk_name*

For example, follow these steps to take a disk off line:

1. Remove the disk from its disk group. For example to remove a disk called `disk01` from the `rootdg` disk group, enter:

   # **voldg rmdisk disk01**

2. Place the disk in an off line state. To do so, you must use the disk access name because the disk no longer has an LSM disk media name after you remove it from the disk group. For example, to place a disk called `dsk8`, which has the disk name of `disk01`, in an off line state, enter:

   # **voldisk offline dsk8**

6.2.5 Placing a Disk On line

Placing a disk in an on line state restores access to a disk that is in an off line state. The disk is placed in the free disk pool and is accessible to LSM again. After bringing a disk back online, the disk must be added to a disk group before it can be used for volumes.

Only disks that are in an off line state can be placed in an on line state.

To place a disk in an on line state, enter:

voldisk online *disk_name*

6.2.6 Evacuating a Disk

You can evacuate (or move) the contents of the volumes 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 RAID5 volume, do not move the contents of the disk to another disk containing a copy of the mirrored volume or part of the striped/RAID5 volume.

To evacuate a disk from LSM control, enter:

volevac [-g *diskgroup*] *disk_name new_disk_name*

For example, to evacuate a disk called `dsk8` to a disk called `dsk9`, enter:

volevac dsk8 dsk9d

6.2.7 Removing an LSM Disk from a Disk Group

An LSM disk no longer in use can be removed from a disk group. Do not remove LSM disks that are in use (for example, contains 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 initialized for later use.

Follow these steps to remove a disk from an LSM disk group:

1. Verify that the LSM disk is not in use by subdisks by entering the following command:

   \# **volassist help**

   For example, to verify that a disk called `dsk8` is not in use, enter:

   \# **volassist help space | grep dsk8**

   Output similar to the following is displayed:

   ```
   Disk: dsk8 len=4106368 used=0 free=4106368 (100.00%)
     dm:dsk8 device:dsk8 da:dsk8
     da:dsk8 space=4106368 used=0 free=4106368 (100.00%)
     device:dsk8 space=4106368 used=0 free=4106368 (100.00%)
     dm:dsk8 space=4106368 used=0 free=4106368 (100.00%)
   ```

   A disk is not in use when 100% is displayed.

   If the LSM disk is in use, you can move or evacuate the data on that disk by using either the `volevac` command or the `volassist mv` command. See Section 6.2.6 for more information on evacuating a disk.

2. Remove the LSM disk from a disk group by entering the following command:

   \# **voldg [-g disk_group] rmdisk *diskmedia_name***

   For example, to remove a disk called `dsk8` from the `rootdg` disk group, enter

   \# **voldg rmdisk dsk8**

3. Display the results by entering the following command:

   \# **voldisk list**

   Output similar to the following is displayed:

   ```
   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         rootdg       online
dsk7         sliced    dsk7         rootdg       online
dsk8         sliced    -            -            online
dsk9         sliced    dsk9         rootdg       online
dsk10        sliced    dsk10        rootdg       online
dsk11        sliced    dsk11        rootdg       online
```

Notice that in this output that the disk called `dsk8` no longer has a disk media name or belongs in a disk group.

6.2.8 Replacing a Disk

You may 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 disk, move volumes to the new disk, and attempt to recover any redundant (mirrored or RAID5) volumes on the disk. You cannot recover non-redundant volumes. You should restore non-redundant 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 and place it in the free disk pool *before* you replace the disk. See Section 6.2.7 for more information on removing a disk from a disk group. 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, enter the `voldiskadm` command and choose `Replace a failed or removed disk` from the main menu. A list of disks is displayed from which you can choose a replacement disk. If you have disks that are initialized for use with the LSM software, but not added to a disk group, you can select one of those disks as a replacement. Do not choose the old disk drive as a replacement even though it may appear in the selection list. If there are no suitable initialized disks, you can add a new disk. SeeSection 6.2.1 for more information on adding a disk for use with the LSM software.

6.2.9 Recovering Volumes on a 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 RAID5 volumes. After successful recovery, the volumes should be available for use.

Alert icons and the Alert Monitor window may provide information when a volume recovery is needed.

If recovery of a volume is not possible, restore the volume from backup.

To recover a volume, enter:

volrecover [-g *disk_group*] [*volume_name*]

For example, to recover a volume called vol01, enter:

volrecover vol01

6.3 Managing Disk Groups

The following sections describe how to use LSM commands to manage disk groups.

6.3.1 Displaying Disk Group Information

A disk group must exist to place a disk in it. To use disk groups, you must know the names of the groups and what disks belong to each group.

To display disk group information, enter:

voldisk list

Output similar to the following is displayed:

```
DEVICE        TYPE      DISK         GROUP        STATUS
dsk5          simple    disk5        dg1          online
dsk8          sliced    dsk8         rootdg       online
dsk9          sliced    disk01       rootdg       online
dsk10         sliced    dsk2          -           online
```

Information displayed under the GROUP column shows the disk group in which the disk belongs. A blank GROUP entry, which is represented by a dash (-), means that the disk is not in a disk group.

In the above output, the disk called dsk5 is in the dg1 disk group, the disks called dsk8 and dsk9 are in the rootdg diskgroup, and the disk called dsk10 does not belong in a disk group.

To display detailed information about a disk group, enter:

voldg list *disk_group*

For example, to display detailed information about the rootdg disk group, enter:

voldg list rootdg

Output similar to the following is displayed:

```
Group:     rootdg
 dgid:      921709207.1025.rio.dec.com
 import-id: 0.1
```

```
flags:
copies: nconfig=default nlog=default
config: seqno=0.1618 permlen=726 free=698 templen=18 loglen=110
config disk dsk0g copy 1 len=726 state=clean online
config disk dsk1g copy 1 len=726 state=clean online
config disk dsk2 copy 1 len=726 disabled
config disk dsk3 copy 1 len=2993 disabled
config disk dsk4 copy 1 len=726 state=clean online
config disk dsk5 copy 1 len=726 state=clean online
log disk dsk0g copy 1 len=110
log disk dsk1g copy 1 len=110
log disk dsk2 copy 1 len=110 disabled
log disk dsk3 copy 1 len=453 disabled
log disk dsk4 copy 1 len=110
log disk dsk5 copy 1 len=110
```

This output is useful to determine:

- Free space in the disk group's LSM configuration database
- Which disk's private regions have an active copy of LSM's configuration databases

In the above example, the `rootdg` disk group has 698 blocks free out of 726, and disks called `dsk0`, `dsk1`, `dsk4`, and `dsk5` have a copy of the `rootdg` LSM configuration database.

6.3.2 Displaying Free Space in a Disk Group

Before you add volumes, make sure you have enough free disk space in a disk group to meet the storage needs of the volume.

To display free space in a disk group, enter:

voldg -g *disk_group_name* free

If you do not specify a disk group, the `voldg free` command displays the free space in the default disk group, `rootdg`. For example, to display the free space in the `rootdg` disk group, enter:

voldg free

Output similar to the following is displayed:

```
GROUP    DISK     DEVICE     TAG    OFFSET  LENGTH   FLAGS
rootdg   dsk8      dsk8      dsk8    726400  102672   -
rootdg   disk01    dsk9      dsk9    0       102128   -
```

The value displayed in the `LENGTH` column is the amount (in 512-byte sectors) of free space on a disk.

6.3.3 Adding a Disk To a Disk Group

You must place disks in to a disk group before LSM can use them. The default disk group (rootdg) is created during LSM installation and always exists on a system running LSM.

Follow these steps to add a disk to an existing disk group:

1. Identify initialized disks that do not belong to a disk group. For example:

   # **voldisk list**

   Output similar to the following is displayed:

   ```
   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        rootdg       online
   dsk7         sliced    -           -            online
   dsk8         sliced    dsk8        dg1          online
   dsk9         sliced    dsk9        dg1          online
   dsk10        sliced    -           -            online
   dsk11        sliced    -           -            unknown
   dsk12        sliced    -           -            unknown
   dsk13        sliced    -           -            unknown
   ```

   Initialized disks that do not belong to a disk group display a STATUS of online and a blank GROUP entry, which is represented by a dash.

   In the above output disks called dsk7 and dsk10 are initialized and not part of a disk group because their status is online and the GROUP column is blank.

2. Display disk groups. For example:

   # **voldg list**

   ```
   NAME         STATE     ID
   rootdg       enabled   927328730.1026.wdt2
   dg1          enabled   929455995.1168.wdt2
   ```

   The above output shows that the system has two disk groups: rootdg and dg1.

3. Enter the voldg adddisk command to add an initialized disk to a disk group:

   # **voldg adddisk [-g *disk_group*] *diskname***

   For example, to add the LSM sliced disk called dsk7 to the rootdg disk group, enter:

   # **voldg adddisk dsk7**

To add the LSM sliced disk called `dsk10` to a disk group called `dg1`, enter:

```
# voldg -g dg1 adddisk dsk10
```

6.3.4 Creating a Disk Group

You must place disks in to a disk group before LSM can use them. 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 its name must be unique. You can only use disks that are online and do not belong to a disk group to create a disk group.

To create a new disk group, enter:

```
# voldg init groupname disk_name
```

For example, to use the LSM sliced disk called `dsk5` to create a new disk group called `dg1`, enter:

```
# voldg init dg1 dsk5
```

6.3.5 Deporting a Disk Group

You can deport a disk group to make its volumes temporarily inaccessible. Once deported, a disk group cannot be used until it is imported. A disk group cannot be deported if any volumes in that disk group are in use.

While a disk group is deported, you cannot remove and reuse any of its disks. To remove disks, you must import the disk group and then either destroy the disk group (to remove all of its disks) or remove selected disks from the disk group.

When a disk group is deported, the host ID stored on all disks in the disk group is cleared and the disk group is not reimported automatically when the system reboots. However, if you specify a host in New Host Expert Option, the specified host imports the disk group when the system reboots.

You cannot deport the `rootdg` disk group.

To deport a disk group, close and unmount all volumes in the disk group and enter:

```
# voldg deport disk_group
```

For example, to deport a disk group called `dg1`, enter:

voldg deport dg1

6.3.6 Importing a 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 and 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.

Import may fail for a number of reasons. It may fail 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.

Follow these steps to import a disk group:

1. Make the disks accessible by entering the following command:

   # **voldisk define *diskname***

2. Enable local access to the disk group by entering the following command:

   # **voldg import *disk_group***

3. Start the volumes in the disk group by entering the following command:

   # **volrecover -g *disk_group* -sb**

When a disk group is created or imported on a system, that system writes a lock on all disks in the disk group. If you move disks from a system that has crashed or failed to detect the group before the disk is moved, the locks stored on the disks will remain and must be cleared, and the system returns the following error message:

```
voldg:disk group groupname: import failed: Disk
        in use by another host
```

To clear locks on a specific set of devices, enter the following command:

```
# voldisk clearimport diskname
```

In some cases, you may want to import a disk group when some disks are not available. The import command fails if some disks for the disk group are not found among the disk drives attached to the system. If the import command fails, one of the following error messages is displayed:

```
voldg: Disk group groupname: import failed: Disk for
        disk group not found

voldg:Disk group groupname: import failed: Disk group has
          no valid configuration copies
```

6.3.7 Increasing Free Space in a Configuration Database

Typically, LSM maintains four separate physical disks with active configuration database copies for each disk group. When the disk group runs out of space in the configuration database, LSM displays the following message when creating an LSM record:

```
volmake: No more space in disk group configuration
```

If you run out of disk space, you can increase the size of the configuration database on each disk that is smaller than the current disk group configuration by reducing the number of configuration log copies. Reducing the number of configuration copies on disks, effectively increases the amount of space available for the remaining configurations. However, make sure that there are sufficient copies of the database available for redundancy.

Follow these steps to reduce the number of configuration copies:

1. Display the amount of free space. For example:

   ```
   # voldg list rootdg
   ```

   Information similar to the following is displayed:

   ```
   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
   ```

The configuration database size and the log size of a disk group corresponds to the smallest configuration database size and log size of any disk in the disk group.

In this display, the `free=166` record indicates that there is enough space in the `rootdg` disk group to create 166 additional LSM configuration records.

In addition, disk `dsk13` has a configuration database size of 173 sectors, and disks `dsk11g` and `dsk10g` each have a configuration database size of 347 sectors. The configuration database size of the `rootdg` disk group is 173 sectors, which corresponds to the smallest configuration database.

2. Create a backup copy of the configuration database.
3. Reduce the number of configuration copies. For example, to reduce the number of configuration copies on `dsk13` from 2 to 1, enter:

   ```
   # voldisk moddb dsk13 nconfig=1
   ```

6.4 Managing Volumes

The following sections describe how to use LSM commands to manage LSM volumes. See Chapter 5 for information on creating a volumes.

6.4.1 Displaying Volume Information

To display the volume, plex, and subdisk record information for all volumes, enter:

```
# volprint -ht
```

LSM displays output similar to the following using the abbreviations in Table 6–1.

Table 6–1: volprint Abbreviations

Abbreviation	Meaning
dg	Disk Group
dm	Disk
pl	Plex
sd	Subdisk
v	Volume

```
Disk group: rootdg

DG NAME         NCONFIG      NLOG     MINORS   GROUP-ID
```

```
DM NAME         DEVICE       TYPE     PRIVLEN  PUBLEN   STATE
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

dg rootdg       default      default  0        921709207.1025.rio.dec.com

dm disk01       dsk3         sliced   4096     4106368  -
dm rz8a         dsk0a        nopriv   0        262144   -
dm rz8b         dsk0b        nopriv   0        261120   -
dm rz8d         dsk0d        nopriv   0        2618597  -
dm rz8e         dsk0e        nopriv   0        2618597  -
dm rz8g         dsk0g        simple   1024     0        -
dm rz10a        dsk1a        nopriv   0        262144   -
dm rz10b        dsk1b        nopriv   0        261120   -
dm rz10d        dsk1d        nopriv   0        2618597  -
dm rz10e        dsk1e        nopriv   0        2618597  -
dm rz10g        dsk1g        simple   1024     0        -
dm rz12         dsk2         sliced   1024     4109440  -
dm rz17         dsk4         sliced   1024     4109440  -
dm rz19         dsk5         sliced   1024     4109440  -

v  rootvol      root         ENABLED  ACTIVE   262144   ROUND     -
pl rootvol-02   rootvol      ENABLED  ACTIVE   262144   CONCAT    -        RW
sd rz10a-01p    rootvol-02   rz10a    0        16       0         dsk1a    ENA
sd rz10a-01     rootvol-02   rz10a    16       262128   16        dsk1a    ENA
pl rootvol-01   rootvol      ENABLED  ACTIVE   262144   CONCAT    -        RW
sd rz8a-01p     rootvol-01   rz8a     0        16       0         dsk0a    ENA
sd rz8a-01      rootvol-01   rz8a     16       262128   16        dsk0a    ENA

v  swapvol      swap         ENABLED  ACTIVE   261120   ROUND     -
pl swapvol-02   swapvol      ENABLED  ACTIVE   261120   CONCAT    -        RW
sd rz10b-01     swapvol-02   rz10b    0        261120   0         dsk1b    ENA
pl swapvol-01   swapvol      ENABLED  ACTIVE   261120   CONCAT    -        RW
sd rz8b-01      swapvol-01   rz8b     0        261120   0         dsk0b    ENA

v  vol-rz8d     fsgen        ENABLED  ACTIVE   2618597  SELECT    -
pl vol-rz8d-02  vol-rz8d     ENABLED  ACTIVE   2618597  CONCAT    -        RW
sd rz10d-01     vol-rz8d-02  rz10d    0        2618597  0         dsk1d    ENA
pl vol-rz8d-01  vol-rz8d     ENABLED  ACTIVE   2618597  CONCAT    -        RW
sd rz8d-01      vol-rz8d-01  rz8d     0        2618597  0         dsk0d    ENA

v  vol-rz8e     fsgen        ENABLED  ACTIVE   2618597  SELECT    -
pl vol-rz8e-02  vol-rz8e     ENABLED  ACTIVE   2618597  CONCAT    -        RW
sd rz10e-01     vol-rz8e-02  rz10e    0        2618597  0         dsk1e    ENA
pl vol-rz8e-01  vol-rz8e     ENABLED  ACTIVE   2618597  CONCAT    -        RW
sd rz8e-01      vol-rz8e-01  rz8e     0        2618597  0         dsk0e    ENA

Disk group: dg1

DG NAME         NCONFIG      NLOG     MINORS   GROUP-ID
DM NAME         DEVICE       TYPE     PRIVLEN  PUBLEN   STATE
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

dg dg1          default      default  6000     922382892.1625.rio.dec.com

dm dg101        dsk9         sliced   4096     4106368  -

v  v1           fsgen        ENABLED  ACTIVE   2097152  SELECT    -
pl v1-01        v1           ENABLED  ACTIVE   2097152  CONCAT    -        RW
sd dg101-01     v1-01        dg101    0        2097152  0         dsk9     ENA
```

To display information on a particular volume, enter:

```
# volprint -t volume_name
```

For example, to display the information about a volume called `volspec`, enter:

```
# volprint -t volspec
```

Output similar to the following is displayed;

```
Disk group: rootdg

V  NAME     USETYPE  KSTATE   STATE   LENGTH   READPOL  PREFPLEX

v  volspec  fsgen    ENABLED  ACTIVE  2097152  SELECT   -
```

To display the plexes for this volume, enter:

```
# volprint –e 'assoc="volspec"'
```

To list information on all volumes, enter:

```
# volprint –vt
```

To show detailed information for all volumes, enter:

```
# volprint -vl
```

6.4.2 Changing Volume Attributes

You can use the following commands to change volume attributes, such as ownership, permissions, and the values in the comment and command fields:

- The `voledit` command sets those attributes that are not usage-type-dependent.
- The `volume` command sets only those attributes that are usage-type-dependent.

To volume attributes, enter:

```
# volume set field=value ...volume_name
```

For example, to change the owner of `vol01` to `susan`, the group to `staff`, and the permissions to read/write for owner, group, and other, enter:

```
# voledit set user=susan group=staff mode=0666 vol01
```

Table 6–2 describes the attributes that you can set for each command.

Table 6–2: Setting Volume Attributes

Command	Attribute	Description
voledit	comment	The comment field
	tutil0, tutil1, tutil2	Temporary and permanent utility fields (for internal use only).
	putil0, putil1, putil2	
	fstype	String indicating file system type
	writeback	Boolean (on/off) specifying read error correction mode
	user	Owner of volume
	group	Group of volume
	mode	Permission mode for volume
volume	len	Numeric length of volume
	log type	(region/undef) specifier of dirty region logging mode for volume
	loglen	Length of the dirty region logging log
	start opts	Options to be executed with the volume start operation

6.4.3 Setting the Volume Read Policy

LSM offers the choice of three read policies:

- The round policy reads each plex in turn in round-robin fashion.
- The prefer policy reads preferentially from a plex that was labeled as the preferred plex. You can change the read policy from prefer to round (or vice versa in the case of prefer) or to a different preferred plex.
- The select policy chooses a default policy based on plex association to the volume.

The volume rdpol command sets the read policy for a volume. Use one of the following commands to set a read policy:

volume rdpol round *volume_name*

or

volume rdpol prefer *volume_name preferred_plex_name*

For example, to set the read policy for volume vol01 to a round-robin read, enter:

```
# volume rdpol round vol01
```

To set the policy for the same volume to read preferentially from the plex vol01-02, enter:

```
# volume rdpol prefer vol01 vol01-02
```

6.4.4 Recovering a Volume

You can recover a volume. The recovery operations depend on the type of volume and include starting disabled volumes, resynchronizing mirrors in mirrored volumes, and resynchronizing parity in RAID5 volumes. After successful recovery, the volume should be available for use.

Alert icons and the Alert Monitor window may provide information when a volume recovery is needed.

In some cases, recovery may not be possible. If the volume recovery fails, you can attempt to restore the volume from backup.

To recover all volumes, enter:

```
# volrecover [-g disk_group] startall
```

6.4.5 Stopping a Volume

LSM automatically starts and stops a volume whenever the system is booted or shut down. When a volume will no longer be needed, you should stop the volume before deleting it. You cannot stop a volume if it is in use or it has a mounted file system.

To stop a volume, enter:

```
# volume stop [-g group_name] volume_name ...
```

For example, to stop a volume called vol1 in the dg1 disk group, enter:

```
# volume stop -g dg1 vol1
```

To stop all volumes in the ENABLED state, enter:

```
# volume stopall
```

6.4.6 Starting a Volume

LSM automatically starts and stops a volume whenever the system is booted or shut down, so it is not normally necessary to have to explicitly start a volume. You may need to manually start a volume after it is first created by using the volmake command, or if it was explicitly stopped as described in Section 6.4.5. Because you cannot use RAID5 volumes in a TruClustered environment, they can not be started in clusters.

To start a volume, enter

```
# volume start [-g group_name] volume_name ...
```

For example, to start a volume called vol1 in the dg1 disk group, enter:

```
# volume start -g dg1 vol1
```

Starting a volume changes the volume state from DISABLED or DETACHED to ENABLED. If you cannot start a volume, it remains in its current state.

6.4.7 Renaming a Volume

You can rename a volume. The new volume 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 allows you to specify a new mount point for the file system. You cannot rename volumes that are part of an AdvFS domain.

To rename a volume, enter:

```
# voledit rename old_volume_name new_volume_name
```

For example, to rename a volume called v1 to vol01, enter:

```
# voledit rename v1 vol01
```

6.4.8 Removing a Volume

You can remove a volume. Removing a volume destroys all of the data in that volume. Only remove a volume 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 will only work if the file system is UFS.

To remove a volume, enter:

Follow these steps to remove a volume:

1. Close the volume, or if it contains a file system, unmount it.

   For example, if the volume is used in an AdvFS multi-volume domain, enter the following command to deconfigure the LSM volume from AdvFS and close it:

```
# rmvol /dev/vol/advfs_vol
```

If the volume contained a UFS file system that is no longer needed, enter the following command to unmount it and close the LSM volume:

```
# umount /dev/rvol/v_ufs
```

2. Stop all LSM activity to the volume by entering the following command:

```
# volume stop volume_name
```

For example, to stop a volume called v_ufs, enter:

```
# volume stop v_ufs
```

3. Remove the volume along with all associated plexes and subdisks by entering the following command:

```
# voledit -r rm volume_name
```

For example, to remove the volume called volspan, enter:

```
# voledit -r rm volspan
```

6.5 Managing Plexes

The following sections describe how to use LSM commands to manage plexes.

6.5.1 Displaying Plex Information

A plex contains a copy of a volume's data. A volume is mirrored when there are two or more plexes attached to a volume. Listing plexes helps you identify free plexes that you can use to create volumes.

To display free plexes, enter:

```
# volprint –pt
```

To display detailed information about all plexes, enter:

```
# volprint –lp
```

To display detailed information about a specific plex, enter:

```
# volprint –l plex_name
```

6.5.2 Detaching a Plex

When you create a volume and place it on line (ENABLED), LSM provides commands that allow you to temporarily detach a plex from the volume. This is useful, when the hardware on which a plex resides needs repair or when a volume is left unstartable and you must manually start a source plex for the volume.

A detached plex maintains the association to its volume; however, the plex cannot be used for I/O. While the plex is detached, its state is set to STALE. If a `volume start` command runs on the volume (for example, after a system reboot), the plex is reattached and made ACTIVE.

To detach a plex in a mirrored volume, enter:

volplex det *plex_name* ...

For example, to detach a plex called `vol01-02`, enter:

volplex det vol01-02

6.5.3 Attaching a Plex

When the disk is repaired or replaced and is ready for use, you must attach a plex to put it on line (ACTIVE).

If the volume is in use (ENABLED), enter:

volplex att *volume_name plex_name* ...

For example, to attach a plex called `vol01-02` on a volume called `vol01`, enter:

volplex att vol01 vol01-02

If the volume is not in use (not ENABLED), enter:

volmend on *plex_name*

For example, for a plex called `vol01-02`, enter:

volmend on vol01-02

In this case, the state of `vol01-02` is set to STALE, so that the next time the volume starts, the data on the plex is revived from the other plex, and incorporated into the volume with its state set to ACTIVE.

If it becomes necessary to manually change the state of a plex, see the `volmake`(8) and `volmend`(8) reference pages for more information.

6.5.4 Moving Plexes

You can move the date on a plex onto a new plex. For a move operation to be successful, you must ensure:

- The old plex is an active part of an active (ENABLED) volume.

- The new plex is at least the same size or larger than the old plex. A smaller or more sparse new plex may result in an incomplete move.

  If the new plex is longer, or less sparse, than the original plex, the data that exists on the original plex is moved onto the new plex. Any area that was not on the original plex, but is represented on the new plex, is filled from other complete plexes associated with the same volume. If the new plex is longer than the volume itself, the remaining area of the new plex above the size of the volume is not initialized.

- The new plex is not associated with another volume.

To move data from one plex to another, enter:

```
# volplex mv original_plex new_plex_name
```

6.5.5 Copying Plexes

You can copy a plex to another plex. The copied plex contains a complete copy of the volume data. To copy the contents of a volume to a specified plex, you must ensure that the volume to be copied is not enabled and that the plex is not associated with any other volume.

To copy a plex, enter:

```
# volplex cp vol_name new_plex_name
```

The *new_plex* is not associated with the specified volume *vol_name*.

6.5.6 Backup Using a Plex

This section shows how to backup data by manipulating plexes manually. See Section 7.4 for information on how to automatically backed up data by using the `volassist` command.

You can backup of a volume if the volume is mirrored by taking one of the volume's plexes off line for a period of time. This eliminates the need for extra disk space for the purpose of backup only. However, it also eliminates redundancy of the volume during the backup.

Follow these steps to perform a backup of a mirrored volume on an active system:

1. Stop I/O activity and flush any buffers to improve the consistency of the backup. For example, Compaq recommends briefly unmounting the UFS volume in order to create a complete and consistent backup.
2. Disassociate one of the volume's plexes (`vol-01`, for this example):

   ```
   # volplex dis vol-01
   ```

This command should only take a few seconds to execute. It leaves the `vol-01` plex available as an image of the volume frozen at the time of the disassociation.

3. Resume I/O activity. For example, if the volume contained UFS, remount it.

4. Create a temporary volume:

   # **volmake -Ufsgen vol vtmp plex=vol-01**

   # **volume start vtmp**

5. Check the temporary volume, if necessary:

   # **fsck -p /dev/rvol/vtmp**

6. Create a backup using the temporary volume:

   # **dump 0 /dev/rvol/vtmp**

7. Reattach the plex to the volume to mirror the volume:

   # **volplex dis vol-01**

   # **volplex att *volume-name* vol-01**

6.5.7 Removing Plexes

You can remove a plex when it is no loger need, for example:

- To reduce the number of plexes in a volume.
- To increase the length of another plex.
- To remove temporary plex that was created to backup a volume and is no longer required.
- To change the layout of a plex from concatenated to striped.

Note

To save the data on a plex that you plan to remove, you need to know the original configuration of that plex. Several parameters from that configuration, such as stripe width and subdisk ordering, are critical to the construction of a new plex which would contain the same data. Before you remove a plex, record its configuration.

To remove a disk you must disassociate the plex from the volume and remove the plex and any associated subdisks.

To disassociate a plex from a volume and remove it, enter:

volplex -o rm dis *plex_name*

For example, to disassociate a plex called `vol01-02` and remove all associated subdisks, enter:

Note

Without the `-o rm`, the `volplex` command disassociates the plex and subdisks, but does not remove them. To remove the disassociated plex and subdisks, enter:

```
# voledit -r rm plex_name
```

The `voledit -r` command removes multiple objects. Use this command with caution.

6.6 Managing Subdisks

The following sections describe how to use LSM commands to manage subdisks.

6.6.1 Displaying Subdisks

To display general information for all subdisks, enter:

```
# volprint –st
```

Output similar to the following is displayed:

```
Disk group: rootdg

SD NAME         PLEX         DISK     DISKOFFS LENGTH   [COL/]OFF DEVICE   MODE

sd dsk0a-01     vol-dsk0a-01 dsk0a-AdvFS 0     262144   0         dsk0a    ENA
sd dsk0b-01     vol-dsk0b-01 dsk0b-swap 0      262144   0         dsk0b    ENA
sd dsk0g-01     vol-dsk0g-01 dsk0g-AdvFS 0     1526572  0         dsk0g    ENA
sd dsk1g-01     vol-dsk1g-01 dsk1g-AdvFS 0     1429762  0         dsk1g    ENA
sd lsm03-01     vol01-01     lsm03    0        409600   0         dsk3     ENA
sd root01-01    rootvol-01   root01   0        262144   0         dsk1a    ENA
sd swap01-01    swapvol-01   swap01   0        258048   0         dsk1b    ENA
```

To display detailed information about a subdisk, enter:

```
# volprint –l subdisk_name
```

For example, to obtain all database information on a subdisk called `disk02-01`, enter:

```
# volprint -l vol-dsk1g-01
```

Output similar to the following is displayed:

```
# Disk group: rootdg
```

```
Plex:     vol-dsk1g-01
info:     len=1429762
type:     layout=CONCAT
state:    state=ACTIVE kernel=ENABLED io=read-write
assoc:    vol=vol-dsk1g sd=dsk1g-01
flags:    busy complete
```

6.6.2 Joining Subdisks

You can join two or more subdisks together to form a single, larger subdisk. Subdisks can only be joined together 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 striped volume, the subdisks must be in the same column.

To join a subdisk, enter:

\# **volsd join *subdisk1 subdisk2 new_subdisk***

6.6.3 Splitting Subdisks

You can divide a subdisk into two or more subdisks. Once split, the smaller subdisks can be moved elsewhere or joined later. This 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 subdisk remains the same as the selected subdisk. The new, smaller subdisks occupy the same regions of the disk that the original subdisk occupied.

A log subdisk cannot be split.

To split a subdisk, enter:

\# **volsd –s *size* split *sd newsd newsd2***

For example, to split a subdisk called `disk02-01` to a subdisk called `disk02-01` and `disk02-02`

\# **volsd dis disk02--01 disk02-01 disk02-02**

If the existing subdisk is associated with a plex before the `split` command, both of the resulting subdisks are associated with the same plex.

6.6.4 Changing Subdisk Information

To change subdisk information, enter:

\# **voledit set *field=value ... subdisk_name ...***

For example, to change the comment field of a subdisk called disk02-01, enter:

```
# voledit set comment="New comment" disk02-01
```

Use the voledit command to change the following subdisk fields:

- The putil *[n]* fields
- The tutil *[n]* fields
- The len field (only if the subdisk is disassociated)

Table 6–3 describes these subdisk fields.

Table 6–3: The putil and util Fields

Field	Description
putil0	Reserved for use by the LSM utilities and is retained after a reboot.
putil1	Reserved for use by high-level utilities, such as the Visual Administrator interface (dxlsm) and the LSM Support Operations interface (voldiskadm). This field is retained after a reboot.
putil2	Reserved for use by the system administrator or site-specific applications. This field is retained after a reboot.
tutil0	Reserved for use by the LSM utilities and is cleared after a reboot.
tutil1	Reserved for use by high-level utilities, such as dxlsm and voldiskadm. This field is cleared after a reboot.
tutil2	Reserved for use by the system administrator or site-specific applications. This field is cleared after a reboot.

Note

Entering data in the putil0 field prevents you from using the subdisk as part of a plex, if it is not already being so used.

6.6.5 Moving Subdisks

You can move portions of a volume 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 RAID5 volume to a disk that already contains a copy or part of that volume.

If this task fails and leaves some unused subdisks (that is, subdisks that are not associated with a volume) on the system, you can use the Remove Subdisk task to free the space occupied by the unused subdisks.

You must ensure that the following conditions are met before you move a subdisk:

- Both source and destination subdisks must be the same size.
- The subdisk being moved must be part of an active plex on an active (ENABLED) volume. To display the status of subdisks, enter:

  ```
  # volprint -l
  ```

- The new subdisk must not be associated with any other plex.

To move a subdisk, enter:

```
# volsd mv old_subdisk_name new_subdisk_name
```

6.6.6 Removing a Subdisk

You can remove a subdisk that is not associated with a volume. This returns the disk space occupied by unused subdisks to the free space pool.

To remove a subdisk you must disassociate the subdisk from a plex, then remove it. To disassociate and remove a subdisk, enter:

```
# volsd -orm dis subdisk_name
```

For example, to disassociate and remove a subdisk called `disk02-01`, enter:

```
# volsd -orm dis disk02-01
```

7

LSM Preventative Maintenance Procedures

LSM-related preventative maintenance procedures enable you to restore your LSM configuration if a disk or system fails. Preventative maintenance procedures include using the redundancy features built into LSM, backing up your system regularly, backing up copies of critical data needed to reconstruct your LSM configuration, and monitoring the LSM software.

This chapter describes LSM-related preventative maintenance procedures that you should perform.

7.1 The LSM Hot-Sparing Feature

The LSM software supports hot-sparing that enables you to configure a system to automatically react to failures on mirrored or RAID5 LSM objects.

With hot-sparing enabled, LSM detects failures on LSM objects and relocates the affected subdisks to designated spare disks or free disk space within the disk group. LSM then reconstructs the LSM objects that existed before the failure and makes them redundant and accessible.

When a partial disk failure occurs (that is, a failure affecting only some subdisks on a disk), redundant data on the failed portion of the disk is relocated and the existing volumes comprised of the unaffected portions of the disk remain accessible.

By default, hot-sparing is disabled. To enable hot-sparing enter the following command:

```
# volwatch -s
```

Note

Only one `volwatch` daemon can run on a system or cluster node at any time.

7.1.1 Hot-Spare Overview

The volwatch daemon monitors for failures involving LSM disks, plexes, or RAID5 subdisks. When such a failure occurs, volwatch daemon:

- Detects LSM events resulting from the failure of an LSM disk, plex, or RAID5 subdisk.
- Sends electronic mail to the root account (and other specified accounts) with notification about the failure and identifies the affected LSM objects.
- If hot-spare support is enabled, determines which subdisks to relocate, finds space for those subdisks in the disk group, relocate the subdisks, and notifies the root account (and other specified accounts) of these actions and their success or failure.

  When a partial disk failure occurs (that is, a failure affecting only some subdisks on a disk), redundant data on the failed portion of the disk is relocated and the existing volumes comprised of the unaffected portions of the disk remain accessible.

Note

Hot-sparing is only performed for redundant (mirrored or RAID5) subdisks on a failed disk. Non-redundant subdisks on a failed disk are not relocated, but you are notified of the failure.

Hot-sparing does not guarantee the same layout of data or the same performance after relocation. You may want to make some configuration changes after hot-sparing occurs.

7.1.1.1 Mail Notification

When an exception event occurs, the volwatch command uses mailx(1) to send mail to:

- The root account
- The user accounts specified when you use the rcmgr command to set the VOLWATCH_USERS variable in the /etc/rc.config.common file. See the rcmgr(8) reference page for more information on the rcmgr command.
- The user account that you specify on the command line with the volwatch command.

There is a 15 second delay before the failure 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. The following is a sample mail notification when a failure is detected:

```
Failures have been detected by the Logical Storage Manager:

failed disks:

medianame
 …

failed plexes:

plexname

 …

failed log plexes:

plexname

 …

failing disks:

medianame
 …

failed subdisks:

subdiskname

 …

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

The following list describes the sections of the mail message:

- The `medianame` list under `failed disks` specifies disks that appear to have completely failed;
- The `medianame` list 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 medianame list appears under both failed disks: and failing disks.
- The `plexname` list under `failed plexes` shows plexes that are detached due to I/O failures experienced while attempting to do I/O to subdisks they contain.
- The `plexname` list under `failed log plexes` indicates RAID5 or dirty region log (DRL) plexes that have experienced failures. The subdiskname list specifies subdisks in RAID5 volumes that are detached due to I/O errors.

If a disk fails completely, the mail message lists the disk that has failed and all plexes that use the disk. For example:

```
To: root
Subject: Logical Storage Manager failures on mobius.lsm.com

Failures have been detected by the Logical Storage Manager

failed disks:
 disk02

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

failing disks:
  disk02
```

This message shows that disk02 was detached by a failure, and that plexes home-02, src-02, and mkting-01 were also detached (probably due to a disk failure).

If plex or disk is detached by a failure, the mail sent lists the failed objects. If a partial disk failure occurs, the mail identifies the failed plexes. For example, if a disk containing mirrored volumes experiences a failure, a mail message similar to the following is sent:

```
To: root
Subject: Logical Storage Manager failures on mobius.lsm.com

Failures have been detected by the Logical Storage Manager:

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

To determine which disks are causing the failures in this message, enter:

volstat -sff home-02 src-02

This produces output such as the following:

```
                            FAILED
TYP NAME                 READS    WRITES
sd  disk01-04                0         0
sd  disk01-06                0         0
sd  disk02-03                1         0
sd  disk02-04                1         0
```

This display indicates that the failures are on disk02 and that subdisks disk02-03 and disk02-04 are affected.

Hot-sparing automatically relocates the affected subdisks and initiates necessary recovery procedures. However, if relocation is not possible or the hot-sparing feature is disabled, you must investigate the problem and recover the plexes. For example, sometimes these errors are caused by cabling failures. Check at the cables connecting your disks to your system. If there are any obvious problems, correct them and recover the plexes with the following command:

```
# volrecover -b volhome volsrc
```

This command starts a recovery of the failed plexes in the background (the command returns before the operation is done). If an error message appears, or if the plexes become detached again, you must replace the disk.

7.1.2 Initailizing Spare Disks

To use hot-sparing, you should configure disks as a spare, which identifies the disk as an available site for relocating failed subdisks. The LSM software does not use disks that are identified as spares for normal allocations unless you explicitly specify otherwise. This ensures that there is a pool of spare disk space available for relocating failed subdisks.

Spare disk space is the first space used to relocate failed subdisks. However, if no spare disk space is available or if the available spare disk space is not suitable or sufficient, free disk space is used.

You must initialize a spare disk and placed it in a disk group as a spare before it can be used for replacement purposes. If no disks are designated as spares when a failure occurs, LSM automatically uses any available free disk space in the disk group in which the failure occurs. If there is not enough spare disk space, a combination of spare disk space and free disk space is used.

When hot-sparing selects a disk for relocation, it preserves the redundancy characteristics of the LSM object to which the relocated subdisk belongs. For example, hot-sparing ensures that subdisks from a failed plex are not relocated to a disk containing a mirror of the failed plex. If redundancy cannot be preserved using available spare disks and/or free disk space, hot-sparing does not take place. If relocation is not possible, mail is sent indicating that no action was taken.

When hot-sparing takes place, the failed subdisk is removed from the configuration database and LSM takes precautions to ensure that the disk space used by the failed subdisk is not recycled as free disk space.

Follow these guidelines when choosing a disk to configuring as a spare:

- The hot-spare feature works best if you specify at least one spare disk in each disk group containing mirrored or RAID5 volumes.

- If a given disk group spans multiple controllers and has more than one spare disk, set up the spare disks on different controllers (in case one of the controllers fails).
- For hot-sparing to succeed for a mirrored volume, the disk group must have at least one disk that does not already contain one of the volume's mirrors. This disk must be a spare disk with some available space or a regular disk with some free space.
- For hot-sparing to succeed for a mirrored and striped volume, the disk group must have at least one disk that does not already contain one of the volume's mirrors or another subdisk in the striped plex. This disk should either be a spare disk with some available space or a regular disk with some free space.
- For hot-sparing to succeed for a RAID5 volume, the disk group must have at least one disk that does not already contain the volume's RAID5 plex or one of its log plexes. This disk should either be a spare disk with some available space or a regular disk with some free space.
- If a mirrored volume has a DRL log subdisk as part of its data plex (for example, `volprint` does not list the plex length as `LOGONLY`), that plex cannot be relocated. Therefore, place log subdisks in plexes that contain no data (log plexes). By default, the `volassist` command creates log plexes.
- Hot-sparing is capable of creating a new mirror of the root disk if the root disk is mirrored and it fails. The `rootdg` disk group should contain an empty spare disk that satisfies the restrictions for mirroring the root disk.
- Although it is possible to build LSM objects on spare disks, it is preferable to use spare disks for hot-sparing only.
- When relocating subdisks off a failed disk, LSM attempts to use a spare disk large enough to hold all data from the failed disk.

To initialize a disk as a spare that has no associated subdisks, use the `voldiskadd` command and enter **y** at the following prompt:

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

To initialize an existing LSM disk as a spare disk, enter:

```
# voledit set spare=on medianame
```

For example, to initialize a disk called `test03` as a spare disk, enter:

```
# voledit set spare=on test03
```

To remove a disk as a spare, enter:

```
# voledit set spare=off medianame
```

For example, to make a disk called `test03` available for normal use, enter:

```
# voledit set spare=off test03
```

7.2 Replacement Procedure

In the event of a disk failure, mail is sent, and if `volwatch` was configured to run with hot sparing support with the `-s` option, `volwatch` attempts to relocate any subdisks that appear to have failed. This involves finding appropriate spare disk or free disk space in the same disk group as the failed subdisk.

To determine which disk from among the eligible spare disks to use, `volwatch` tries to use the disk that is closest to the failed disk. The value of closeness depends on the controller, target, and disk number of the failed disk. For example, a disk on the same controller as the failed disk is closer than a disk on a different controller; a disk under the same target as the failed disk is closer than one under a different target.

If no spare or free disk space is found, the following mail message is sent explaining the disposition of volumes on the failed disk:

```
Relocation was not successful for subdisks on disk dm_name
in volume v_name in disk group dg_name.
No replacement was made and the disk is still unusable.

The following volumes have storage on medianame:

volumename
…

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.
```

If non-RAID5 volumes are made unusable due to the failure of the disk, the following is included in the mail message:

```
The following volumes:

volumename
…

have data on medianame 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.
```

If RAID5 volumes are made unavailable due to the disk failure, the following message is included in the mail message:

```
The following RAID5 volumes:
```

```
volumename
…

have storage on medianame 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.
```

If spare disk space is found, LSM attempts to set up a subdisk on the spare disk and use it to replace the failed subdisk. If this is successful, the `volrecover` command runs in the background to recover the contents of data in volumes on the failed disk.

If the relocation fails, the following mail message is sent:

```
Relocation was not successful for subdisks on disk dm_name in
volume v_name in disk group dg_name.
No replacement was made and the disk is still unusable.

error message
```

If any volumes are rendered unusable due to the failure, the following is included in the mail message:

```
The following volumes:

volumename
…

have data on dm_name 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.
```

If the relocation procedure completes successfully and recovery is under way, the following mail message is sent:

```
Volume v_name Subdisk sd_name relocated to newsd_name,
but not yet recovered.
```

Once recovery has completed, a message is sent relaying the outcome of the recovery procedure. If the recovery was successful, the following is included in the mail message:

```
Recovery complete for volume v_name in disk group dg_name.
```

If the recovery was not successful, the following is included in the mail message:

```
Failure recovering v_name in disk group dg_name.
```

7.2.1 Moving Relocated Subdisks

When hot-sparing occurs, subdisks are relocated to spare disks or available free disk space within the disk group. The new subdisk locations may not provide the same performance or data layout that existed before hot-sparing took place. After a hot-spare procedureis completed, you may want to move the relocated subdisks to improve performance, to keep the spare disk space free for future hot-spare needs, or to restore the configuration to its previous state.

- Note the characteristics of the subdisk before relocation. This information is available from the mail messages sent to root. For example, look for a mail 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.
```

- Note the new location for the relocated subdisk For example, look for a mail 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.
```

- Fix or replace the disk that experienced the failure using the procedures described in Section 8.1.
- Move the relocated subdisk to the desired location.

Note

The RAID5 volumes are not redundant while you move the subdisk.

For example, to move the relocated subdisk `disk05-01` back to the original disk `disk-02`, enter:

```
# volassist move volhome !disk05 disk02
```

7.3 Save the LSM Configuration

It is recommended that you use the `volsave` command to create a copy the current LSM configuration on a regular basis. You can use the `volrestore` command to recreate the configuration if the disk group's configuration is lost.

The `volsave` command only saves information about the LSM configuration; it does not save:

- Data in LSM volumes. To ensure that you can recover the saved configuration, you must back up the volume on which it resides.

- Configuration information for volumes associated with the boot disk. After the rootdg disk group is restored, you must reencapsulate the boot disk partitions as described in Chapter 4.

The volsave command saves information about an LSM configuration in a set of files called a **description set**, which is stored by default in a time stamped directory in /usr/var/lsm/db.

When you run volsave, a description set is created, which consists of the following files:

- allvol.DF

  A volmake description file for all volumes, plexes, and subdisks in a disk group. The volsave command creates a separate subdirectory and description file for each disk group on the system.

- voldisk.list

  A description of the disks. This file is the output of the voldisk list command.

- volboot

  The contents of the /etc/vol/volboot file.

- header

  A header file for the description set, containing a checksum, a magic number, the date of the file's creation, and the version of the volsave command.

To create a backup copy of the current LSM configuration using the default backup directory and verify the LSM configuration information in the default directory, enter:

volsave

Output similar to the following is displayed:

```
LSM configuration being saved to
   /usr/var/lsm/db/LSM.19991226203620.skylark

volsave does not save configuration for volumes used for
root, swap, /usr or /var.
LSM configuration for following system disks not saved:
dsk8a dsk8b

LSM Configuration saved successfully.
```

To verify the save, enter:

cd /usr/var/lsm/db/LSM.19991226203620.skylark

ls

Output similar to the following is displayed:

```
dg1.d     header  volboot dg2.d    rootdg.d  voldisk.list:
```

In this example, the `volsave` command created the following files and directories:

- A time stamped subdirectory, `LSM.19991226203620.skylark`, containing the `header`, `volboot`, and `voldisk.list` description files
- A *diskgroup*`.d` subdirectory for each of the system's three disk groups, `dg1`, `dg2`, and `rootdg`.
- An `allvol.DF` file in each of the *diskgroup*`.d` subdirectories. This file is a `volmake` description file for all volumes, plexes, and subdisks in that disk group.

To save the LSM configuration in a timestamped subdirectory in a directory other than `/usr/var/config`, enter:

```
# volsave -d /usr/var/config/LSM.%date
```

Output similar to the following is displayed:

```
LSM configuration being saved to
    /usr/var/config/LSM.19991226203658
:
LSM Configuration saved successfully.
```

To save an LSM configuration to a specific directory called `/usr/var/LSM.config1`, enter:

```
# volsave -d /usr/var/LSM.config1
```

Output similar to the following is displayed:

```
   LSM configuration being saved to /usr/var/LSM.config1
:
  LSM Configuration saved successfully.
```

7.4 Back Up Volumes

Backing up a volume requires a mirrored plex that is at least large enough to store the complete contents of the volume. Using a smaller plex results in an incomplete copy.

Note

Use the AdvFS backup utilities to backup volumes used with AdvFS instead of the LSM methods.

The methods described in this section do not apply to RAID5 volumes.

7.4.1 Backing Up A Non-Mirrored Volume

Follow these steps to backup a non-mirrored volume:

1. Ensure there is enough free disk space to create another plex for the volume that you want to back up. You can determine this by comparing the output of the `voldg free` command for the disk group and the `volprint -vt` command for the volume.
2. Create a snapshot plex by entering the following command:

   \# **volassist snapstart *volume_name***

   For example, to create a snapshot plex for the volume called `vol01`, enter:

   \# **volassist snapstart vol01**

   This creates a write-only backup plex that is attached to and synchronized with the specified volume.
3. When the snapstart operation is complete, the plex state changes to `SNAPDONE`. You can then complete the snapshot operation. Select a convenient time and inform users of the upcoming snapshot. Warn them to save files and refrain from using the system briefly during that time.

   When you are ready to create the snapshot, make sure there is no activity on the volume. For UFS volumes, It is recommended that you unmount the file system briefly to ensure that the snapshot data on disk is consistent and complete (all cached data has been flushed to the disk).
4. Create a snapshot volume that reflects the original volume by entering the following command:

   \# **volassist snapshot *volume_name temp_volume name***

   For example to create temporary volume called `vol01-temp` for a volume called `vol01`, enter:

   \# **volassist snapshot vol01 vol01-temp**

   This operation detaches the finished snapshot (which becomes a normal plex), creates a new normal volume, and attaches the snapshot plex to it. The snapshot then becomes a normal, functioning plex with a state of ACTIVE. At this point, you can mount and resume normal use of the volume.

5. Check the temporary volume's contents. For example, to check the UFS file system on a volume called `vol01--temp`, enter:

   ```
   # fsck -p /dev/rvol/rootdg/vol01-temp
   ```

6. Perform the backup by entering the following command:

   ```
   # dump 0 /dev/rvol/disk-group/temp_volume_name
   ```

   For example, to back up a volume called `vol01--temp` in the `rootdg` disk group, enter:

   ```
   # dump 0 /dev/rvol/rootdg/vol01-temp
   ```

7. After the backup is completed, remove the temporary volume by entering the following commands:

   ```
   # volume stop temp_volume_name
   # voledit -r rm temp_volume_name
   ```

   For example to remove a temporary volume called `vol01--temp`, enter:

   ```
   # volume stop vol01-temp
   # volume -r rm vol01-temp
   ```

7.4.2 Backing Up A Mirrored Volume

If a volume is mirrored, you can back up the volume by temporarily dissociating one of the plexes from the volume. This method eliminates the need for extra disk space for the purpose of backup only.

Warning

If the volume has only two plexes, redundancy is not available during the time of the backup.

Follow these steps to back up a mirrored volume:

1. Stop all activity on the volume. For UFS volumes, it is recommended that you unmount the file system briefly to ensure that the data on disk is consistent and complete (all cached data has been flushed to the disk).
2. Dissociate one of the volume's plexes by entering the following command:

   ```
   # volplex dis plex_name
   ```

   For example, to dissociate a plex called `vol01-02`, enter:

   ```
   # volplex dis vol01-02
   ```

This operation usually takes only a few seconds. It leaves the plex called `vol01--02` available as an image of the volume frozen at the time of dissociation.

3. Mount and resume normal use of the volume.
4. Create a temporary volume by entering the following commands:

```
# volmake -Ufsgen vol temp_volume_name plex=plex_name
# volume start temp_volume_name
```

   For example to create a temporary volume called `vol01--temp` using a plex called `vol01--02`, enter:

```
# volmake -Ufsgen vol vol01-temp plex=vol01-02
# volume start vol01-temp
```

5. Check the temporary volume by entering the following command:

```
# fsck -p /dev/rvol/rootdg/temp_volume_name
```

6. Perform the backup using the temporary volume by entering the following command:

```
# dump 0 /dev/rvol/rootdg/temp_volume_name
```

7. After the backup is completed, remove the temporary volume and reattach the plex by entering the following commands:

```
# volplex dis plex_name
# voledit -r rm temp_volume_name
# volplex att volume_name plex_name &
```

7.5 Collect LSM Performance Data

LSM provides two types of performance information — I/O statistics and I/O traces:

- I/O statistics are collected using the `volstat` command, which provides the most commonly-used information.
- I/O tracing are collected using the `voltrace` command, which provides advanced and in-depth information.

Note

In a cluster environment, `volstat` and `voltrace` report statistics relative to the node on which the commands are executed. These commands do not provide statistics for all the nodes within a cluster.

7.5.1 Gathering I/O Statistics

The volstat command provides access to information for activity on volumes, plexes, subdisks, and disks uses with the LSM software. You can use the volstat command to report I/O statistics for LSM objects during system boot time or for specified time intervals. Statistics for a specific LSM object or all objects are displayed. If a disk group is specified, statistics are displayed only for objects in that disk group; otherwise, statistics for the default disk group (rootdg) are displayed.

The amount of information displayed depends on the options you specified with the volstat command. You can also reset the statistics information to zero, which is useful for measuring the impact of a particular operation. For information on available options, see the volstat(8) reference page.

7.5.1.1 Statistics Recorded by LSM

The LSM software records the following three I/O statistics:

- A count of read and write operations.
- The number of read and write blocks.
- The average operation time. This time reflects the total time it took to complete an I/O operation, including the time spent waiting in a disk's queue on a busy device.

The LSM software records these statistics for logical I/O for each volume. The statistics are recorded for the following types of operations: reads, writes, atomic copies, verified reads, verified writes, plex reads, and plex writes.

For example, one write to a two-plex volume results in at least five operations — one for each plex, one for each subdisk, and one for the volume. Similarly, one read that spans two subdisks shows at least four reads — one read for each subdisk, one for the plex, and one for the volume.

The LSM software also maintains other statistical data. For example, read and write failures that appear for each mirror, and corrected read and write failures for each volume accompany the read and write failures that are recorded.

To display statical data for volumes, enter:

```
# volstat
```

Output similar to the following is displayed:

```
OPERATIONS  BLOCKS        AVG TIME(ms)
TYP NAME     READ     WRITE      READ     WRITE   READ  WRITE
```

```
vol v1          3        72        40     62541    8.9   56.5
vol v2          0        37         0       592    0.0   10.5
```

To display statical data for all LSM objects, enter:

volstat -vpsd

Output similar to the following is displayed:

```
OPERATIONS   BLOCKS        AVG TIME(ms)
TYP NAME      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
```

To display statistics on volumes in the `rootdg` disk group in one second intervals, enter:

volstat -i 1

Output similar to the following is displayed:

```
OPERATIONS     BLOCKS        AVG TIME(ms)
TYP NAME         READ     WRITE      READ     WRITE   READ  WRITE

Mon Jun  8 15:11:16 1998
vol v1             3        72        40     62541    8.9   56.5
vol v2         14015        37     14015       592    0.3   10.5

Mon Jun  8 15:11:17 1998
vol v1             0         0         0         0    0.0    0.0
vol v2          2606         0      2606         0    0.3    0.0
```

To display error statistics on a volume called `testvol`, enter:

volstat -f cf testvol

Output similar to the following is displayed:

```
CORRECTED        FAILED
TYP NAME         READS   WRITES    READS   WRITES
```

```
vol testvol         1         0         0         0
```

Additional volume statistics are available for RAID5 configurations. See the volstat(8) reference page for details.

7.6 Monitor LSM Events and Configuration Changes

You use the volwatch and the volnotify commands to monitor LSM events and configuration changes.

The volwatch shell script sends mail to the root login (default) and other user accounts that you specify when certain LSM configuration events occur, such as a plex detach caused by a disk failure.

To specify another mail recipient or multiple mail recipients, use the rcmgr command to set the rc.config.common variable VOLWATCH_USERS. The LSM volwatch script uses VOLWATCH_USERS whenever the system is booted or LSM is restarted.

To specify a user named user1@mail.com as a mail recipient, enter:

```
# rcmgr -c set VOLWATCH_USERS root@dec.com user1@mail.com
```

LSM events are sent to the EVM Event Management system using volnotify command. The volnotify command integrates with EVM by default, and runs automatically when LSM starts.

The following LSM events are reported by the volnotify command within EVM:

- -i Display disk group import, deport, and disable events
- -f Display plex, volume, and disk detach events
- -d Display disk change events
- -c Displays disk group change events

While the LSM volnotify events reported within EVM are configured through the rc.config.common variable LSM_EVM_OPTS, the LSM_EVM_OPTS settings should not normally be changed because certain system software depend on these values for operation.

Note

For cluster environments, the volnotify command only reports LSM events that occur locally on that node. Therefore, to receive LSM events that occur anywhere within the cluster, do not disable the volnotify integration within EVM.

Subscribers can display LSM events through the LSM `volnotify` EVM template called `lsm.volnotify.evt`. This EVM template is used to display LSM events in cluster and non-cluster environments.

To display LSM events from the EVM log, enter:

```
# evmget -f "[name *.volnotify]" | evmshow -t \
"@timestamp @@"
```

To display LSM events in real time, enter:

```
# evmwatch -f "[name *.volnotify]" | evmshow -t \
"@timestamp @@"
```

For more information, see the `volnotify`(8), `volwatch`(8), and `EVM`(5) reference pages.

7.7 Monitor Volume States

You can use the `volprint` command to display volume information. The `volprint` command displays state information that indicates a variety of normal and exceptional conditions.

7.7.1 Volume States

Volume states indicate whether or not the volume is initialized, written to, and the accessibility of the volume. Table 7–1 describes the volume states.

Table 7–1: Volume States

State	Means
EMPTY	The volume contents are not initialized. The kernel state is always DISABLED when the volume is EMPTY.
CLEAN	The volume is not started (kernel state is DISABLED) and its plexes are synchronized.
ACTIVE	The volume was started (kernel state is currently ENABLED) or was in use (kernel state was ENABLED) when the machine was rebooted. If the volume is currently ENABLED, the state of its RAID-5 plex at any moment is not certain (since the volume is in use). If the volume is currently DISABLED, this means that the plexes cannot be guaranteed to be consistent, but are made consistent when the volume is started.

Table 7–1: Volume States (cont.)

SYNC	The volume is either in read-writeback recovery mode (kernel state is currently ENABLED) or was in read-writeback mode when the machine was rebooted (kernel state is DISABLED). With read-writeback recovery, plex consistency is recovered by reading data from blocks of one plex and writing the data to all other writable plexes. If the volume is ENABLED, this means that the plexes are being resynchronized via the read-writeback recovery. If the volume is DISABLED, it means that the plexes were being resynchronized via read-writeback when the machine rebooted and therefore, still need to be synchronized.
NEEDSYNC	The volume requires a resynchronization operation the next time it starts.

The interpretation of these states during volume startup is modified by the persistent state log for the volume (for example, the DIRTY/CLEAN flag). If the clean flag is set, this means that an ACTIVE volume was not written to by any processes or was not even open at the time of the reboot; therefore, it is considered CLEAN. The clean flag is always set when the volume is marked CLEAN.

7.7.2 RAID5 Volume States

RAID5 volumes have their own set of volume states as described in Table 7–2.

Table 7–2: RAID5 Volume States

State	Means
EMPTY	The volume contents are not initialized. The kernel state is always DISABLED when the volume is EMPTY.
CLEAN	The volume is not started (kernel state is DISABLED) and its parity is good. The RAID-5 plex stripes are consistent.
ACTIVE	The volume was started (kernel state is currently ENABLED) or was in use (kernel state was ENABLED) when the system was rebooted. If the volume is currently ENABLED, the state of its RAID-5 plex at any moment is not certain (since the volume is in use). If the volume is currently DISABLED, this means that the parity synchronization is not guaranteed.
SYNC	The volume is either undergoing a parity resynchronization (kernel state is currently ENABLED) or was having its parity resynchronized when the machine was rebooted (kernel state is DISABLED).

Table 7–2: RAID5 Volume States (cont.)

NEEDSYNC	The volume requires a parity resynchronization operation the next time it is started.
REPLAY	The volume is in a transient state as part of a log replay. A log replay occurs when it becomes necessary to use logged parity and data.

7.7.3 Volume Kernel States

The volume kernel state indicates the accessibility of the volume. The volume kernel state allows a volume to have an offline (DISABLED), maintenance (DETACHED), or online (ENABLED) mode of operation. Table 7–3 describes the volume kernel states.

Table 7–3: Volume Kernel States

State	Means
ENABLED	Read and write operations can be performed.
DISABLED	The volume is not accessed.
DETACHED	Read and write operations cannot be performed, but plex device operations and `ioctl` functions are accepted.

7.8 Monitor Plex States

You can use the `volprint` command to display plex information. The `volprint` command displays state information that indicates a variety of normal and exceptional conditions.

7.8.1 Plex States

Each data plex associated with a volume has its state set to one of the values listed in Table 7–4.

LSM utilities use plex states to:

- Monitor operations on plexes,
- Track whether a plex was in active use at the time of a system failure
- Indicate whether volume contents have been initialized to a known state
- Determine if a plex contains a valid copy (mirror) of the volume contents

Although the LSM utilities automatically maintain a plex's state, you may need to manually change the plex state. For example, if a disk begins to fail, you can temporarily disable a plex located on the disk.

Table 7–4: LSM Plex States

State	Means
EMPTY	The plex is not initialized. This state is set when the volume state is also EMPTY.
CLEAN	The plex was running normally when the volume was stopped. The plex was enabled without requiring recovery when the volume is started.
ACTIVE	The plex is running normally on a started volume. The plex condition flags (NODAREC, REMOVED, RECOVER, and IOFAIL) may apply if the system is rebooted and the volume restarted.
STALE	The plex was detached, either by the `volplex det` command or by an I/O failure. The `volume start` command changes the state for a plex to STALE if any of the plex condition flags are set. STALE plexes are reattached automatically by calling `volplex att` when a volume starts.
OFFLINE	The plex was disabled by the `volmend off` command. See `volmend(8)` for more information.
SNAPATT	This is a snapshot plex that is attached by the `volassist snapstart` command. When the attach is complete, the state for the plex is changed to SNAPDONE. If the system fails before the attach completes, the plex and all of its subdisks are removed.
SNAPDONE	This is a snapshot plex created by `volassist snapstart` command that fully attached. You can turn a plex in this state into a snapshot volume with the `volassist snapshot` command. If the system fails before the attach completes, the plex and all of its subdisks are removed. See volassist(8) for more information.
SNAPTMP	This is a snapshot plex that was attached by the `volplex snapstart` 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 snapshot plex created by `volplex snapstart` command that is fully attached. You can turn a plex in this state into a snapshot volume with `volplex snapshot` command. If the system fails before the attach completes, the plex is dissociated from the volume. See volplex(8) for more information.
TEMP	This is a plex that is associated and attached to a volume with the `volplex att` command. If the system fails before the attach completes the plex is dissociated from the volume.

Table 7–4: LSM Plex States (cont.)

State	Means
TEMPRM	This is a plex that is being associated and attached to a volume with `volplex att`. 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	This is a plex that is associated and attached to a volume with the `volplex att` command. If the system fails before the attach completes, the plex and its subdisks are dissociated from the volume and removed.

7.8.2 Plex State Cycle

During normal LSM operation, plexes automatically cycle through a series of states. At system startup, volumes are automatically started and the `volume start` operation makes all CLEAN plexes ACTIVE. If all goes well until shutdown, the volume-stopping operation marks all ACTIVE plexes CLEAN and the cycle continues.

Deviations from this cycle indicates abnormalities that the LSM software attempts to normalize, for example:

- If a crash occurs between startup and shutdown, the volume-starting operation may find a mirrored volume has no CLEAN plexes, only ACTIVE plexes. The mirrored volume is first placed in the NEEDSYNC state, then into SYNC state once resynchronization starts. After the plexes are resynchronized, the volume is placed into the ACTIVE state.
- If an I/O error occurs between startup and shutdown that causes a plex to become disabled, the volume-stopping operation marks that plex as STALE. Any STALE plexes require recovery. When the system restarts, data is copied from an ACTIVE to a STALE plex and makes the STALE plex ACTIVE.

7.8.3 Plex Kernel States

The plex kernel state indicates the accessibility of the plex. The plex kernel state is monitored in the volume driver and allows a plex to have an offline (DISABLED), maintenance (DETACHED), or online (ENABLED) mode of operation. No user intervention is required to set these states; they are maintained internally. On a system that is operating properly, all plexes are set to ENABLED.

Table 7–5 describes the plex kernel states.

Table 7–5: Plex Kernel States

State	Means
ENABLED	A write request to the volume will be reflected to the plex, if the plex is set to ENABLED for write mode. A read request from the volume is satisfied from the plex if the plex is set to ENABLED.
DISABLED	The plex is not accessed.
DETACHED	A write to the volume is not reflected to the plex. A read request from the volume will never be satisfied from the plex device. Plex operations and `ioctl` functions are accepted.

7.9 Monitor LSM Daemons

The `vold` and `voliod` daemons must be running for the LSM software to properly work. These daemons are normally started automatically when the system boots.

To determine the state of the volume daemon, enter:

```
# voldctl mode
```

The following table shows messages that may display and possible actions to take if `vold` is disabled or not running.

Message from voldctl mode	Status of vold	How to change
`mode:enabled`	Running and enabled	—
`mode:disabled`	Running, but disabled	**voldctl enable**
`mode:not-running`	Not running	**vold**

See the `vold`(8) reference page for more information on the `vold` daemon.

The volume extended I/O daemon (`voliod`) allows for some extended I/O operations without blocking calling processes. The correct number of `voliod` daemons is automatically started when LSM is started. There are typically several `voliod` daemons running at all times. It is recommended that you run at least one `voliod` daemon for each processor on the system.

Follow these steps to check and/or change the `voliod` daemons:

1. Display the current `voliod` state by entering the following command:

   ```
   # voliod
   ```

   This is the only method for checking on `voliod`, because the `voliod` processes are kernel threads and are not visible as output of the `ps` command.

Output similar to the following may display:

```
0 volume I/O daemons running
```

2. If no voliod daemons are running, or if you want to change the number of daemons, enter the following command:

   ```
   # voliod set n
   ```

   Where n is the number of I/O daemons. Set the number of LSM I/O daemons, which is either two or the number of central processing units (CPUs) on the system, whichever is greater. For example, on a single CPU system, enter:

   ```
   # voliod set 2
   ```

   On a four CPU system, enter:

   ```
   # voliod set 4
   ```

3. Verify the change by entering the following command:

   ```
   # voliod
   ```

   Output similar to the following should display:

   ```
   2 volume I/O daemons running
   ```

See the voliod(8) reference page for more information on the voliod daemon.

7.10 Trace LSM I/O Operations

You use the voltrace command trace volume operations. Using the voltrace command, you can set I/O tracing masks against a group of volumes or to the system as a whole. You can then use the voltrace command to display ongoing I/O operations relative to the masks.

The trace records for each physical I/O show a volume and buffer-pointer combination that enables you to track each operation even though the traces may be interspersed with other operations. Like the I/O statistics for a volume, the I/O trace statistics include records for each physical I/O done, and a logical record that summarizes all physical records. For additional information, see the voltrace(8) reference page.

To trace volumes, enter:

```
# voltrace -l
```

Output similar to the following is displayed:

```
926159 598519126 START read vdev v2 dg rootdg dev 40,6 block 895389 len 1 concurrency 1 pid 3943
926159 598519127 END read vdev v2 dg rootdg op 926159 block 895389 len 1 time 1
926160 598519128 START read vdev v2 dg rootdg dev 40,6 block 895390 len 1 concurrency 1 pid 3943
926160 598519128 END read vdev v2 dg rootdg op 926160 block 895390 len 1 time 0
```

8

Troubleshooting the LSM Software

This chapter provides information that helps you troubleshoot the LSM software.

8.1 Recovering from a Disk Failure

LSM's hot-sparing feature automatically detects disk failures and notifies you of the failures by an electronic mail. If hot-sparing is disabled or you miss the electronic mail, you may notice disk failures through the output of the `volprint` command to look at the status of the disks. You may also see driver error messages on the console or in the system messages file. See Section 7.1 for more information on the LSM hot-sparing feature.

If one plex of a volume encounters a disk I/O failure (for example, because the disk has an uncorrectable format error), LSM disables the disk and does not retry the same I/O to that disk. However, before disabling the disk, LSM attempts to correct errors on the plex.

For read errors, LSM logs the error, and then attempts to read from other plexes. If the data is read successfully, LSM tries to correct the read error by writing the data back to the original plex.

If the write is successful, the data is returned, and message similar to the following is displayed:

```
Dec 4 17:27:32 xebec vmunix: io/vol.c(volerror):
Correctable read error on vol-dsk10g...
```

To display more information on a volume called `vol-dsk10g`, enter:

```
# volstat -f cf vol-dsk10g
```

Output similar the following is displayed:

```
                             CORRECTED            FAILED
TYP NAME                  READS    WRITES       READS    WRITES
 vol vol-dsk10g             1         0           0         0
```

If the write fails, the bad plex is detached, I/O stops on that plex, but continues on the remaining plexes of the volume and the following message is written to the LSM kernel change log to record that the plex is detached and not used:

```
Dec  4 18:42:31 xebec vmunix: io/vol.c(volerror): Uncorrectable read error on vol...
Dec  4 18:42:31 xebec vmunix: voliod_error: plex detach - volume vol-dsk10g,
                                            plex vol-dsk10g-02...
```

To display more information, enter:

```
# volprint -ht vol-dsk10g
```

Output similar to the following is displayed:

```
v  vol-dsk10g      gen             ENABLED  ACTIVE   819200   PREFER
pl vol-dsk10g-01   vol-dsk10g      ENABLED  ACTIVE   819200   CONCAT
sd dsk10g-01       vol-dsk10g-01   0        0        819200   dsk10g
pl vol-dsk10g-02   vol-dsk10g      DISABLED NODEVICE 819200   CONCAT
sd sd-dsk8g-1      vol-dsk10g-02   1        1        819199   dsk8
```

A disk failure on a mirrored volume results ina console error similar to the following:

```
Dec  5 10:44:37 xebec vmunix: io/vol.c(volerror): Uncorrectable read error on vol...
Dec  5 10:44:37 xebec vmunix: io/vol.c(volerror): Uncorrectable read error on vol...
Dec  5 10:44:37 xebec vmunix: voliod_error: plex detach - volume vol-dsk10g,
                                            plex vol-dsk10g-02...
```

To display more information, enter:

```
# volstat -f cf vol-dsk10g
```

Output similar to the following is displayed:

```
                           CORRECTED              FAILED
TYP NAME                READS    WRITES        READS    WRITES
vol vol-dsk10g            1         0            1         0
```

Or, enter:

```
# volprint -ht vol-dsk10g
```

Output similar to the following is displayed:

```
v  vol-dsk10g    gen           ENABLED  ACTIVE   819200   PREFER
pl vol-dsk10g-01 vol-dsk10g    ENABLED  ACTIVE   819200   CONCAT
sd dsk10-01      vol-dsk10g-03 0        0        819200   dsk10
pl vol-dsk10g-02 vol-dsk10g    DETACHED IOFAIL   819200   CONCAT
sd sd-dsk8g-1    vol-dsk10g-02 1        1        819199   dsk8
```

For write errors, if the disk is still mirrored, the bad plex is detached and a message written to the LSM kernel change log to record that the plex is detached and is no longer used.

If the write succeeded on at least one plex in the volume, the write is considered successful.

If the write failed to all plexes, LSM returns a failure error and detaches the disk from its disk group.

8.1.1 Replacing a Disk that is Beginning to Fail

Often a disk has recoverable (soft) errors before it fails completely. If a disk is getting an unusual number of soft errors, use the following procedure to replace it.

1. Detach the disk from its disk group by running `voldiskadm` and choosing `Remove a disk for replacement` from the main menu.

   If there are initialized disks available as replacements, you can specify the disk as part of this operation. Otherwise, you must specify the replacement disk later by choosing `Replace a failed or removed disk` from the main menu.

   When you select a disk to remove for replacement, all volumes that will be affected by the operation are displayed. For example, the following output might be displayed:

   ```
   The following volumes will lose mirrors as a result of this
   operation:

        home src

   No data on these volumes will be lost.

   The following volumes are in use, and will be disabled
   as a result of this operation:

        mkting

   Any applications using these volumes will fail future
   accesses. These volumes will require restoration from
   backup.

   Are you sure you want do this? [y,n,q,?] (default: n)
   ```

2. If any volumes would be disabled, quit from `voldiskadm` and save the volume. Either back up the volume or move the volume off of the disk.

   For example, to move the volume `mkting` to a disk other than `disk02`, enter the following command:

   ```
   # volassist move mkting disk02
   ```

   After the volume is backed up or moved, enter the `voldiskadm` command again and continue to remove the disk for replacement.

3. After the disk is removed, specify a replacement disk by choosing `Replace a failed or removed disk` from the main menu in `voldiskadm` menu interface.

8.1.2 Replacing a Failed Disk

If a disk that was in use by LSM fails completely and is detached, you can replace the disk with a new disk. To replace a disk, enter the `voldiskadm` command and choose Replace a failed or removed disk from the main menu.

If you have any disks that are initialized for LSM but have never been added to a disk group, you can select one of those disks as a replacement. Do not choose the old disk drive as a replacement even though it may appear in the selection list. If there are no suitable initialized disks, you can choose to initialize a new disk.

If a disk failure caused a volume to be disabled, the volume must be restored from backup after the disk is replaced. To identify volumes that wholly reside on disks that were disabled by a disk failure, use the volinfo command. Any volumes that are listed as Unstartable must be restored from backup.

To display disk status, enter:

```
# volinfo
```

Output similar to the following is displayed:

```
home           fsgen    Started
mkting         fsgen    Unstartable
src            fsgen    Started
```

To restart the Unstartable volume called mkting, enter:

```
# volume -o bg -f start mkting
```

The -o bg option recovers plexes as a background task.

8.1.3 Reattaching Disks

A disk reattach operation may be necessary if a disk has experienced a full failure and hot-sparing is not possible, or if LSM is started with some disk drivers unloaded and unloadable (causing disks to enter the failed state). If the problem is fixed, it may be possible to use the volreattach command to reattach the disks without plexes being flagged as stale, as long as the reattach happens before any volumes on the disk are started.

The volreattach command is called as part of disk recovery from the voldiskadm menus. If possible, volreattach will reattach the failed disk media record to the disk with the same device name in the disk group in which it was located before and will retain its original disk media name. After a reattach takes place, recovery may or may not be necessary. The reattach may fail if the original (or another) cause for the disk failure still exists.

To check whether a reattach is possible, enter:

```
# volreattach -c
```

This displays the disk group and disk media name where the disk can be reattached, without performing the operation.

For more information, see the `volreattach`(8) reference page.

8.2 Recovering from a Boot Disk Failure

When the boot disk is mirrored, 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 the plex on the boot disk.
- Experience slow performance (depending on the problem encountered with the disk containing one of the plexes in the `root` or `swap` volumes).

To reboot the system before the original boot disk is repaired, you can boot from any disk that contains a valid `root` volume.

If all copies of `rootvol` are corrupted, and you cannot boot the system, you must reinstall the system.

Replacing a boot disk is a more complex process than replacing other disks because boot-critical data must be placed in specific areas on specific disks in order for the boot process to find it. How you replace a failed boot disk depends on:

- If you've mirrored the root disk and enabled hot-sparing support.
- If the errors are correctable and the same disk can be re-used. This is known as *re-adding* a disk. If you reuse the boot disk, you should monitor it and replace it during your next maintenance cycle.
- If the disk has completely failed and must be replaced.

The sections that follow give instructions for re-adding or replacing the boot disk, as well as other information related to boot disk recovery.

8.2.1 Hot-Sparing and Boot Disk Failures

If the boot disk fails on a system that has the boot (root) disk mirrored and the hot-sparing feature enabled, LSM automatically attempts to replace the failed root disk mirror with a new mirror. To do this, a surviving mirror of the root disk is used to create a new mirror on either a spare disk or a disk with sufficient free space. This ensures that there are always at least two mirrors of the root disk that can be used for booting.

For hot-sparing to succeed, the `rootdg` disk group must have enough spare or free space to accommodate the volumes from the failed root disk. Also, the `rootvol` and `swapvol` volumes require contiguous disk space. If there is not enough contiguous space on a single new disk, each of these volumes can be relocated to a different new disk.

See Chapter 4 for more information on mirroring the boot disk.

8.2.2 Re-adding and Replacing Boot Disks

Normally, replacing a failed disk is as simple as putting a new disk somewhere on the controller and running LSM replace disk commands. It's even possible to move the data areas from that disk to available space on other disks, or to use a hot-spare disk already on the controller to replace the failure. For data that is not critical for booting the system, it doesn't matter where the data is located. All data that is not boot critical is only accessed by LSM after the system is fully operational. LSM can find this data for you. On the other hand, boot-critical data must be placed in specific areas on specific disks in order for the boot process to find it.

When a disk fails, there are two possible routes that can be taken to correct the action. If the errors are transient or correctable, then the same disk can be re-used. This is known as *re-adding* a disk. On the other hand, if the disk has truly failed, then it should be completely replaced.

8.2.2.1 Re-adding A Failed Boot Disk

Re-adding a disk is the same procedure as replacing a disk, except that the same physical disk is used. Usually, a disk that needs to be re-added has been *detached*, meaning that the LSM software has noticed that the disk has failed and has ceased to access it.

If the boot disk has a transient failure, its plexes can be recovered using the following steps. The `rootvol` and `swapvol` volumes can have two or three LSM disks per physical disk, depending on the layout of the original root disk.

1. To list the LSM disks that are associated with the failed physical disk, enter:

   ```
   # voldisk list
   ```

   Output similar to the following is displayed:

   ```
   DEVICE       TYPE      DISK      GROUP      STATUS
   dsk10        sliced     -          -         error
   dsk10b       nopriv     -          -         error
   dsk10f       nopriv     -          -        error
   dsk21        sliced    dsk21      rootdg     online
   dsk21b       nopriv    dsk21b     rootdg     online
   -             -        dsk10      rootdg    removed was:dsk10
   -             -        dsk10b     rootdg    removed was:dsk10b
   ```

```
-              -         dsk10f    rootdg   removed was:dsk10f
```

In this example, if dsk10 was the failed boot disk, then you can assume that dsk10, dsk10b, and dsk10f are the LSM disks associated with the physical disk dsk10.

2. Enter the following commands to add the LSM disks back to the rootdg disk group:

```
# voldisk online dsk10 dsk10b dsk10f
# voldg -k adddisk dsk10=dsk10
# voldg -k adddisk dsk10b=dsk10b
# voldg -k adddisk rootdsk10=dsk10f
```

3. Resynchronize the plexes in the rootvol and swapvol volumes:

```
# volrecover -sb rootvol swapvol
```

8.2.2.2 Replacing a Failed Boot Disk

Follow these steps to replace a failed boot disk under LSM control with a new disk:

1. Disassociate the plexes on the failed disk from rootvol and swapvol. Also, if /usr or /var were encapsulated on the boot disk, disassociate their plexes on the failed disk:

```
# volplex -o rm dis rootvol-02 swapvol-02 vol-dsk1g
```

2. Remove all LSM disks configured on the boot disk:

```
# voldg rmdisk dsk1a disk1b dsk1g dsk1f
# voldisk rm dsk1a dsk1b dsk1g dsk1f
```

3. Mirror the LSM volumes on the book disk onto the new disk, as described in Chapter 4. The replacement disk must have at least as much storage capacity as was in use on the old disk.

8.2.3 Stale or Unusable Plexes on the Boot Disk

If a disk is unavailable when the system is running, any plexes of volumes that reside on that disk will become stale, meaning the data on that disk is out of date relative to the other plexes of the volume.

During the boot process, the system accesses only one copy of the root volume (the copies on the boot disk) until a complete configuration for this volume can be obtained. If the plex of the root volume that was used for booting is stale, you must reboot the system from another boot disk that contains nonstale plexes. This problem can occur if the boot disk was replaced and restarted without adding the disk back into the LSM configuration. The system will boot normally, but the plexes that reside on the newly booted disk will be stale.

Another possible problem can occur if errors in the LSM headers on the boot disk prevents LSM from properly identifying the disk. In this case, LSM will be unable to know the name of that disk. This is a problem because plexes are associated with disk names, and therefore any plexes on that disk are unusable.

If either of these situations occurs, the LSM daemon `vold` will notice it when it is initializing the system as part of the `init` processing of the boot sequence. It will output a message describing the error, describe what can be done about it, and halt the system. For example, if the plex `rootvol-01` of the root volume `rootvol` on disk `disk01` of the system was stale, `vold` output is similar the following:

```
lsm:vold: Warning Plex rootvol-01 for root
volume is stale or unusable.

lsm:vold: Error: System boot disk does not have a valid root plex
Please boot from one of the following disks:

       Disk: disk02                    Device: dsk2

lsm:vold: Error: System startup failed
```

This informs you that the disk `disk02` contains usable copies of the root and swap plexes and should be used for booting and `dsk2` is the name of the system backup disk. When this message appears, you should reboot the system and boot from the device that corresponds to `dsk2`.

Once the system is booted, you must determine the problem. If the plexes on the boot disk were stale, they are caught up automatically as the system starts. If there is a problem with the private area on the disk, you must readd or replace the disk.

If the plexes on the boot disk were unavailable, you will receive mail from the `volwatch` command describing the problem.

To list that status of disks, enter:

```
# voldisk list
```

Output similar the following is displayed:

```
DEVICE       TYPE       DISK      GROUP      STATUS
-            -          disk02    rootdg     failed was:  dsk1
dsk2         sliced     disk02    rootdg     online
```

8.2.4 Failure To Obtain Crash Dumps

During a system crash or panic, a crash dump is temporarily saved to swap space. If the swap device is configured to use one or more LSM volumes, all the LSM swap volume's underlying disk partitions are used separately to maximize crash dump space, even when the swap volume was mirrored on different disk partitions. This does not cause any problems providing that the LSM mirrored swap are properly configured to not resynchronize the mirrors upon reboot because doing so could destroy the crash dump before it's saved to the file system.

If a mirrored swap volume is performing resynchronization upon reboot, you need to verify its configuration. If the secondary volumes (for example, LSM volumes other than swapvol) are performing resynchronization, this is probably due to the volume not being configured with start_opts=norecov option.

To check the start_opts option for a swap volume called v1, enter:

```
# volprint -m v1 | grep start_opts
```

Output similar to the following is displayed:

```
start_opts="
```

To change the start_opts option for a swap volume called v1, enter:

```
# volume set start_opts=norecov v1
```

To display the change, enter:

```
# volprint -m v1 | grep start_opts
```

Output similar to the following is displayed:

```
start_opts="norecov
```

If the LSM volume swapvol is performing resynchronization, this is typically because this volume does not have its device minor number set to 1. See Chapter 4 for information on how to setup root and swap volumes.

To check the swapvol volume's minor number, enter:

```
# ls -l /dev/*vol/swapvol
```

Output similar to the following is displayed:

```
crw-------   1 root     system    40,  1 Mar 16 16:00 /dev/rvol/swapvol
brw-------   1 root     system    40,  1 Mar 16 16:00 /dev/vol/swapvol
```

8.3 Resynchronizing Volumes

When storing data redundantly, using mirrored or RAID5 volumes, LSM takes necessary measures to ensure that all copies of the data match exactly. However, under certain conditions (usually due to complete system failures), small amounts of the redundant data on a volume can become inconsistent or unsynchronized. Aside from normal configuration changes (such as detaching and reattaching a plex), this can only occur when a system crashes while data is being written to a volume. Data is written to the mirrors of a volume in parallel, as is the data and parity in a RAID5 volume. If a system crash occurs before all the individual writes complete, it is possible for some writes to complete while others do not, resulting in the data becoming unsynchronized. For mirrored volumes, it can cause two reads from the same region of the volume to return different results if different mirrors are used to satisfy the read request. In the case of RAID5 volumes, it can lead to parity corruption and incorrect data reconstruction.

When LSM recognizes this situation, it needs to make sure that all mirrors contain exactly the same data and that the data and parity in RAID5 volumes match. This process is called volume resynchronization. Volumes that are part of disk groups that are automatically imported at boot time (such as `rootdg`) are resynchronized when the system boots.

Not all volumes require resynchronization after a system failure. Volumes that were never written or that were inactive when the system failure occurred and did not have any outstanding writes do not require resynchronization. LSM notices when a volume is first written and marks it as dirty. When a volume is closed by all processes or stopped cleanly, all writes will have completed and LSM removes the dirty flag for the volume. Only volumes that are marked dirty when the system reboots require resynchronization.

Resynchronization can be computationally expensive and can have a significant impact on system performance. The recovery process attempts to alleviate some of this impact by attempting to "spread out" recoveries to avoid stressing a specific disk or controller. Additionally, for very large volumes or for a very large number of volumes, the resynchronization process can take a long time. These effects can be addressed by using dirty-region logging for mirrored volumes, or by making sure that RAID5 volumes have valid RAID5 logs.

The exact process of resynchronization depends on the type of volume. RAID5 volumes that contain RAID5 logs can simply replay those logs. If no logs are available, the volume is placed in reconstruct-recovery mode and all parity is regenerated.

LSM automatically recovers mirrored and RAID5 volumes when the system is booted and the volumes are first started.

See the `volume`(8) reference page for more information on resynchronizing volumes.

8.4 Recovering Volumes

The following sections describe recovery procedures for problems relating to LSM volumes.

8.4.1 Listing Unstartable Volumes

An unstartable volume is likely to be incorrectly configured or has other errors or conditions that prevent it from being started. To display unstartable volumes, use the `volinfo` command, which displays information on the accessibility and usability of one or more volumes:

```
# volinfo -g disk_group [volume_name]
```

8.4.2 Recovering a Disabled Volume

If a system crash or an I/O error corrupts one or more plexes of a volume and no plex is CLEAN or ACTIVE, mark one of the plexes CLEAN and instruct the system to use that plex as the source for reviving the others. To place a plex in a CLEAN state, enter:

```
# volmend fix clean plex_name
```

For example, to place one plex called `vol01-02` in the CLEAN state, enter:

```
# volmend fix clean vol01-02
```

See the `volmend`(8) reference pages for more information.

8.5 Recovering RAID5 Volumes

RAID5 volumes are designed to remain available when a disk fails with a minimum of disk space overhead. However, many implementations of RAID5 can become vulnerable to data loss after a system failure, and some types of disk failures can also affect RAID5 volumes adversely. The following sections describe how system and disk failures affect RAID5 volumes, and the types of recovery needed.

8.5.1 System Failures and RAID5 Volumes

A system failure causes the data and parity in the RAID5 volume to become unsynchronized because the disposition of writes that were outstanding at the time of the failure cannot be determined. If this occurs while a RAID5 volume is being accessed, the volume is described as having stale parity. When this occurs, the parity must be reconstructed by reading all the non-parity columns within each stripe, recalculating the parity, and writing out the parity stripe unit in the stripe. This must be done for every stripe in the volume, so it can take a long time to complete.

Caution

While this resynchronization is going on, any failure of a disk within the array will cause the data in the volume to be lost. This only applies to RAID5 volumes *without* log plexes. Compaq recommends configuring all RAID5 volumes with a log.

Having the array vulnerable in this way is undesirable. Besides the vulnerability to failure, the resynchronization process can tax the system resources and slow down system operation.

RAID5 logs reduce the possible damage that can be caused by system failures. Because they maintain a copy of the data being written at the time of the failure, the process of resynchronization consists of simply reading that data and parity from the logs and writing it to the appropriate areas of the RAID5 volume. This greatly reduces the amount of time needed for a resynchronization of data and parity. It also means that the volume never becomes truly stale because the data and parity for all stripes in the volume is known at all times, so the failure of a single disk cannot result in the loss of the data within the volume.

8.5.2 Disk Failures and RAID5 Volumes

A RAID5 disk failure can occur due to an uncorrectable I/O error during a write to the disk (which causes the subdisk to be detached from the array) or due to a disk being unavailable when the system is booted (such as from a cabling problem or having a drive powered down). When this occurs, the subdisk cannot be used to hold data and is considered *stale* and *detached*. If the underlying disk becomes available or is replaced, the subdisk is considered stale and is not used.

If an attempt is made to read data contained on a stale subdisk, the data is reconstructed from data from all other stripe units in the stripe; this operation is called a reconstruct-read. This is a significantly more expensive operation than simply reading the data, resulting in degraded read performance; thus, when a RAID5 volume has stale subdisks, it is considered to be in *degraded mode*.

To display if a RAID5 volume is in degraded mode, enter:

```
# volprint -ht
```

Output similar to the following is displayed:

```
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       r5vol   RAID5   ENABLED DEGRADED        20480   RAID    -
pl      r5vol-01        r5vol   ENABLED ACTIVE  20480   RAID    3/16    RW
sd      disk00-00       r5vol-01        disk00  0       10240   0/0     dsk4d1
sd      disk01-00       r5vol-01        disk01  0       10240   1/0     dsk2d1  dS
sd      disk02-00       r5vol-01        disk02  0       10240   2/0     dsk3d1  -
pl      r5vol-l1        r5vol   ENABLED LOG     1024    CONCAT  -       RW
sd      disk03-01       r5vol-l1        disk00  10240   1024    0       dsk3d0  -
pl      r5vol-l2        r5vol   ENABLED LOG     1024    CONCAT  -       RW
sd      disk04-01       r5vol-l2        disk02  10240   1024    0       dsk1d1  -
```

The output shows that volume `r5vol` is in degraded mode, as shown by the STATE, which is listed as DEGRADED. The failed subdisk is `disk01-00`, as shown by the flags in the last column. The `d` indicates that the subdisk is detached and the `S` indicates that the subdisk contents are stale.

It is also possible that a disk containing a RAID5 log could experience a failure. This has no direct effect on the operation of the volume; however, the loss of all RAID5 logs on a volume makes the volume vulnerable to a complete failure.

The following `volprint` output shows a failure within a RAID5 log plex as indicated by the plex state being BADLOG, where the RAID5 log plex `r5vol-l1` has failed.

```
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       r5vol   RAID5   ENABLED ACTIVE  20480   RAID    -
pl      r5vol-01        r5vol   ENABLED ACTIVE  20480   RAID    3/16    RW
sd      disk00-00       r5vol-01        disk00  0       10240   0/0     dsk4d1  ENA
sd      disk01-00       r5vol-01        disk01  0       10240   1/0     dsk2d1  dS
sd      disk02-00       r5vol-01        disk02  0       10240   2/0     dsk3d1  ENA
pl      r5vol-l1        r5vol   DISABLED        BADLOG  1024    CONCAT  -       RW
sd      disk03-01       r5vol-l1        disk00  10240   1024    0       dsk3d0  ENA
pl      r5vol-l2        r5vol   ENABLED LOG     1024    CONCAT  -       RW
sd      disk04-01       r5vol-l2        disk02  10240   1024    0       dsk1d1  ENA
```

8.5.3 RAID5 Recovery

The following are the types of recovery typically needed for RAID5 volumes:

- Parity resynchronization
- Stale subdisk recovery
- Log plex recovery

These types of recoveries are discussed in the sections that follow. Parity resynchronization and stale subdisk recovery are typically performed when the RAID5 volume is started, shortly after the system boots, or by calling the `volrecover` command.

If hot-sparing is enabled at the time of a disk failure, system administrator intervention is not required (unless there is no suitable disk space available for relocation). Hot-sparing will be triggered by the failure and the system administrator will be notified of the failure by electronic mail. Hot-sparing will automatically attempt to relocate the subdisks of a failing RAID5 plex. After any relocation takes place, the hot-sparing daemon (`volspared`) will also initiate a parity resynchronization. In the case of a failing RAID5 log plex, relocation will only occur if the log plex is mirrored; `volspared` will then initiate a mirror resynchronization to recreate the RAID5 log plex. If hot-sparing is disabled at the time of a failure, the system administrator may need to initiate a resynchronization or recovery.

8.5.3.1 Parity Resynchronization

In most circumstances, a RAID5 array will not have stale parity. Stale parity should only occur after all RAID5 log plexes for the RAID5 volume have failed, and then only if there is a system failure. Furthermore, even if a RAID5 volume has stale parity, it is usually taken care of as part of the `volume start` process.

However, if a volume without valid RAID5 logs starts and the process is killed before the volume is resynchronized, the result is an active volume with stale parity.

To display volume state, enter:

```
# volprint -ht
```

Output similar to the following is displayed:

```
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       r5vol     RAID5   ENABLED NEEDSYNC        20480   RAID    -
pl      r5vol-01          r5vol   ENABLED ACTIVE  20480   RAID    3/16    RW
sd      disk00-00         r5vol-01        disk00  0       10240   0/0     dsk4d1  ENA
sd      disk01-00         r5vol-01        disk01  0       10240   1/0     dsk2d1  ENA
sd      disk02-00         r5vol-01        disk02  0       10240   2/0     dsk3d1  ENA
```

This output displays the volume state as NEEDSYNC, indicating that the parity needs to be resynchronized. The state could also have been SYNC, indicating that a synchronization was attempted at start time and that a synchronization process should be doing the synchronization. If no such process exists or if the volume is in the NEEDSYNC state, a synchronization can be manually started using the volume resync command.

To resynchronize the a RAID5 volume called `r5vol`, enter:

```
# volume resync r5vol
```

Parity is regenerated by issuing VOL_R5_RESYNC ioctls to the RAID5 volume. The resynchronization process starts at the beginning of the RAID5 volume and resynchronizes a region equal to the number of sectors specified by the `-o iosize` option to the volume command or, if `-o iosize` is not specified, the default maximum I/O size. The resync command then moves onto the next region until the entire length of the RAID5 volume is resynchronized.

For larger volumes, parity regeneration can take a significant amount of time and it is possible that the system can shut down or crash before the operation is completed. Unless the progress of parity regeneration is kept across reboots, the process starts over again.

To avoid this situation, parity regeneration is checkpointed, meaning that the offset up to which the parity is regenerated is saved in the configuration database. The `-o checkpt=size` option to the volume command controls how often the checkpoint is saved; if not specified, it defaults to the default checkpoint size. Because saving the checkpoint offset requires a transaction, making the checkpoint size too small can significantly extend the time required to regenerate parity. After a system reboot, a RAID5 volume that has a checkpoint offset smaller than the volume length will start a parity resynchronization at the checkpoint offset.

8.5.3.2 Stale Subdisk Recovery

Like parity resynchronization, stale subdisk recovery is usually done at volume start time. However, it is possible that the process doing the recovery may get killed, or that the volume was started with an option to prevent subdisk recovery. It's also possible that the disk on which the subdisk resides was replaced without any recovery operations being performed.

To recover a stale subdisk in a RAID5 volume, enter:

```
# volume recover r5vol dsk01-00
```

To recover multiple stale subdisks in a RAID5 volume at once with only the name of the volume, enter:

```
# volume recover r5vol
```

8.5.3.3 Log Plex Recovery

RAID5 log plexes may become detached due to disk failures. To reattach failed RAID5 log plex, enter:

```
# volplex att r5vol r5vol-l1
```

8.6 Startup Problems

The following sections describe LSM command and startup problems and suggests corrective actions.

8.6.1 I/O and System Delays Caused by Disk Failure

When a mirrored LSM disk fails, the system may hang for several minutes before resuming activity.

If you observe long delays in LSM recovery from disk failure, this is usually due to the underlying device driver, not LSM. When an initial I/O operation fails, there may be a delay as the device driver waits or retries the I/O. The length of the delay depends on the particular tolerances for that drive (for example, time for drive spin-up, ECC calculation time, retries and recalibration by the drive, other I/O being handled due to command-tag queuing, bus/device initialization time after reset, and so on).

LSM does not perform additional retries or generate additional delays when an I/O fails on a device. Once the underlying device driver returns an I/O failure error to LSM, LSM processes the error immediately (for example, issues another read to the other plex to recover and mask the error).

To reduce such delays, see the driver documentation for instructions on changing the retry parameter settings.

8.6.2 An LSM Command Fails to Execute

When an LSM command fails to execute, LSM may display the following message:

```
Volume daemon is not accessible
```

This message often means that the volume daemon `vold` is not running.

Verify that the `vold` daemon is enabled by entering the following command:

```
# voldctl mode
```

Output similar to the following is displayed:

```
mode: enabled
```

Verify that the two `voliod` or more daemons are running by entering the following command:

```
# voliod
```

Output similar to the following is displayed:

```
2 volume I/O daemons are running
```

8.6.3 LSM Volume I/O or Mirroring Fails to Complete

Follow these steps if I/O to a LSM volume or mirroring of a LSM volume does not complete:

- Check whether or not the LSM I/O daemon, `voliod`, is running:

  ```
  # voliod
  ```

  Output similar to the following should display:

  ```
  2 volume I/O daemons are running
  ```

- If the volume is in the ENABLED/SYNC state, the mirror resynchronization may have terminated abnormally. Restart the volume synchronization:

  ```
  # volrecover
  ```

  If resynchronization still does not start, make sure the `volume` command is not running in the background and that the volume's `rwback` offset is not progressing by entering the following commands:

  ```
  # ps aux | grep volume
  ```

  Output similar to the following is displayed:

  ```
  root 4322 0.0 0.0 1.62M 160K console S + 19:36:15 0:00.00 grep volume
  ```

  ```
  # volprint -vl my_vol | grep flags
  ```

  Output similar to the following is displayed:

  ```
  flags:    open rwback (offset=121488) writeback
  ```

  ```
  # sleep 120 ; volprint -vl my_vol | grep flags
  ```

  Output similar to the following is displayed:

  ```
  flags:    open rwback (offset=121488) writeback
  ```

  If the `ps` command output shows no `volume` commands running and the volume's `rwback` offset remains the same, use the `volume -o force` command to restart resynchronization by entering the following command:

  ```
  # volume -o force resync volume_name
  ```

8.6.4 Failures While Creating Volumes or Adding Disks

When creating a new volume or adding a disk, the operation may fail with the following message:

```
No more space in disk group configuration
```

This message could mean that you are out of space in the disk group's configuration database. Check to see if any disks were configured with 2 or more configuration databases. If all disks with active configuration databases are configured to use 1 configuration database within their private region, then check if a disk with a smaller private region can be reconfigured to deactive the configuration database within this smaller private region.

If all the disks have `nconfig` set to 1 and the same size private regions, you can reconfigure and/or add disks with larger private regions.

To display if the `rootdg` disk group is using a disk with more than 1 configuration database, enter:

voldg list rootdg

Output similar the following is displayed:

```
Group:     rootdg
dgid:      921610896.1026.rio.dec.com
import-id: 0.1  flags:
copies:    nconfig=default nlog=default
config:    seqno=0.1091 permlen=1496 free=1490 templen=3 loglen=226
config disk dsk7 copy 1 len=1496 state=clean online
config disk dsk7 copy 2 len=1496 disabled
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=226
log disk dsk7 copy 2 len=226 disabled
log disk dsk8 copy 1 len=453
log disk dsk9 copy 1 len=453
log disk dsk10 copy 1 len=453
```

To increase the `rootdg` disk group free space from 1490 to 2987 by changing `dsk7` to have 1 configuration database copy instead of 2 within its private region, enter:

voldisk moddb dsk7 nconfig=1

To display the results, enter:

voldg list rootdg

Output similar the following is displayed:

```
Group:     rootdg
dgid:      921610896.1026.rio.dec.com
import-id: 0.1  flags:
copies:    nconfig=default nlog=default
config:    seqno=0.1091 permlen=2993 free=2987 templen=3 loglen=453
config disk dsk7 copy 1 len=2993 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=453
log disk dsk8 copy 1 len=453
```

```
log disk dsk9 copy 1 len=453
log disk dsk10 copy 1 len=453
```

You can check the active configuration database sizes on each disk within a disk group to see if you can reconfigure a disk with a smaller private region to deactivate the configuration database within its smaller private region.

Follow steps to disable a configuration database:

1. Display the current configuration:

   # **voldg list rootdg**

   Output similar the following is displayed:

   ```
   Group:     rootdg
   dgid:      921610896.1026.rio.dec.com
   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
   ```

2. To disable the configuration databases on `dsk7`, so the `rootdg` configuration database free size will increase from 341 to 2987, enter:

   # **voldisk moddb dsk7 nconfig=0**

3. Display new configuration:

   # **voldg list rootdg**

   Output similar the following is displayed:

   ```
   Group:     rootdg
   dgid:      921610896.1026.rio.dec.com
   import-id: 0.1  flags:
   copies:    nconfig=default nlog=default
   config:    seqno=0.1081 permlen=2993 free=2987 templen=3 loglen=453
   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 dsk8 copy 1 len=453
   log disk dsk9 copy 1 len=453
   log disk dsk10 copy 1 len=453
   ```

If all disks have 1 configuration copy and you cannot disable disks with smaller private regions, then you can add and use disks with a larger private region. Follow these steps to specify a private region larger than the default of 4096:

1. Enter the `voldisksetup` command with the `privlen` option to specify a new private region size.

2. Use the `voldisk moddb` command as described earlier in this section to deactivate the smaller disks.

Note

For a disk group with 4 or more disks, you should enable and configure at least 4 of the disks to be large enough to contain the disk group's configuration database.

Follow these steps to add 4 disks with larger configuration databases and disable the configuration database on the smaller disks, so only the new disks with the larger private region are used:

1. Display the current disk group configuration by entering the following command:

   ```
   # voldg list rootdg
   ```

   Output similar the following is displayed:

   ```
   Group:     rootdg
   dgid:      921610896.1026.rio.dec.com
   import-id: 0.1  flags:
   copies:    nconfig=default nlog=default
   config:    seqno=0.1091 permlen=2993 free=2987 templen=3 loglen=453
   config disk dsk7 copy 1 len=2993 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=453
   log disk dsk8 copy 1 len=453
   log disk dsk9 copy 1 len=453
   log disk dsk10 copy 1 len=453
   ```

2. Increase the private region size by entering the following commands:

   ```
   # voldisksetup -i dsk3 privlen=8192
   # voldisksetup -i dsk4 privlen=8192
   # voldisksetup -i dsk12 privlen=8192
   # voldisksetup -i dsk13 privlen=8192
   ```

3. Add the disks to the disk group by entering the following command:

   ```
   # voldg adddisk dsk3 dsk4 dsk12 dsk13
   ```

4. Deactivate the smaller disks by entering the following commands:

   ```
   # voldisk moddb dsk7 nconfig=0
   # voldisk moddb dsk8 nconfig=0
   # voldisk moddb dsk9 nconfig=0
   # voldisk moddb dsk10 nconfig=0
   ```

5. Display the new configuration by entering the following command:

voldg list rootdg

Output similar to the following is displayed:

```
Group:     rootdg
dgid:      921610896.1026.rio.dec.com
import-id: 0.1  flags:
copies:    nconfig=default nlog=default
config:    seqno=0.1116 permlen=6017 free=6007 templen=3 loglen=911
config disk dsk3 copy 1 len=6017 state=clean online
config disk dsk4 copy 1 len=6017 state=clean online
config disk dsk12 copy 1 len=6017 state=clean online
config disk dsk13 copy 1 len=6017 state=clean online
log disk dsk3 copy 1 len=911
log disk dsk4 copy 1 len=911
log disk dsk12 copy 1 len=911
log disk dsk13 copy 1 len=911
```

8.6.5 Mounting a File System or Opening an LSM Volume Fails

If a file system cannot be mounted or an open function on an LSM volume fails, check if `errno` is set to EBADF. This could mean that the LSM volume is not started.

To determine whether or not the volume is started, enter:

volinfo -g rootdg

Output similar the following is displayed:

```
vol1         fsgen  Startable
vol-dsk3h    fsgen  Started
vol2         fsgen  Started
swapvol1     gen    Started
rootvol      root   Started
swapvol      swap   Started
```

To start volume `vol1`, enter:

volume -g rootdg start vol1

8.7 Restoring an LSM Configuration

You use the `volrestore` command to restore an LSM configuration that you saved when using the `volsave` command. If you enter the `volrestore` command with no options, `volrestore` attempts to restore all disk groups. If you use the `-i` (interactive) option, `volrestore` prompts you before restoring each disk group.

Before the `volrestore` command restores the LSM configuration, it validates the checksum that is part of the description set.

By default, the `volrestore` command restores the whole configuration, using the description set in the directory under `/usr/var/lsm/db` that has the latest timestamp. You can specify options to the command to use a different directory and to restore a specific volume or disk group. For example, this command restores only the volume called myvol01 in the staffdg disk group:

```
# volrestore -g staffdg -v myvol01
```

When you restore a specific disk group, the volrestore command attempts to reimport the disk group based on configuration information on disks that belong to that disk group. If the import fails, volrestore recreates the disk group by reinitializing all disks within that disk group and recreating all volumes, unassociated plexes, and unassociated subdisks, based on information in the volmake description file, allvol.DF

Notes

The volrestore command does not restore volumes associated with the root, /usr, and /var file systems and the primary swap area. These partitions must be reencapsulated to use LSM volumes.

See the Tru64 UNIX Clusters documentation before using volrestore in Tru64 UNIX cluster environment.

When you restore a complete LSM configuration, the volrestore command attempts to reenable the vold based on the configuration databased found on the rootdg disks. If the complete LSM configuration does not need to be restored, you can use the -i (interactive) option with volrestore. The volrestore command prompts you before restoring each file, enabling you to skip specific disk groups.

If vold cannot be enabled, you are given the option of recreating the rootdg disk group and any other disk groups using the other files in the saved LSM description set. The rootdg disk group is recreated first, and vold is put in the enabled mode. Then, the other disk groups are enabled. The disk groups are recreated by first attempting to import them based on available disks in that disk group. If the import fails, the disk group is reinitialized and all volumes in that disk group are also recreated based on the volmake description files.

When volumes are restored using the volmake description file, the plexes are created in the DISABLED EMPTY state. The volrestore command does not attempt to start or enable such volumes. You must use volmend or volume to set the plex states appropriately before starting the volume. The volrestore command warns you to check the state of each disk associated with a volume before using volmend to set plex states; to carefully find out which disks in the LSM configuration could have had failures because saving the LSM configuration; and to use `volmend` to mark plexes on those disks to be STALE. In addition, any plex that was detached or disabled at any point during or after the LSM configuration was saved should be marked "STALE" using `volmend`.

To restore a disk group called `dg1`, enter the following command, and the system will display output similar to this example:

```
# volrestore -g dg1
Using LSM configuration from /usr/var/lsm/db/LSM.19991226203620.skylark
Created at Tue Dec 26 20:36:30 EST 1999 on HOST skylark

Would you like to continue ? [y,n,q,?] (default: n) y
Working .
Restoring dg1
vol1 in diskgroup dg1 already exists. (Skipping ..)
vol2 in diskgroup dg1 already exists. (Skipping ..)
vol3 in diskgroup dg1 already exists. (Skipping ..)
```

8.7.1 Conflicts While Restoring the Configuration

When `volrestore` executes, it can encounter conflicts in the LSM configuration, for example, if another volume uses the same plex name or subdisk name, or the same location on a disk. When `volrestore` finds a conflict, it displays error messages and the configuration of the volume, as found in the saved LSM description set. In addition, it removes all volumes created in that disk group during the restoration. The disk group that had the conflict remains imported, and `volrestore` continues to restore other disk groups.

If `volrestore` fails because of a conflict, you can use the `volrestore -b` option to do the best possible restoration in a disk group. You will then have to resolve the conflicts and restore the volumes in the affected disk group.

8.7.2 Failures in Restoring the Configuration

The restoration of volumes fails if one or more disks associated with the volumes are unavailable, for example due to disk failure. This, in turn, causes the restoration of a disk group to fail. To restore the LSM configuration of a disk group, enter:

```
# volrestore -b -g diskgroup
```

The volumes associated with the failed disks can then be restored by editing the `volmake` description file to remove the plexes that use the failed disks. Note that editing the description file affects the checksum of the files in the backup directory, so you must override the checksum validation by using the `-f` option.

8.8 Reinstalling the Operating System

If you reinstall the operating system, LSM-related information, such as data in the LSM private areas on reinstalled disks (containing the disk identifier and copies of the LSM configuration), is removed, which makes the disk unusable to LSM. The only volumes saved are those that reside on, or have copies on, disks that are not directly involved with reinstallation. Volumes on disks involved with the reinstallation are lost during reinstallation. If backup copies of these volumes are available, you can restore them after reinstallation. The system root disk is always involved in reinstallation.

To reinstall the operating system system and recover the LSM configuration you need to:

1. Prepare the system for the installation. This includes replacing any failed disks or other hardware, and detaching any disks not involved in the reinstallation.
2. Install the operating system.
3. Recover the LSM configuration.
4. Complete the configuration by restoring information in volumes affected by the reinstallation and recreate system volumes (such as `rootvol`, `swapvol`, and `usr`).

8.8.1 Preparing the System for the Operating System Reinstallation

To prevent the loss of data on disks not involved in the reinstallation, you should only involve the root disk in the reinstallation procedure. It is recommended that any other disks (that contain volumes) be disconnected from the system before you start the reinstallation procedure.

Disconnecting the other disks ensures that they are unaffected by the reinstallation. For example, if the operating system was originally installed with a file system on the second drive, the file system may still be recoverable. Removing the second drive ensures that the file system remains intact.

8.8.2 Reinstalling the Operating System

After failed or failing disks are replaced and disks uninvolved with the reinstallation are detached, reinstall the operating system and LSM as described in the *Installation Guide*.

While the operating system installation progresses, make sure no disks other than the root disk are accessed in any way. If anything is written on a disk other than the root disk, the LSM configuration on that disk could be destroyed.

8.8.3 Recovering the LSM Configuration

Use the `volrestore` procedure to recover the LSM configuration information that was previously saved with `volsave`. If the LSM configuration information can not be restored using `volrestore`, use the following procedure to reinitialize LSM.

Warning

Executing the `volsetup` command with the `-o force` option destroys any existing LSM configuration information on a system.

Once the LSM subsets have been loaded, recover the LSM configuration by doing the following:

1. Shut down the system.
2. Physically reattach the disks that were removed from the system.
3. Reboot the system. When the system comes up, make sure that all disks are configured in the kernel and that special device files have been created for the disks.
4. Run the `volsetup` script. This script checks for an existing LSM configuration and starts LSM if one exists. If an existing configuration is found, the script displays the following message:

   ```
   LSM has detected the presence of an existing configuration.
   Check the current configuration and use '-o force' option to
   destroy the existing configuration if necessary.
   ```

5. Recreate the LSM configuration. If the LSM configuration was previously saved using the `volsave` command, use the `volrestore` command. Otherwise, you must recreate the volumes, plexes subdisks, disks, and disk groups using the procedures described in Chapter 5.
6. Restore the volume's data using the appropriate backup and restore command. For example, to restore an AdvFS or UFS file system that was backed up with the `vdump` command, you would use the `vrestore` command.
7. If the `root` file system, swap partition, and/or `usr` file system were previously under LSM control, you can reconfigure the system disk under LSM control and mirror the disk using the procedures described in Chapter 4.

The configuration preserved on the disks not involved with the reinstallation has now been recovered. However, because the root disk has been reinstalled, it appears to LSM as a non-LSM disk. Therefore, the configuration of the preserved disks does not include the root disk as part of the LSM configuration.

Note

If the root disk of your system and any other disk involved in the reinstallation were not under LSM control at the time of failure and reinstallation, then the reconfiguration is complete at this point. If other disks containing volumes or volume plexes are to be replaced, follow the replacement procedures in Section 8.2.2.2 .

8.8.4 Completing the Configuration

If the boot disk (or another disk) was involved with the reinstallation, any volume or volume plexes on that disk (or other disks no longer attached to the system) are now inaccessible. If a volume had only one plex (contained on a disk that was reinstalled, removed, or replaced), then the data on that the volume is lost and must be restored from backup. In addition, the system's root file system and swap area are not located on volumes any longer.

8.8.4.1 Removing the Root and Swap Volumes

Remove volumes associated with root and swap areas, and their associated disks. This must be done if the root disk was under LSM control prior to reinstallation. The volumes to remove are:

- `rootvol`, which contains the root file system
- `swapvol`, which contains the swap area

Follow these steps to remove the `rootvol` and `swapvol` volumes:

1. Stop the root and swap volumes and remove them by entering the following commands:

   ```
   # volume stop rootvol
   # voledit -r rm rootvol
   # volume stop swapvol
   # voledit -r rm swapvol
   ```

2. Remove the LSM disks used by `rootvol` and `swapvol`. For example, if disk `dsk3` was associated with `rootvol` and disk `dsk3b` was associated with `swapvol`:

```
# voldg rmdisk dsk3 dsk3b
# voldisk rm dsk3 dsk3b
```

8.8.4.2 LSM Volumes for /usr and /var Partitions

If /usr and /var partitions were on LSM volumes prior to the reinstallation, then remove the LSM disks associated with them using the voledit command in the previous example shown for rootvol and swapvol.

8.8.4.3 Restoring Volumes from Backup

After configuring the volumes, you must determine which volumes need to be restored from backup. The volumes to be restored include any volumes that had all plexes residing on disks that were removed or reinstalled. These volumes are invalid and must be removed, recreated, and restored from backup. If only some plexes or a volume exist on reinitialized or removed disks, these plexes must be removed. The plexes can be readded later.

Follow these steps to restore the volumes:

1. Establish which LSM disks have been removed or reinstalled:

   ```
   # voldisk list
   ```

   Output similar to the following is displayed:

   ```
   DEVICE   TYPE       DISK       GROUP      STATUS
   dsk0      sliced     -          -          error
   dsk1      sliced     disk02     rootdg     online
   dsk2      sliced     disk03     rootdg     online
   -        -          disk01     rootdg     failed was:  dsk0
   ```

   This output shows that the reinstalled root device, dsk0 is not recognized as an LSM disk and is marked with a status of error. disk02 and disk03 were not involved in the reinstallation and are recognized by LSM and associated with their devices (dsk1 and dsk2). The former disk01, the LSM disk that had been associated with the replaced disk device, is no longer associated with the device (dsk0). If there had been other disks (with volumes or volume plexes on them) removed or replaced during reinstallation, these disks would also have a disk device in error state and an LSM disk listed as not associated with a device.

2. Once you know which disks are removed or replaced, display the plexes on disks with a status of failed:

   ```
   # volprint -sF "%vname" -e 'sd_disk = "<disk>"'
   ```

In this command, the variable is the name of a disk with a `failed` status.

Note

Be sure to enclose the disk name in quotes in the command. Otherwise, the command displays an error message.

The `volprint` command displays a list of volumes that have plexes on the failed disk. Repeat this command for each disk with a `failed` status.

3. Check the status of each volume by entering the following command:

   `volprint -th` *volume_name*

   For example, to infomation about a volume called `fnah`, enter:

   `#` **`volprint -th fnah`**

   Output similar to the following is displayed:

```
V  NAME      USETYPE  KSTATE   STATE    LENGTH READPOL  PREFPLEX
PL NAME      VOLUME   KSTATE   STATE    LENGTH LAYOUT ST-WIDTH MODE
SD NAME      PLEX     PLOFFS   DISKOFFS LENGTH DISK-MEDIA   ACCESS

v  fnah       fsgen    DISABLED ACTIVE   24000  SELECT   -
pl fnah-01   fnah     DISABLED NODEVICE 24000  CONCAT   -
sd disk01-06 fnah-01  0        519940   24000  disk01   -
```

4. In this output, the only plex of the volume is shown in the line beginning with `pl`. The `STATE` field for the plex called `fnah-01` is `NODEVICE`. The plex has space on a disk that was replaced, removed, or reinstalled. Therefore, the plex is no longer valid and you must remove it.

   Because the `fnah-01` plex was the only plex of the volume, the volume contents are irrecoverable except by restoring the volume from a backup. You must also remove the volume. If a backup copy of the volume exists, you can restore the volume later. Keep a record of the volume name and its length, you will need it for the backup procedure.

   Remove the volume by entering the following command:

   `voledit -r rm` *volume_name*

   For example, to remove a volume called `fnah`, enter:

   `#` **`voledit -r rm fnah`**

   It is possible that only part of a plex is located on the failed disk. If the volume has a striped plex associated with it, the volume is divided between several disks. For example, the volume called `vol01` has one striped plex, striped across three disks, one of which is the reinstalled disk `disk01`. The output of the `volprint -th` command for `vol01` displays output similar to the following:

```
V  NAME       USETYPE  KSTATE   STATE    LENGTH  READPOL  PREFPLEX
PL NAME       VOLUME   KSTATE   STATE    LENGTH  LAYOUT   ST-WIDTH MODE
SD NAME       PLEX     PLOFFS   DISKOFFS LENGTH  DISK-MEDIA        ACCESS

v  vol01      fsgen    DISABLED ACTIVE   4224    SELECT   -
pl vol01-01   vol01    DISABLED NODEVICE 4224    STRIPE   128    RW
sd disk02-02  vol01-01 0        14336    1408    disk02          dsk1
sd disk01-05  vol01-01 1408     517632   1408    disk01   -
sd disk03-01  vol01-01 2816     14336    1408    disk03          dsk2
```

This output shows three disks, across which the plex `vol01-01` is striped (the lines starting with `sd` represent the stripes). The second stripe area is located on the LSM disk called `disk01`. This disk is no longer valid, so the plex called `vol01-01` has a state of `NODEVICE`. Because this is the only plex of the volume, the volume is invalid and must be removed. If a copy of `vol01` exists on the backup media, it can be restored later.

Note

Keep a record of the volume name and length of any volumes you intend to restore from backup.

Use the `voledit` command to remove the volume, as described earlier.

A volume that has one plex on a failed disk may also have other plexes on disks that are still valid. In this case, the volume does not need to be restored from backup, because the data is still valid on the valid disks. The output of the `volprint -th` command for a volume with one plex on a failed disk (`disk01`) and another plex on a valid disk (`disk02`) displays output similar to the following:

```
V  NAME   USETYPE  KSTATE   STATE    LENGTH  READPOL  PREFPLEX
PL NAME   VOLUME   KSTATE   STATE    LENGTH  LAYOUT   ST-WIDTH  MODE
SD NAME   PLEX     PLOFFS   DISKOFFS LENGTH  DISK-MEDIA         ACCESS

v  foo       fsgen    DISABLED ACTIVE   10240   SELECT   -
pl foo-01    foo      DISABLED ACTIVE   10240   CONCAT   -      RW
sd disk02-01 foo-01   0        0        10240   disk02   dsk1
pl foo-02    foo      DISABLED NODEVICE 10240   CONCAT          RW
sd disk01-04 foo-02   0        507394   10240   disk01   -
```

This volume has two plexes, `foo-01` and `foo-02`. The first plex, `foo-01`, does not use any space on the invalid disk, so it can still be used. The second plex, `foo-02`, uses space on the invalid disk, `disk01`, and has a state of `NODEVICE`. Mirror `foo-02` must be removed. However, the volume still has one valid plex containing valid data. If the volume needs to be mirrored, another plex can be added later. Note the name of the volume if you want to create another plex later.

To remove an invalid plex, you must dissociated the plex from the volume and remove the plex. To remove the plex called `foo-02`, enter:

```
# volplex -o rm dis foo-02
```

5. Once all the volumes are cleaned up, you must clean up the disk configuration as described in the following section.

8.8.4.4 Disk Cleanup

Once all invalid volumes and volume plexes are removed, the disk configuration can be cleaned up. Each disk that was removed, reinstalled, or replaced (as determined from the output of the `voldisk list` command) must be removed from the configuration.

To remove the disk, use the `voldg` command. To remove the failed `disk01`, enter:

```
# voldg rmdisk disk01
```

If the `voldg` command returns an error message, some invalid volume plexes exist. Repeat the processes described in "Volume Cleanup" until all invalid volumes and volume plexes are removed.

8.8.4.5 Reconfiguring the root Volume

Once all the invalid disks are removed, you can replace or reinstall disks to add them to LSM control. If the root disk was originally under LSM control (the root file system and the swap area were on volumes), or you now want to put the root disk under LSM control, add this disk first, for example:

```
# /usr/sbin/volencap devname
```

See Chapter 4 for more information.

When the encapsulation is complete, reboot the system to multiuser mode.

8.8.4.6 Reconfiguring Volumes

After the boot disk is encapsulated, you can replace other disks. If the disks were reinstalled during the operating system reinstallation, they should be encapsulated; otherwise, add them.

Once the disks are added to the system, you can recreate the volumes that were removed and restore their contents from backup.

To recreate the volumes `fnah` and `vol01`, enter:

```
# volassist make fnah 24000
# volassist make vol01 4224 layout=stripe nstripe=3
```

To replace the plex removed from the volume `foo` using `volassist`, enter:

```
# volassist mirror foo
```

Once you restore the volumes and plexes, the recovery is complete and your system should be configured as it was prior to reinstalling the Tru64 UNIX operating system.

8.9 Deconfiguring Additional Swap

Follow these steps to deconfigure and remove additional swap volumes that were previously configured for use with the LSM software:

1. Deconfigure the swap space to no longer use the LSM volumes. This can be done by updating the `vm:swapdevice` entry in the `sysconfigtab` file to not reference the LSM volumes. If the swap space was configured using the `/etc/fstab` file, update this file accordingly.

   See the *System Administration* and `swapon`(8) reference pages for more information.

2. Reboot the system to affect the change.

3. Stop and remove the volumes. For example, to stop and remove a volume called `swapvol1`, enter:

   ```
   # volume stop swapvol1
   # voledit -rf rm swapvol1
   ```

8.10 Removing the LSM Software

Follow these steps to deconfigure and remove LSM from a system.

Warning

Deconfiguring LSM causes any data currently under LSM to be lost and no longer accessible. You should unencapsulate and/or backup any needed data before proceeding.

1. Reconfigure any system file systems and/or swap space to no longer be on a LSM volume. If root and swap are configured under LSM, enter the `volunroot` command and reboot the system. Also, unencapsulate the /usr and /var if these are configured under LSM. See Chapter 4 if `/usr` and `/var` are encapsulated under LSM with the root and swap. If additional swap space was configured using LSM volumes, deconfigure them as described in Section 8.9.

2. Unmount any other filesystems that were using LSM volumes so all LSM volumes can be closed. Update the /etc/fstab file if necessary to no longer mount any file systems on an LSM volume. Stop applications that are using raw LSM volumes and reconfigure them to no longer use LSM volumes.

3. Note of which disks are currently configured under LSM by entering the following command:

   ```
   # voldisk list
   ```

4. Once all the LSM volumes are no longer in-use, restart LSM in disabled mode by entering the following command:

   ```
   # vold -k -r reset -d
   ```

   This command fails if any volumes are open.

5. Stop LSM's volume and I/O daemons by entering the following command:

   ```
   # voliod -f set 0
   # voldctl stop
   ```

6. Update the disk labels using the list of disks under LSM from step 3 above. For each disk that was previously configured under LSM as sliced (for example, entire disk was under LSM), repartition and update the disk labels using the -rw option by entering the following commands:

   ```
   # disklabel -rw dsk4
   # disklabel -rw dsk5
   ```

   For each disk partition that was configured under LSM as a simple disk, update the partition's fstype to unused using the -s option with the disklabel command. For example:

   ```
   # disklabel -s dsk6c unused
   ```

   Also, update the disk partition fstype field for any nopriv disks that were previously under LSM to either unused or the appropriate value depending on whether the partition still contains valid data. For example, if dsk2g was an an LSM nopriv disk that still contains a valid UNIX file system and dsk2h was a LSM nopriv disk that no longer contains valid data, enter:

   ```
   # disklabel -s dsk2g 4.2BSD
   # disklabel -s dsk2h unused
   ```

7. Remove the LSM directories by entering the following command:

   ```
   # rm -r /etc/vol /dev/vol /dev/rvol /etc/vol/volboot
   ```

8. Delete the following LSM entries in the /etc/inittab file:

```
lsmr:s:sysinit:/sbin/lsmbstartup -b </dev/console >/dev/console 2>&1 ##LSM
lsm:23:wait:/sbin/lsmbstartup </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 by entering the following command:

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

   Output displays the show the installed LSM subsets.

10. Delete the installed LSM subsets by entering the following command:

    ```
    # setld -d OSFLSMBASE500 OSFLSMBIN500
    OSFLSMCLSMTOOLS500
    ```

11. Deconfigure LSM from the kernel. For example, for system named `rio`, replace the `pseudo-device lsm 1` entry in the `/sys/conf/RIO` file to `pseudo-device lsm 0`

    You can make this change either prior to running the `doconfig` command or while running `doconfig` command. For example:

    ```
    # doconfig -c RIO
    ```

12. Copy the new kernel to root and reboot the system by entering the following command:

    ```
    # cp /sys/RIO/vmunix /
    # shutdown now
    ```

9 Using the Storage Administrator

This chapter describes how to manage LSM objects, including disks, disk groups, volumes, plexes, and subdisks using the Storage Administrator GUI. The tasks described in this chapter can also be accomplished by using:

- The LSM commands. See Chapter 5 for more information on the LSM commands.
- The `voldiskadm` menu interface. See Appendix D for more information on the `voldiskadm` menu interface.
- The Visual Administrator. See Appendix B for more information on the Visual Administrator

See Appendix A for more information on how to track Storage Administrator activities, how to use the Storage Administrator, and how to customize the Storage Administrator GUI.

9.1 Overview

The Storage Administrator is the Java-based graphical user interface (GUI) for LSM. The Storage Administrator displays a hierarchical view of LSM objects and their relationships. You use the Storage Administrator to view and manage LSM objects on a local or remote (client) system.

The Storage Administrator provides dialog boxes in which you enter information to create or manage LSM objects. Completing a dialog box may be the equivalent of entering several commands.

Storage Administrator consists of a server (daemon) and a client. The Storage Administrator server runs on a system on which the LSM software is initialized and running. The Storage Administrator client runs on any machine that supports the Java run-time environment.

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 unless the disk to be mirrored is a clusterwide root, an individual member boot partition, a quorum disk, or swap disk.

- Software-based RAID5 technology is not supported in clusters. If you want RAID5 functionality, you must use hardware RAID devices.
- AdvFS file systems are supported in all modes.
- UFS file systems are supported in read-only mode.

9.1.1 Installing and Starting the Storage Administrator

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. See Chapter 2 for information on initializing the LSM software.

To use the Storage Administrator, you must either log in as `root` or have your user name is in the `/etc/group` file for the system in the group defined for the Storage Administrator administration (`lsmsa_admin` by default).

You can start the Storage Administrator in:

- Full operational mode (the default)

  Full operational mode, allows you to view and perform all LSM administrative tasks. To start Storage Administrator in full operational mode, enter:

  ```
  # /usr/bin/lsmsa
  ```

  The Session Initiation dialog box is displayed is which you enter the following information, then click on the Ok button:

  – The name of the system to be administered in the Host Server field

  – The password associated with the account

  – The `root` or your user name

- Read-only mode

  Read-only mode allows you to view LSM objects but not perform administrative tasks. The system allows you to select objects and display dialog boxes, but will not allow you to complete tasks.

  To start the Storage Administrator server in read-only mode, edit the following line in the `/usr/lib/java/application/lsmsa/properties` file to:

  ```
  vrts.server.readonly=true
  ```

  To restore the Storage Administrator to full operational mode, edit the line to:

  ```
  vrts.server.readonly=false
  ```

You must stop and restart the LSM software to affect the change. Use `/sbin/init.d/lsmsa stop` command to stop LSM and the `/sbin/init.d/lsmsa start` command to restart LSM.

If the Storage Administrator does not start during the boot process, enter:

```
/sbin/init.d/lsmsa stop
```

Wait a few minutes and restart the Storage Administrator by entering the following command:

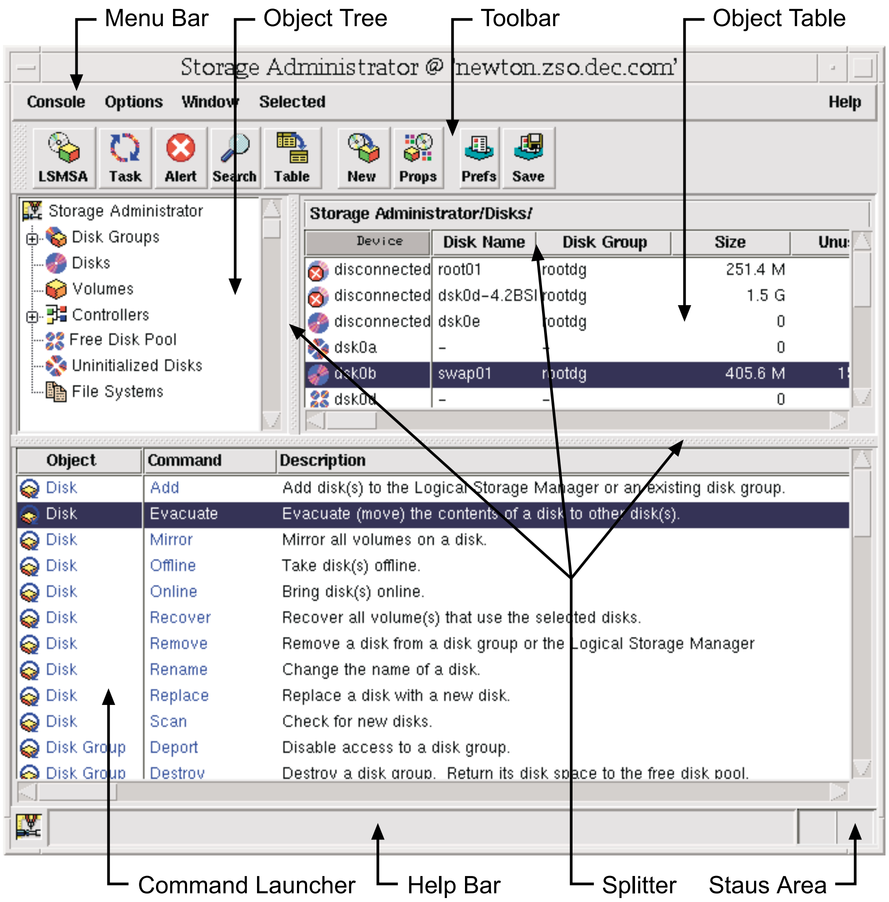
```
/sbin/init.d/lsmsa start
```

You can also access Storage Administrator from the SysMan Menu and from SysMan Station.

9.1.2 Main Window

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 selected 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 9–1 shows the components of the main window.

Figure 9–1: The Storage Administrator Main Window



The Object Tree displays the hierarchical relationship between LSM objects. 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:

- The Console menu that 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.
- The Options menu that 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.
- The Window menu that 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.
- The Selected menu that is dynamic and changes its options based on the type of object that you select. By default, the Selected menu is grayed out.
- The Help menu that provides access to online Storage Administrator help.

The toolbar consists of the following buttons that provide access to the following windows:

- LSMSA button that Launches an additional Storage Administrator Main window.
- Task button that launches the Task Request Monitor window.
- Alerter button that launches the Alert Monitor window.
- Search button that launches the Object Search window.
- Table button that launches a window that contains a copy of the main Object Table.
- New button that launches the New Volume dialog box that is used to create a volume.
- Props button that launches the Object Properties dialog box for a selected object.
- Prefs button that launches the Preferences dialog box.
- Save button that saves the current preference settings for use in future the Storage Administrator sessions.

9.2 Running Tasks

You perform most tasks by selecting objects or tasks, then provide information in resulting dialog boxes. You run tasks using:

- The menu bar

- A pop-up menu
- The Command Launcher

9.2.1 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.

9.2.2 Using the Menu Bar

You can launch tasks from the Console and Selected menus in the menu bar. Choose New from the Console menu to create new LSM volumes, disk groups, and file systems. The context-sensitive Selected menu launches tasks on a selected object.

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 Selected menu, choose Rename, enter information in the Rename Volume dialog box, and click on the OK button.

9.2.3 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 the OK button.

9.2.4 Using the Command Launcher

The Command Launcher window contains a list of objects and associated tasks. To display the Command Launcher, choose Command Launcher from the Window menu. To hide the Command Launcher, 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 the OK button.

Caution

The Command Launcher does not restrict context. If you choose inappropriate commands and ignore the warning dialogs, you can perform operations that may result in permanent loss of data.

9.2.5 Exiting the Storage Administrator

To exit the Storage Administrator client, from the Console menu, choose Exit. If you choose Close and there is no other Storage Administrator Main window open, the system exits.

9.3 Disk Management

The following sections describe disk management tasks that you can complete using the Storage Administrator.

9.3.1 Adding a 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 with the LSM software, 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 disk group.

Follow these steps to add a disk for use with the LSM software:

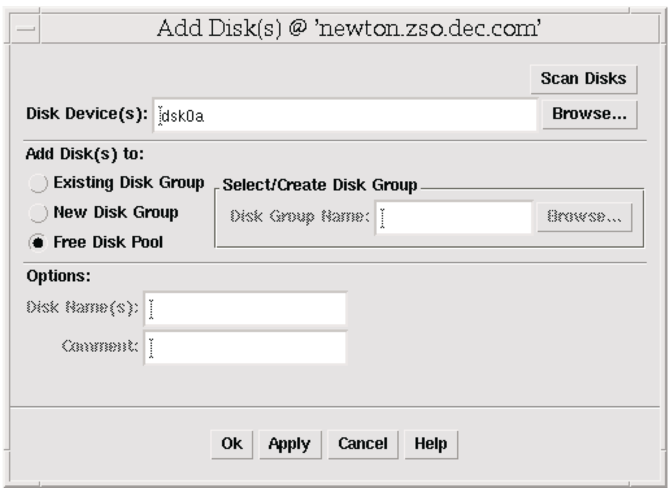
1. In the Object Tree, select Disk and in the Object Table, choose a disk to be placed under LSM control.

   If the disk is new and does not show up in the Object Table when you highlight Disks, click on the Storage Administrator (at the top of the Object Tree) and choose Scan Disks from the Selected menu.

   In the Scan Disks dialog box, click on the Ok button to begin the search. From the Object Tree displayed, select a disk to add.

2. From the Selected menu, choose Add.

The Add Disk dialog box is displayed:



3. In the Add Disk(s) dialog box:

   a. If the correct disk device name is not displayed, type the disk device name or click on Browse to select the disk. You can type 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. Type 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. Type the name of the new disk group in the Disk Group Name field. The new disk group is created.

   c. To add disks to an existing disk group (that is, one that is not deported):

      Specify the LSM disk names for the disks, type 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.

      Enter a comment if desired.

   d. 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.

   e. Click on the Ok button.

You can add one or more unused disks to a disk group and designate them as hot-spares. If an I/O failure occurs, hot-relocation automatically relocates any redundant (mirrored or RAID5) subdisks to the spare disk(s) and restores the affected LSM objects and data. You are notified of the failure and relocation details by electronic mail. See Chapter 7 for more information on the hot-spare feature.

If you designate a hot-relocation spare, provide at least one per disk group. In the event of disk failure, any disk in the disk group can write to it. This will not work across disk groups.

To add a disk as a hot-relocation spare:

1. In the Object Tree, select Disk Group and in the Object Table, select the LSM disk to be designated as a hot-relocation spare.
2. Choose Properties from the Selected menu.
3. In the Disk Properties window:
   a. Select the General tab.
   b. Select Spare.
   c. Click on the Ok button.

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.

9.3.2 Evacuating a Disk

You can evacuate (or move) the contents of the volumes 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 RAID5 volume, do not move the contents of the disk to another disk containing a copy of the mirrored volume or part of the striped/RAID5 volume.

Follow these steps to evacuate a disk from LSM control:

1. In the Object Tree, select Disk and in the Object Table, select the disk that contains the objects and data to be moved to another disk.

2. From the Selected menu, choose Evacuate.

   The Evacuate dialog box is displayed:

3. In the Evacuate Disk dialog box:

   a. If the correct disk name is not displayed in the Disk Name field, type the disk name or click on Browse then click on the Object Tree to select the disk.

   b. Type 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 the Ok button.

9.3.3 Mirroring a Disk

Using LSM mirrored volumes (RAID1) is a common and effective way to improve data availability. If one disk fails on a mirrored volume, the data can still be accessed from the other copy, or plex. By mirroring data using disks connected to different controllers or buses, you can improve data availability even further because the data will still be accessible if a controller, cable, or storage cabinet fails. Therefore, it is helpful to understand a system's I/O hardware topology; that is, knowing which disk reside on which I/O bus.

Besides improving data availability, you can also use mirroring to significantly improve read performance because multiple reads to the same volume can be done simultaneously using the multiple copies of data. For example, read performance can potentially improve by a factor of two on a mirrored volume with two plexes because twice as many reads can be performed done at the same time.

Writes to the volume result in multiple, simultaneous write requests to each plex, so the time it takes to write to a volume may be slightly longer because of slight performance deviations between individual disks. For example, an individual write might take an additional 5 percent on average to complete because the volume write must wait for both writes to be completed on both plexes (disks).

You can improve overall I/O performance with mirroring because the larger performance gains for read often more than offset the slight degradation for writes. Comparing the number of read operations to the number of write operations on a volume using the `volstat` command can help give you better insight into whether mirroring can also help improve overall performance as well as provide higher data availability.

Because a volume can be changed to either add or remove a mirror on line with the LSM software, the overall performance implications can be measured on the actual I/O workload without stopping or disrupting service to a volume.

Mirrored volumes created with the `volassist` command will have dirty region logging (DRL) enabled by default. DRL is used with the LSM software mirrored volumes to keep track of which regions within the volume are dirty. While DRL may add a modest overhead to writes to the mirrored volume, DRL significantly reduces the time it takes to resynchronize a mirrored volume when rebooting a system after a failure because only the dirty regions within the volume need to be resynchronized instead of the entire volume.

While using a DRL with a mirrored volume is not required and has no affect on data integrity, DRL's dramatic reduction in resynchronization time often more than offset its overhead, so it is usually preferable to configure mirrored volumes with DRL enabled.

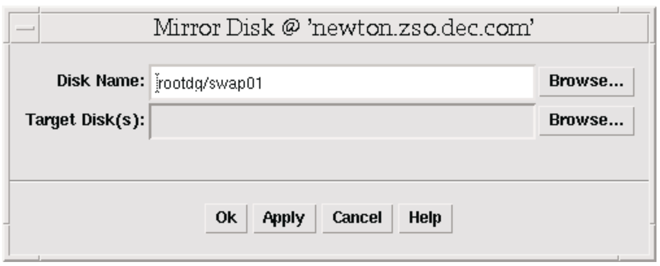
Note

In a TruCluster environment, the resynchronization overhead and time is significantly higher. Therefore, mirrored volumes should always be configured with DRL enabled when in a TruCluster environment.

Follow these steps to mirror all concatenated volumes on a disk:

1. In the Object Tree, select Disk and in the Object Table, select the disk that contains the volumes to be mirrored onto another disk.
2. From the Selected menu, choose Mirror.

   The Mirror Disk dialog box is displayed:



3. In the Mirror Disk dialog box:

   a. If the correct disk name is not displayed in the disk name field, type the disk name or click on Browse to select the disk.

      If you choose Browse, total space being mirrored is displayed so you can choose your target disk accordingly.

   b. To specify the disks to contain the new mirrors, type the target disk name or click on Browse and complete the Target Disk dialog box.

   c. Click on Ok.

9.3.4 Placing a Disk On line

Placing a disk in an on line state restores access to a disk that is in an off line state. The disk is placed in the free disk pool and is accessible to LSM again. After bringing a disk back online, the disk must be added to a disk group before it can be used for volumes.

Only disks that are in an off line state can be placed in an on line state.

Follow these steps to place a disk on line:

1. In the Object Tree, select Disk and in the Object Table and select the disk to be brought on line.
2. From the Selected 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, type the disk name or click on Browse to select the disk.

   b. Click on the Ok button.

9.3.5 Recovering Volumes on a 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 RAID5 volumes. After successful recovery, the volumes should be available for use.

Alert icons and the Alert Monitor window may provide information when a volume recovery is needed.

If recovery of a volume is not possible, restore the volume from backup.

Follow these steps to recover all volumes on a disk:

1. In the Object Tree, select Disk and in the Object Table and select the disk that contains the volumes to be recovered.
2. From the Selected 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, type the disk name or click on Browse to select the disk.

   b. Click on the Ok button.

9.3.6 Removing a Disk from an LSM Disk Group

An LSM disk no longer in use can be removed from an disk group. Do not remove LSM disks that are currently in use (for example, contains subdisks for a volume), doing so can result in loss of data or of data redundancy.

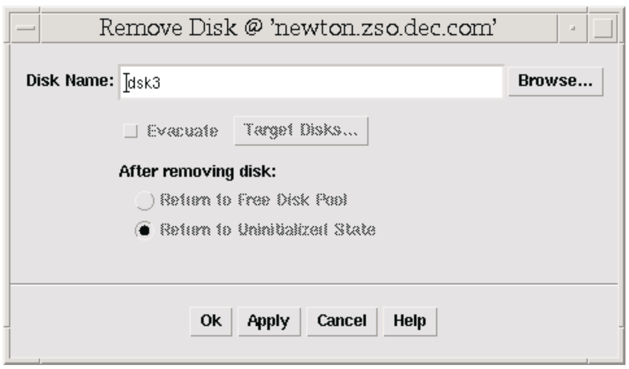
After an LSM disk is removed from a disk group, it is still initialized for use with the LSM software. Therefore, after removing the disk from a disk group, it can be either immediately added to another disk group, removed from LSM, or left initialized for later use.

Follow these steps to remove a disk from an LSM disk group:

Follow these steps to remove a disk from an LSM disk group:

1. In the Object Tree, select Disk and in the Object Table and select the disk to be removed.
2. From the Selected menu, choose Remove.

   The Remove Disk dialog box is displayed:



3. In the Remove Disk dialog box:

   a. If the correct disk name is not displayed, enter the disk name or click on Browse to select the disk.

   b. To move the contents of the disk to another disk(s) before the disk is removed, select Evacuate. Click Target Disks to specify the disk(s) to which you want the contents of the disk 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 the Ok button.

9.3.7 Renaming a Disk

Because disk access names are defined by the operating system and media names are defined by you, you can only rename disk media names for disks in a disk group.

Follow these steps to rename the disk media name for an LSM disk:

1. In the Object Tree, select Disk and in the Object Table, select the disk to be renamed.

2. From the Selected 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, enter the disk name or click on Browse to select the disk.

   b. Enter the new LSM disk name.

   c. Click on the Ok button.

9.3.8 Replacing a Disk

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 RAID5) volumes on the disk. You cannot recover nonredundant volumes. You should 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. You may need to replace a disk if the disk fails and needs to be removed and repaired.

If you replace a good disk, you need to remove the disk from its disk group and place it in the free disk pool *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.

Follow these steps to replace a disk:

1. In the Object Tree, select Disk and in the Object Table, select the disk to be replaced.

2. From the Selected menu, choose Replace.

   The Replace Disk dialog box is displayed:

3. In the Replace Disk dialog box:

   a. If the correct disk name is not displayed, 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 the Ok button.

9.3.9 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.

Follow these steps to scan for a new disk:

1. Click on the scan button if it appears in the dialog box or click on Disk Scan in the Command Launcher.

   The Scan Disk dialog box is displayed.

2. In the Scan Disks dialog box, click on the Ok button.
3. To view disks that are found, click on the Free Disk Pool in the Object Tree.

9.3.10 Taking a Disk Off Line

You can place a disk in an off line state to prevent LSM from accessing it. You must remove a disk from its disk group before you take it off line. An off line disk remains unavailable until you restore access to the disk by placing it on line.

You place a disk in an off line state to protect it from unintentional use, for example, if attempts to access it may have a negative effect on the system. You cannot take a disk that is in use off line.

Follow these steps to take a disk offline:

1. In the Object Tree, select Disk and in the Object Table, select the disk to be taken offline.
2. From the Selected 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, enter the disk name or click on the Browse button to select the disk.

   b. Click on the Ok button.

9.4 Disk Group Management

The following sections describe disk group management tasks that you can complete using the Storage Administrator.

9.4.1 Adding a Disk to a Disk Group

To add a disk to a disk group, follow the instructions for adding a disk and in the Add Disk dialog box and specify an existing disk group as described in Section 9.3.1.

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 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 they can be added to a disk group or the free disk pool. Disks cannot be added 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.

9.4.2 Creating a Disk Group

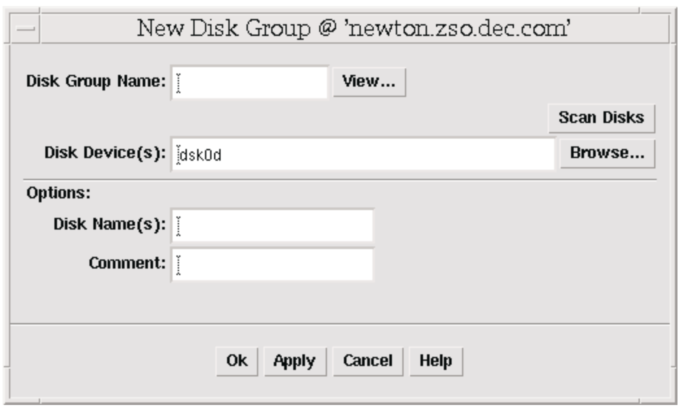
You must place disks in to a disk group before LSM can use them. The default disk group (`rootdg`) is usually created during LSM installation and always exists on a system running LSM. You can create additional disk groups to organize your disks into logical sets.

Each new disk group must contain at least one disk and its name must be unique. You can only use disks that are online and do not belong to a disk group to create a disk group.

Follow these steps to create a disk group:

1. From the Console menu, choose New then Disk Group.

   The New Disk Group dialog box is displayed:



2. In the New Disk Group dialog box:

   a. Enter the name of the disk group to be created. Click 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. Select 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, Storage Administrator assigns a default name to the disk.

      Enter a comment if desired.

   e. Click on the Ok button.

9.4.3 Deporting a Disk Group

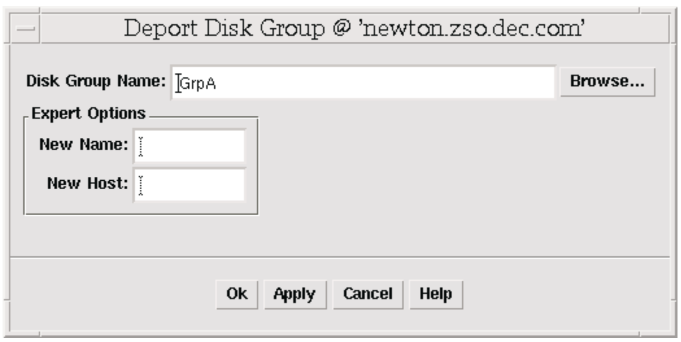
After a disk group is created, the LSM software automatically imports it for use whenever the system is booted.

To disable access to a disk group, you deport the disk group. All the volumes within the disk group should be stopped before deporting the disk group.

Follow these steps to deport a disk group:

1. Select the Storage Administrator (at the top of the Object Tree).
2. From the Selected menu, choose Deport Disk Group.

   The Deport Disk Group dialog box is displayed:



3. In the Deport Disk Group 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. 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 reboot, enter the host ID in the New Host field

   c. Click on the Ok button.

9.4.4 Importing a 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 and 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.

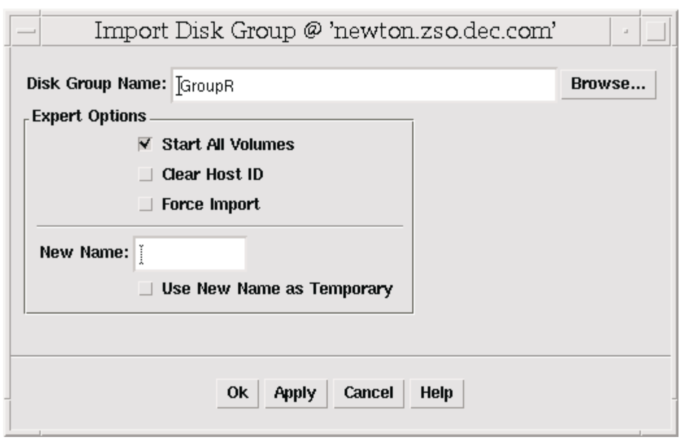
Import may fail for a number of reasons. It may fail if the host cannot find one or more disks in the disk group. If the import fails because a disk has failed, you can import the disk group by selecting the Force Import expert option. If the import fails for another reason, a forced import can cause 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.

Follow these steps to import a disk group:

1. Select the Storage Administrator (at the top of the Object Tree).
2. From the Selected menu, choose Import Disk Group.

   The Import Disk Group dialog box is displayed:



3. In the Import Disk Group 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. 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(s) in the disk group.

      To force the disk group import when the host cannot access all disks in the disk group, select Force Import. Use this option with caution.

   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 returned when the system is rebooted.

e. Click on the Ok button.

9.4.5 Destroying a Disk Group

You can destroy a disk group *permanently* to remove the group from LSM control. It reinitializes all of 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.

Follow these steps to destroy a disk group:

1. Select the Storage Administrator (at the top of the Object Tree).
2. From the Selected 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 or click on Browse to select the disk group.

   b. Click on the Ok button.

9.4.6 Moving a Disk Group

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.

Follow these steps to move a disk group from one system to another:

1. Unmount and stop all volumes in the disk group to be moved.
2. Follow the instructions in Section 9.4.3 to deport the disk group to be moved to the other system.
3. Attach all of the physical disks in the disk group to the new system.
4. On the new system, follow the instructions in Section 9.4.4 to import the disk group.

5. Select the Storage Administrator (at the top of the Object Tree) and from the Selected menu, choose Scan Disks to set up the newly attached disks on the system. This runs the disk setup commands appropriate for the operating system.
6. Follow the instructions in Section 9.4.7 to restart and recover all volumes in the disk group on the new system.

9.4.7 Recovering Volumes in a Disk Group

You can recover volumes in a given 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 RAID5 volumes. After successful recovery, the volumes are available for use.

Alert icons and the Alert Monitor window may provide you information to know when volume recovery is needed.

In some cases, recovery may not be possible. If the volume recovery fails, you can attempt to restore the volume from backup.

Follow these steps to recover all volumes in a disk group:

1. In the Object Tree, select Disk Group and in the Object Table, select the disk group containing the volumes to be recovered.
2. From the Selected menu, choose Recover

   The Recover Disk Groups dialog box is displayed.
3. In the Recover Disk Groups dialog box:

   a. Enter the name of the disk group to be recovered or click on Browse to select the disk group.
   b. Click on the Ok button.

9.4.8 Renaming a Disk Group

You can rename a disk group. If volumes in the disk group are in use (mounted), the disk group is not renamed. Renaming a disk group updates the `/etc/fstab` file.

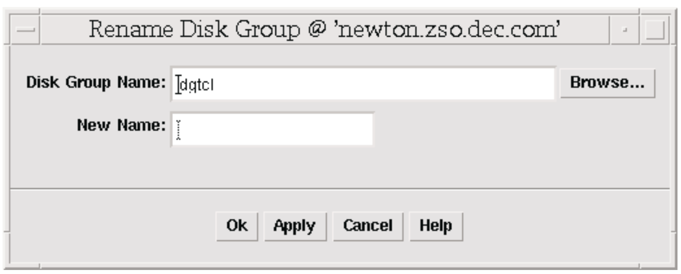
The new disk group name must be unique.

Follow these steps to rename a disk group:

1. In the Object Tree, select Disk Group and in the Object Table, select the disk group to be renamed.

2. From the Selected menu, choose Rename.

   The Rename Disk Group dialog box is displayed:



3. In the Rename Disk Group 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. Enter the new name for the disk group

   c. Click on the Ok button.

9.5 Subdisk Management

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 by using the Storage Administrator:

9.5.1 Joining Subdisks

You can join two or more subdisks together to form a single, larger subdisk. Subdisks can only be joined together 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 striped volume, the subdisks must be in the same column.

Follow these steps to join subdisks:

1. In the Object Tree, select Volume and in the Object Table, select the volume with the subdisks to be joined.
2. From the Selected 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.

   a. From the Selected menu, choose Join.

      The Join Subdisk dialog box is displayed:

   b. In the Join Subdisks dialog box:

      Enter the name of the disk group that contains the subdisks to be joined.

      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.

      Enter the name of the new, combined subdisk.

      Click on the Ok button.

4. Close the Volume Layout Details window. The new volume layout can be viewed by reopening the window.

9.5.2 Moving a Subdisk

You can move portions of a volume 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 RAID5 volume to a disk that already contains a copy or part of that volume.

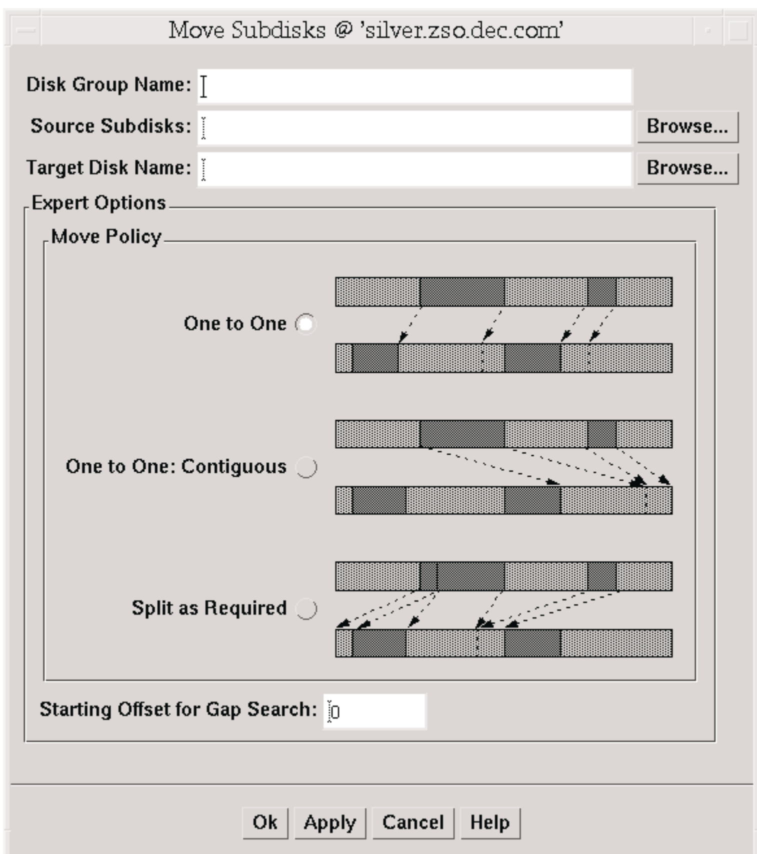
If this task fails and leaves some unused subdisks (that is, subdisks that are not associated with a volume) on the system, you can use the Remove Subdisk task to free the space occupied by the unused subdisks.

Follow these steps to move a subdisk:

1. In the Object Tree, select Volume and in the Object Table, select the volume with the subdisk to be moved.
2. From the Selected menu, choose Show Layout.
3. In the Volume Layout Details window, select the subdisk to be moved to another disk.

   a. From the Selected menu, choose Move.

      The Move Subdisks dialog box is displayed:



   b. In the Move Subdisks dialog box:

      Enter the name of the disk group that contains the subdisk to be moved.

If the correct source subdisk name is not displayed, enter the subdisk's name or click on Browse to select the subdisk.

Enter the name of the target disk to which the subdisk should be moved or click on Browse to select a disk.

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(s). The Split as Required option allows the subdisk(s) to be split if needed.

Specify the minimum disk offset for the subdisk. Enter the offset in the Starting Offset for Gap Search field.

c. Click on the Ok button.

4. Close the Volume Layout Details window. You can view the new volume layout by reopening the window.

9.5.3 Removing a Subdisk

You can remove a subdisk that is not associated with a volume. This returns the disk space occupied by unused subdisks to the free space pool.

Follow these steps to remove a subdisk:

1. In the Object Tree, select Disk and in the Object Table, select the disk with the subdisk to be removed.
2. From the Selected menu, choose Show Layout.
3. In the Volume Layout Details window, select the subdisk(s) to remove:
4. From the Selected menu, choose Remove.

   The Remove Subdisks dialog box is displayed:

In the Remove Subdisks dialog box:

Enter the name of the disk group that contains the subdisks to be removed.

If the correct subdisk names are not displayed, enter the subdisk names or click on Browse to select the subdisks.

Click on the Ok button.

5. Close the Volume Layout Details window. You can view the new volume layout by reopening the window.

9.5.4 Splitting a Subdisk

You can divide a subdisk into two or more subdisks. Once split, the smaller subdisks can be moved elsewhere or rejoined later. This 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 subdisk remains the same as the selected subdisk. 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.

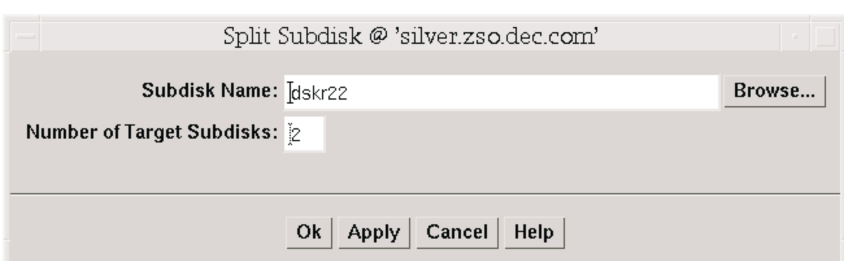
A log subdisk cannot be split.

Follow these steps to split a subdisk into multiple subdisks:

1. In the Object Tree, select Volume and in the Object Table, select the volume with the subdisk to be split.
2. From the Selected menu, choose Show Layout.
3. In the Volume Layout Details window, select the subdisk to be split into multiple subdisks.

   a. From the Selected menu, choose Split.

   The Split Subdisk dialog box is displayed:



   b. In the Split Subdisk dialog box:

   If the correct subdisk name is not displayed, enter the subdisk's name or click on Browse to select the subdisk.

Enter the number of subdisks into which the subdisk should be split. A subdisk can be split into two or more subdisks.

Click on the Ok button.

4. Close the Volume Layout Details window. You can view the new volume layout by reopening the window.

9.6 Volume Management

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.

AdvFS file domains (file systems) operate differently from UFS file systems. Because the volume is part of the AdvFS file domain, once you assign a volume to AdvFS, it is out of the control of the Storage Administrator. Therefore, you cannot stop, remove, rename, mount, or unmount an AdvFS volume. That is, you cannot use the Storage Administrator to perform tasks that compromises the integrity of the AdvFS file domain.

In a cluster, AdvFS file systems are supported in all modes; UFS file systems are supported in read-only mode.

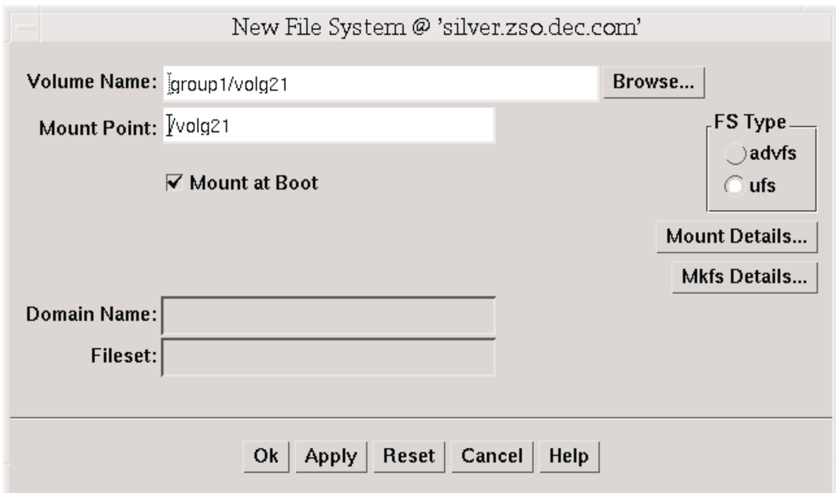
9.6.1 Adding a File System to a Volume

You can place a new file system on an existing volume and mount the file system. If Mount at Boot is selected, the `/etc/fstab` file is automatically updated.

Follow these steps to add a file system to an existing volume:

1. In the Object Tree, select Volume and in the Object Table, select the volume to contain the file system.
2. From the Selected menu, choose File System then New.

   The New File System dialog box is displayed:



3. In the New 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. Type 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 desired.

   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 Mkfs Details dialog box. Click on Help in the Mkfs 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 the Ok button.

9.6.2 Adding a Log to a Volume

You can add a log to a mirrored or RAID5 volume.

When you add a log to a mirrored volume, dirty region logging (DRL) is activated for that volume. DRL uses the log to track the regions of the volume that change due to I/O writes. If a system failure occurs, DRL uses the information in the log to recover only the portions of the volume that need recovery. This speeds up recovery time for mirrored volumes.

For DRL to be in effect, a mirrored volume must have at least one DRL log. You can create additional DRL logs (on different disks) to mirror the DRL information.

A RAID5 volume log speeds up the resynchronization time for RAID5 volumes after a system failure. A RAID5 log maintains a copy of the data and parity being written to the volume at any given time. If a system failure occurs, LSM can replay the RAID5 log to resynchronize the volume. This copies the data and parity that was being written at the time of failure from the log to the appropriate areas of the RAID5 volume.

You can create multiple RAID5 logs (on different disks) to mirror the log information. Ideally, each RAID5 volume should have at least two logs.

Follow these steps to add a log to a volume:

1. In the Object Tree, select Volume and in the Object Table, select the volume to contain the log.
2. From the Selected menu, choose Log then Add.

   The Add Log dialog box is displayed:

3. In the Add Log dialog box:

   a. If the correct volume name is not displayed, enter the volume name or click on the Browse button 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 the Ok button.

9.6.3 Adding a Mirror to a Volume

You can create a mirror (copy) of a volume on a disk that is not being used. Once mirrored, the data in the volume is redundant. If a disk fails, the data remains available on the surviving mirror. A volume can have multiple mirrors, but each must reside on a separate disk. Sufficient disk space must be available. You cannot mirror a RAID5 volume.

You can only use disks in the same disk group to create a new mirror. If no disks are assigned, LSM uses available disk space to create the mirror. Adding a mirror requires resynchronization, so this task may take some time.

A volume can contain up to 32 mirrors.

Follow these steps to add one or more mirrors to an existing volume:

1. In the Object Tree, select Volume and in the Object Table, select the volume to be mirrored.
2. From the Selected menu, choose Mirror then Add.

   The Add Mirror dialog box is displayed:

3. In the Add Mirror 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 the Layout button to:

      Specify the layout for the mirror: concatenated or striped and, if striped, unit size.

      Add more than one mirror and supply comments.
   c. Click on the Assign Disks button to place the mirror on a specific disk.

d. Click on the Ok button.

9.6.4 Disabling a Mirror in a Volume

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.

Once disabled, the mirror remains detached from its volume until you either reattach the mirror or restart the volume. If a volume only has two mirrors and one mirror is disabled, the volume is not redundant while the mirror is disabled.

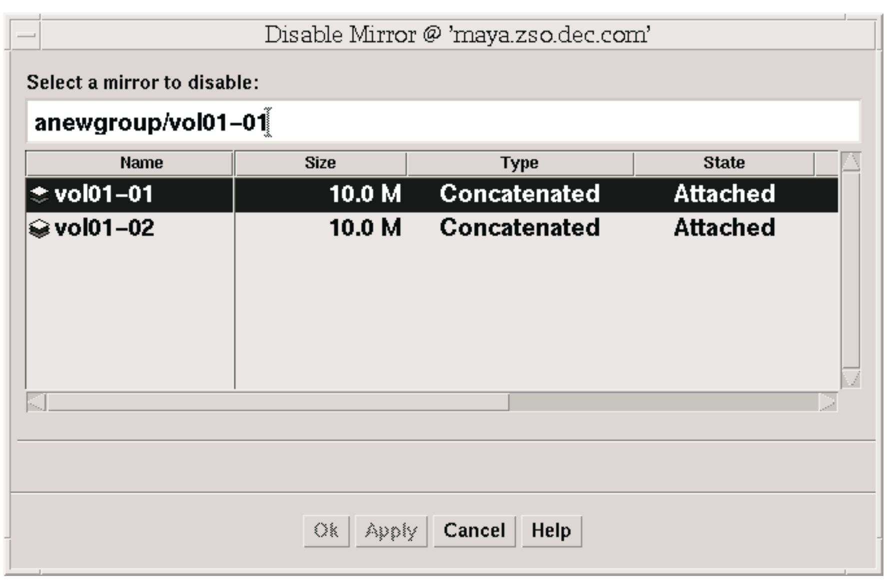
The last mirror in a volume cannot be disabled.

When a volume is restarted, any disabled (detached) mirrors are reattached to the volume automatically.

Follow these steps to disable a mirror in a volume:

1. In the Object Tree, select Volume and in the Object Table, select the volume that contains the mirror to be disabled.
2. From the Selected menu, choose Mirror then Disable.

   The Disable Mirror dialog box is displayed:



3. In the Disable Mirror dialog box:

   a. Select the mirror to be disabled.

   b. Click on the Ok button.

9.6.5 Repairing a Mirror in a Volume

You can repair a disabled mirror and reattach it to its volume. Repairing a mirror involves copying data from an active mirror on the volume to the mirror being attached. Once attached, the mirror is accessible for reads and writes. This task recovers the mirror so that it has the same contents as other mirrors in the volume.

Alert icons and the Alert Monitor window may provide you with information when a mirror needs to be repaired.

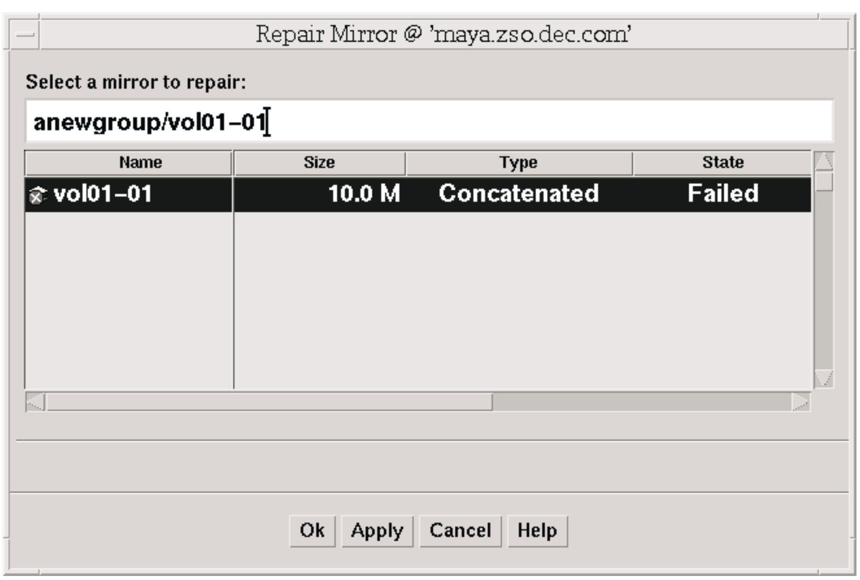
Depending on the amount of data in the volume, this task may take some time.

Follow these steps to repair a mirror:

1. In the Object Tree, select Volume and in the Object Table, select the volume that contains the mirror to be repaired.

2. From the Selected menu, choose Mirror then Repair.

   The Repair Mirror dialog box is displayed:



3. In the Repair Mirror dialog box

   a. Select the mirror to be repaired.

   b. Click on the Ok button.

9.6.6 Checking a File System on a Volume

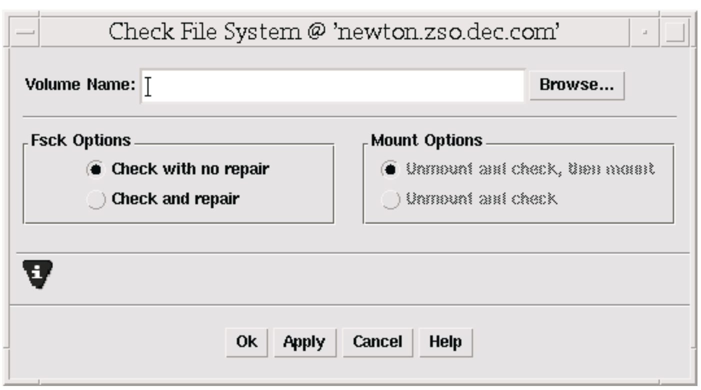
Checking a file system only applies to UFS file systems because it uses the `fsck` utility, which is not compatible with AdvFS file systems. You can check the file system with or without repairing it. Checking a file system may take some time.

If you are running a cluster, UFS file systems are mounted read-only, so the Storage Administrator cannot check the file system.

Follow these steps to check a UFS file system on a volume:

1. From the Object Tree, select File System and in the Object Table, select a UFS file system.
2. From the Selected menu, choose Check.

   The Check File System dialog box is displayed:



3. In the Check File System dialog box:

   a. If the correct file system name is not displayed, enter the file system name or click on Browse to select the file system.

   b. Choose the `fsck` option:

      Check with no repair

      Check and repair

   c. Choose the mount option:

Unmount and check then mount

Unmount and check

d. Click on the Ok button.

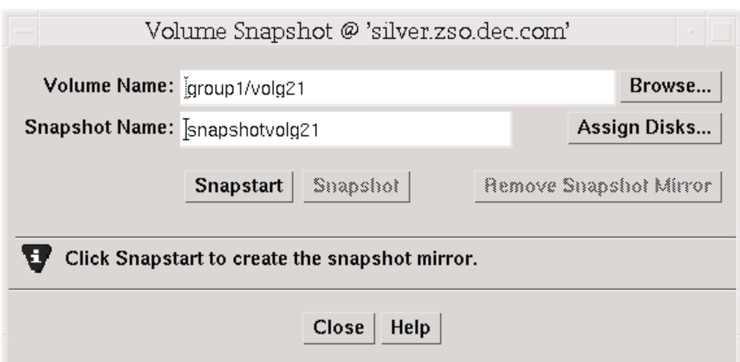
9.6.7 Creating a Copy of a Volume Data

You can create a shapshot (temporary mirror) of a volume that tracks volume activity. In a cluster, you cannot take a snapshot of the clusterwide root.

Follow these steps to create or stop a volume snapshot:

1. From the Object Tree, select Volume and in the Object Table, select a volume on which to collect data.
2. From the Selected menu, choose Snapshot.

   The Volume Snapshot dialog box is displayed:



3. In the Volume Snapshot dialog box:

   a. If the correct volume name is not displayed, enter the volume name or click on Browse to select the volume.

   b. If the correct snapshot name (volume) is not displayed, enter the volume name or click on Assign Disks to select the volume.

4. You can create the mirror, start, or stop it:

   - Click on the Snapstart button to create the snapshot mirror.
   - Click on the Snapstop button to stop the snapshot process and create a new volume to attach to the broken mirror so you can access it.
   - Click on the Remove Snapshot Mirror button to remove the volume that was created.

9.6.8 Creating a Volume

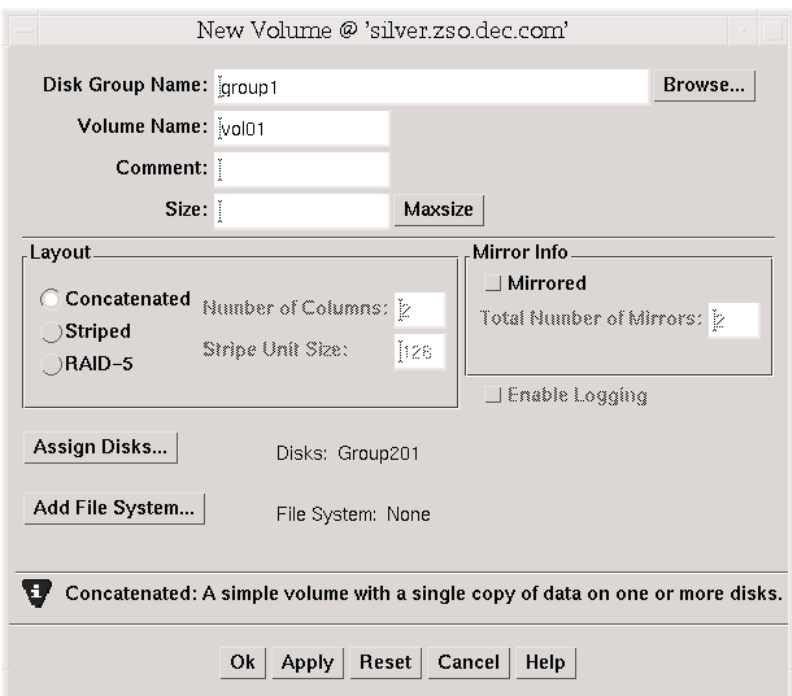
You can create a volume that is less than or equal to the available free space on the disk(s). If no disks are assigned, the Storage Administrator uses available space on disks in the selected disk group.

The data in a striped or concatenated volume is not protected against disk failure unless the volume is mirrored.

Follow these steps to create a volume:

1. From the Console menu, choose New then Volume.

   The New Volume dialog box is displayed:



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 determine 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.

   RAID5 - Enter the number of columns and stripe unit size. This option is not available in a cluster.

f. Check the Enable Logging box to create logging for the new volume.

g. If you have chosen a concatenated or striped volume, you can choose to mirror it.

   To mirror the volume, select Mirrored. In the Total Number of Mirrors field, type the total number of mirrors for the volume. Note that each plex is a mirror, so if you create a volume and one mirror of that volume, the total number of mirrors is 2.

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 the Ok button.

i. To place a file system on the volume, click Add File System.

j. In the Add File System dialog box:

   i. Type 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 will be created.

   ii. Select Mount at Boot if you want the `/etc/fstab` file automatically updated and the file system mounted at reboot.

   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 Mkfs Details dialog box. This option is not available in a TruCluster environment.

   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.

   Click on the Ok button to close the Add File System dialog box.

k. Click on the Ok button.

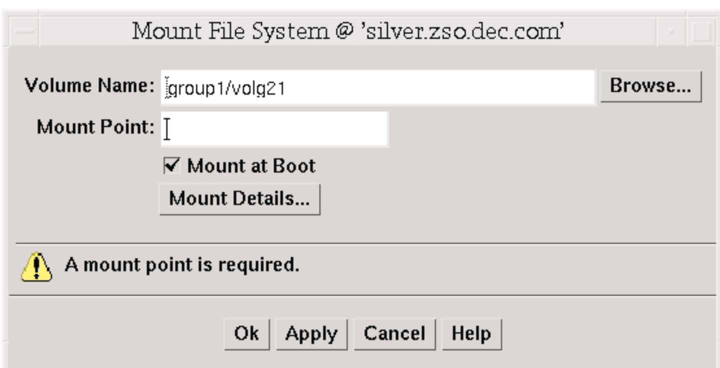
9.6.9 Mounting a UFS File System on a Volume

You can mount a UFS file system that already exists on a volume. If Mount at Boot is selected, the `/etc/fstab` file is automatically updated.

Follow these steps to mount a file system on an existing volume:

1. In the Object Tree, select Volume and in the Object Table, select the volume that contains the UFS file system to be mounted.
2. From the Selected menu, choose File System then Mount.

   The Mount File System dialog box is displayed:

3. In the Mount File System dialog box:

   a. If the correct volume name is not displayed, enter the volume name or click on the Browse button to select the volume.

   b. 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 desired.

   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 the Ok button.

Note

AdvFS filesets do not appear as mounted on volumes because AdvFS does not associate filesets with a specific volume.

9.6.10 Preparing to Restore a Volume from Backup

To reload 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 proceed. This task does not remount the file system.

This procedure is useful for disaster recovery. If a volume's data is corrupted and you need to restore the volume from backup, this procedure prepares the volume for restoration.

Follow these steps to prepare a volume to restore it from backup:

1. In the Object Tree, select Volume and in the Object Table, select the volume to be restored from backup.
2. From the Selected menu, choose Prepare For Restore.

   The Prepare Volume For Restore dialog box is displayed.
3. In the Prepare Volume For Restore 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 the Ok button.

9.6.11 Recovering a Volume

You can recover a volume. The recovery operations depend on the type of volume and include starting disabled volumes, resynchronizing mirrors in mirrored volumes, and resynchronizing parity in RAID5 volumes. After successful recovery, the volume should be available for use.

Alert icons and the Alert Monitor window may provide information when a volume recovery is needed.

In some cases, recovery may not be possible. If the volume recovery fails, you can attempt to restore the volume from backup.

Follow these steps to recover a failed volume:

1. In the Object Tree, select Volume and in the Object Table, select the volume to be recovered.
2. From the Selected menu, choose Recover.

   The Recover Volume dialog box is displayed.

3. In the Recover Volume dialog box:
   a. If the correct volume name is not displayed, enter the volume name or click on the Browse button to select the volume.
   b. Click on the Ok button.

9.6.12 Removing a Volume

Removing a volume destroys all of the data in that volume. Only remove a volume 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 will only work if the file system is UFS.

Follow these steps to remove a volume:

1. In the Object Tree, select Volume and in the Object Table, select the volume remove.
2. From the Selected menu, choose Remove.

   The Remove Volume dialog box is displayed.
3. In the Remove Volume dialog box, click on the Yes button to remove the volume.

9.6.13 Removing a Log from a Volume

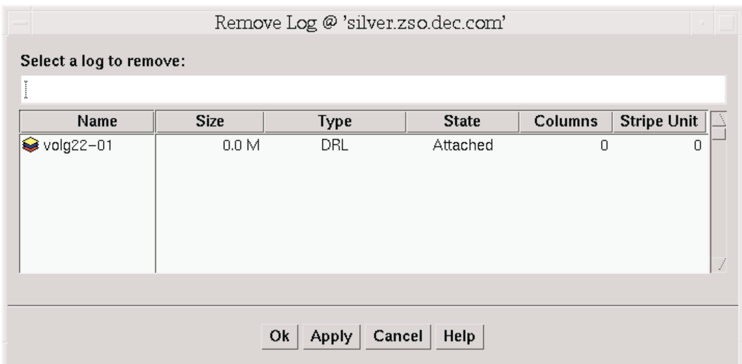
You can remove a DRL log or a RAID5 log from a volume.

If you remove a volume’s only log, logging (either DRL or RAID5 logging) is no longer in effect for that volume. If logging is disabled, recovery time increases.

Follow these steps to remove a log from a volume:

1. In the Object Tree, select Volume and in the Object Table, select the volume that contains the RAID5 or DRL log to be removed.
2. From the Selected menu, choose Log then Remove.

   The Remove Log dialog box is displayed:



3. In the Remove Log dialog box:

   a. If the correct volume name is not displayed, enter the volume name or click on the log table to select the volume.

   b. Click on the Ok button.

9.6.14 Removing a Mirror from a Volume

Removing a mirror for 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. However, this may leave the volume unmirrored and unprotected against disk failure.

If a volume only has two mirrors and one mirror is removed, the volume is no longer redundant. The last mirror cannot be removed from a volume, for that is equivalent to removing the volume.

Follow these steps to remove a mirror from a volume:

1. In the Object Tree, select Volume and in the Object Table, select the volume that contains the mirror to be removed.
2. From the Selected menu, choose Mirror then Remove. The Remove Mirror dialog box is displayed.
3. In the Remove Mirror dialog box:

   a. If the correct mirror name is not displayed, enter the mirror name.

   b. Click on the Ok button.

9.6.15 Renaming a Volume

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 allows you to specify a new mount point for the file system. You cannot rename volumes that are part of an AdvFS domain.

Follow these steps to rename a volume:

1. In the Object Tree, select Volume and in the Object Table, select the volume to be renamed.
2. From the Selected menu, choose Rename.

   The Rename Volume dialog box is displayed:

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 the Ok button.

9.6.16 Resizing a Volume

You can increase or decrease the size of a volume. If the volume contains a UFS file system, this procedure also resizes the file system, which destroys the data in it. A volume containing an unmounted file system cannot be shrunk.

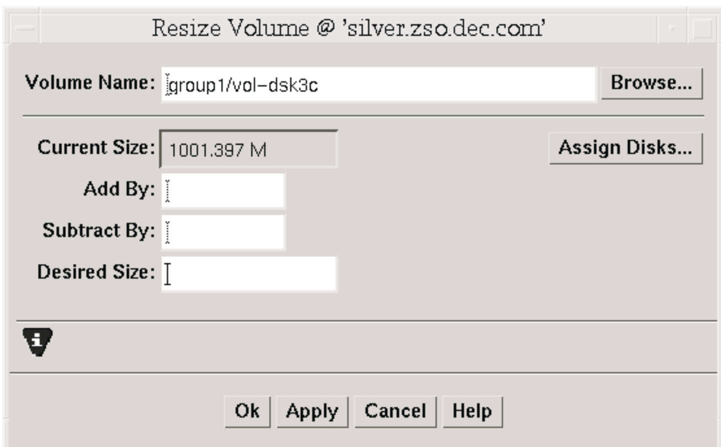
You cannot resize an AdvFS file domain with the Storage Administrator. If you want to resize a domain, use the AdvFS command line commands `addvol` and `rmvol`. See *AdvFS Administration* for more information.

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 disk pool. When the volume size is increased, sufficient disk space must be available. When increasing the size of a volume, LSM assigns the necessary new space from available disks.

Follow these steps to resize a volume:

1. In the Object Tree, select Volume and in the Object Table, select the volume to be resized.
2. From the Selected menu, choose Resize.

   The Resize Volume dialog box is displayed:



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 should be added to the volume.
      - To decrease the volume size by a specific amount of space, use the Subtract By field to specify how much space should be removed from the volume.
      - To specify the new volume size, type 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 the Ok button.

9.6.17 Starting a Volume

You can start a volume. RAID5 is not supported in a cluster. If you are not running a cluster, starting a RAID5 volume enables the volume and resynchronizes parity, if necessary. Starting 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.

Under normal circumstances, volumes are automatically started when the system reboots. You can restart a volume that you stopped manually or to attempt to restart a volume 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, it cannot be started using the procedure described below.

Follow these steps to start a volume:

1. In the Object Tree, select Volume and in the Object Table, select the (stopped) volume to be started.
2. From the Selected 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 the Browse button to select the volume.
   b. Click on the Ok button.

      The volume's state in the Object Table changes to Started.

9.6.18 Stopping a Volume

You can stop a volume. 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, it cannot be stopped using the procedure described below.

Follow these steps to stop a volume:

1. In the Object Tree, select Volume and in the Object Table, select the volume to be stopped.
2. From the Selected 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 the Browse button to select the volume.
    b. Click on the Ok button.

       The volume's state in the Object Table changes to Stopped.

9.6.19 Unmounting a File System on a Volume

You can unmount a mounted UFS file system.

Follow these steps to unmount a file system on a volume:

1. In the Object Tree, select Volume and in the Object Table, select the volume containing the file system to be unmounted.
2. From the Selected 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 the Ok button.

A

Customizing the Storage Administrator

The Storage Administrator is the Java-based graphical user interface (GUI) for LSM. The Storage Administrator displays a hierarchical view of LSM objects and their relationships. You use the Storage Administrator to view and manage LSM objects on a local or remote (client) system.

This appendix describes how to track Storage Administrator activities, how to use the Storage Administrator, and how to customize the Storage Administrator GUI. See Chapter 9 for more information on using the Storage Administrator to complete a particular task for an LSM object.

A.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 log ins
- A server log collects LSM startup information and error messages

By default, a log maintenance shell script called `/usr/lib/java/applications/lsmsa/logMaintenance` runs once a week to save and compress each log file. Compressed files are saved as `<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. Files that have the suffix `.11` are deleted. You can change the suffix number in the `root crontab` file.

A.1.1 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 located in `/var/lsmsa/logs/command`.

The following output shows a sample command log entry for a successful volume creation:

```
Create Volume
Description: Create Volume
User: root
Started: Tue Mar 09 12:07:22 PDT 1999
Finished: Tue Mar 09 12:07:24 PDT 1999

State: Successful
Executed Commands:

/usr/sbin/volassist
-g rootdg make vol04 4m layout=striped stripeunit=128 ncolumn=2
```

The following output shows a sample command log entry for a failed volume creation:

```
Create Volume  FAILED!
Description: Create
VolumeUser: root
Started: Tue Mar 09 12:07:50 PDT 1999
Finished: Tue Mar 09 12:07:51 PDT 1999
State: Failed

Executed Commands:
/usr/sbin/volassist
-g rootdg   make vol05 8g layout=striped stripeunit=12 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
```

A.1.2 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 located in `/var/lsmsa/logs/access`.

The following output shows a sample access log file entry:

```
Mon Apr 05 12:07:22 PDT 1999: user rssn login succeeded

Mon Apr 05 12:22:24 PDT
1999; user jehg login failed with error *User password invalid*
```

Entries for failed access may be logged multiple times due to a security requirement.

A.1.3 Server Log File

The server log tracks LSM startup information and server errors. By default, the server log file is `located in` `/var/lsmsa/logs/server.log`.

The following output shows sample server log file entries:

```
Starting Compaq Storage Administrator RMI Registry
Starting Compaq Storage Administrator Command Server
```

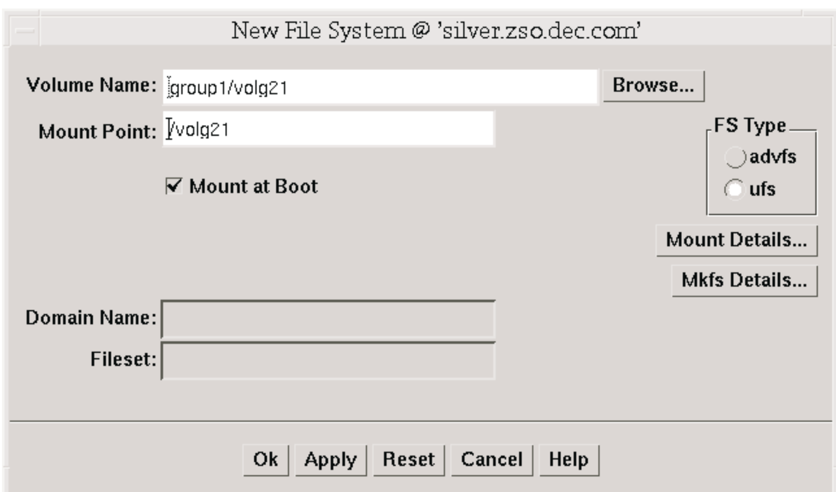
```
Starting Compaq Storage Administrator Server
Fri Mar 12 11:22:21 PST 1999
security enabled
rebinding ....
rebound

 //maya.zso.dec.com:2410/vrts.remote.vrtsServer
```

A.2 Working with Dialog Boxes

Storage Administrator displays dialog boxes in which you provide information as shown in Figure A–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 A–1: Sample 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.

A.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 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 the Browse button 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.

A.2.2 Specifying Object Sizes in Dialog Boxes

The following table shows 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 as described in Section A.3.4. By default, sizes are displayed in kilobytes unless you specify otherwise.

A.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.

A.3.1 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 +), 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.

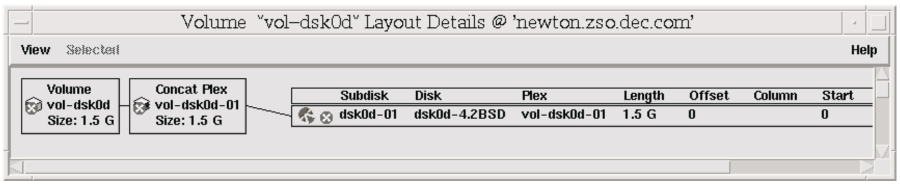
  For example, to display the volumes in a disk group listed in the Object Table, double-click on the disk group name, then double-click on Volumes. All volumes in the disk group appear in the Object Table.
- It may be useful to view a copy of the Object Table to look at different objects, for example, 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 continues to update.

A.3.2 Volume Layout Details Window

To display the Volume Layout Details window for a volume, highlight the volume in the Main window Object Table and choose Show Layout from the Selected menu.

The Volume Layout Details window displays a graphical view of the selected volume's layout, components, and properties, as shown in Figure A–2.

Figure A–2: Sample Volume Layout 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. Choose Update from the View menu to refresh the display. To change the volume displayed, choose Open from the View menu and specify another volume in the OpenVolume dialog box.

The View menu changes the display of the Volume Layout Details window. 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.

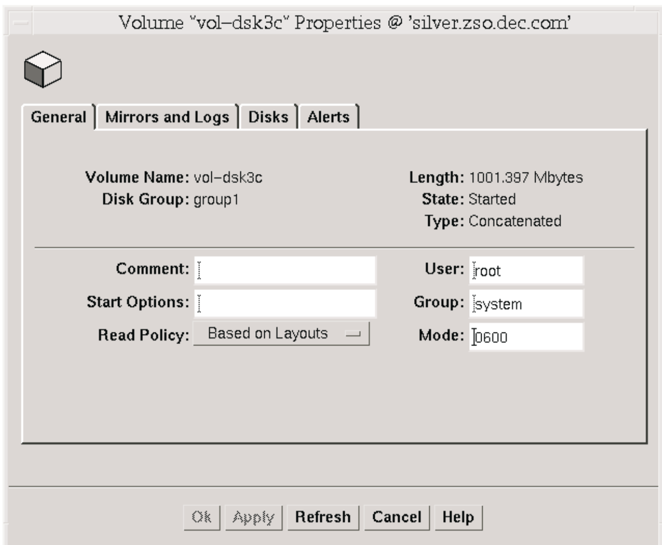
Right-clicking on an object to display context-sensitive pop-up menus.

A.3.3 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 Selected 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 A–3.

Figure A–3: Sample 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 the Help button for a detailed description of the Properties dialog box fields.

To change items in the Properties dialog box, make the changes, then click on the Ok button. This changes the settings for *all* properties tabs in the Properties dialog box.

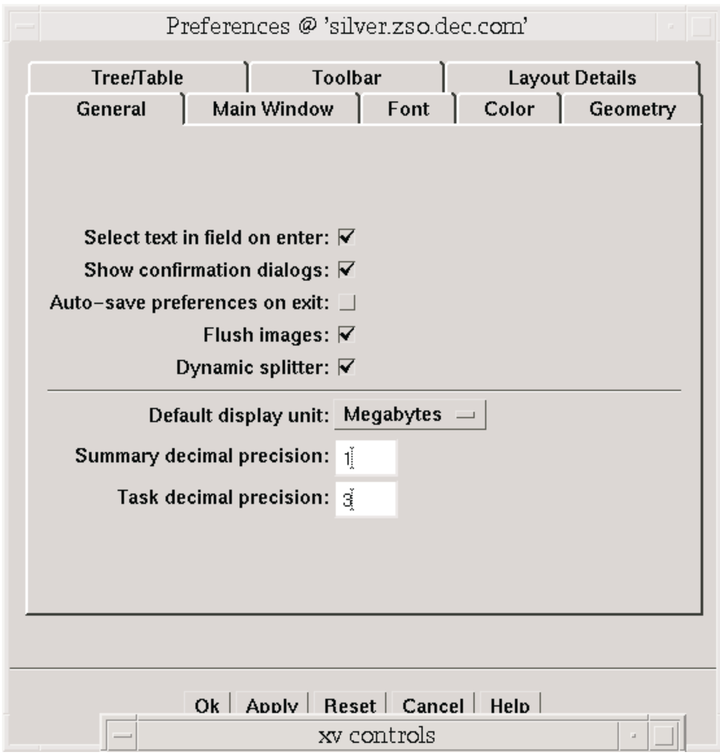
To update the contents of the Properties dialog box to reflect current properties for the object, click on the Refresh button. If you select a different object while a Properties dialog box is open, the contents of the dialog box does not change to reflect the new object selected. You must choose Properties again and open another dialog box.

A.3.4 User Preferences Dialog Box

You can change the way items appear in the Storage Administrator Main window and other windows. The Preferences dialog box contains a set of tabbed pages that display preference options for a particular aspect of Storage Administrator as shown, in Figure A–4. You can customize settings for a single Storage Administrator session or save the settings for future sessions.

To display the Preferences dialog box, from the Options menu, choose Preferences or click the Preferences button on the toolbar.

Figure A–4: Sample Preferences Dialog Box



To change preference settings, make the appropriate selections in the dialog box, then click on the Ok button. This changes the settings for *all* tabs in the Preferences dialog box. To reset the values for all tabs to the previous settings, click on the Reset button before you click on the Ok button.

When you change preference settings, an asterisk appears on the tabbed page that contains changes. This icon disappears when you click on the Ok, Apply, or Reset button. When you click on the Apply or Reset button, an asterisk also appears in the Help bar status area.

Unless you save your preferences, changes only apply to the current session. To save your settings, choose Save Preferences from the Options menu and click on the Save button in the toolbar or 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 auto-save preference is set, Storage

Administrator saves all preference settings when you exit the Storage Administrator session.

A.3.4.1 General Preferences

The General tab window 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 may 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

  Redraws 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.

A.3.4.2 Main Window Preferences

The Main Window tab window sets the preferences for:

- Show Status Bar

  Show/hide 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 and hide the Command Launcher by checking and 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.

A.3.4.3 Font Preferences

The Font tab window 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.

A.3.4.4 Color Preferences

The Color tab window 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.

A.3.4.5 Geometry Preferences

The Geometry tab window 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, the new size is reflected in the Geometry preference for that window.

A.3.4.6 Object Tree/Object 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 a Object Table.
- Selector Table Width

  Sets the width (in pixels) of the Object Table for Browse dialog boxes that only contain an Object Table.
- Visible Selector Rows

  Sets the number of rows displayed in the Object Tree and Object Table in Browse dialog boxes.

A.3.4.7 Toolbar Preferences

The Toolbar tab window 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.

A.3.4.8 Layout Details Preferences

The Layout Details window 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.

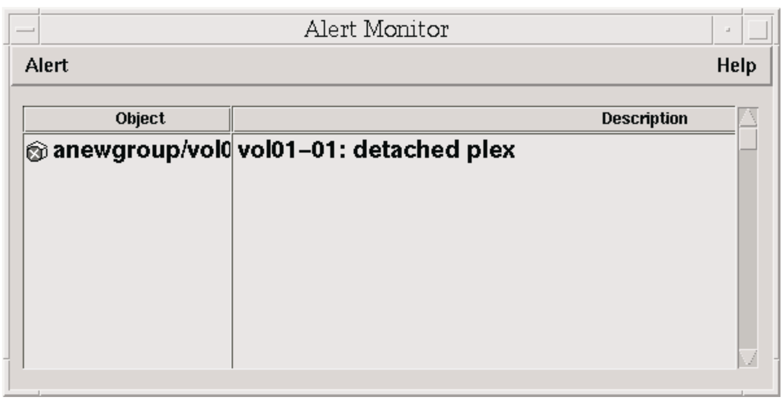
- Subdisk Projection

  When a subdisk is selected, highlights other subdisks on the same disk.

A.3.5 Alert Monitor Window

The Alert Monitor window displays information about failed objects or objects that experienced other errors, as shown in Figure A–5. 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 A–5: Sample Alert Monitor Window



To display the Alert Monitor window either:

- Click the Alert button 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.

A.3.6 Object Table Copy Window

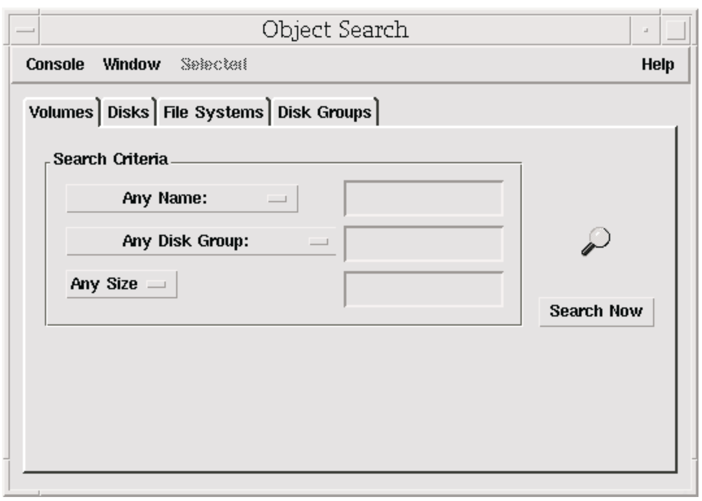
A copy of the Object Table allows you to view different parts of your system at the same time. The windows are dynamic, so updates to the system are reflected in all windows. To display the window copy, choose the Table button from the toolbar or from the Window menu choose Copy Object Table

A.3.7 Search Window

The Search window searches the system for objects that match the specified search criteria. The Search window contains a set of tabbed pages that display search options for a particular type of object, as shown in Figure A–6. You can select the type of objects to search for by clicking on the tab label. The search will only take place on objects of the type you have selected.

To display the Search window, click the Search button in the toolbar or from the Window menu choose Search.

Figure A–6: Sample Search Dialog Box



Specify the search criteria by choosing the drop-down menu selection then entering your criteria. If you enter criteria in more than one box, the search results only reflects the items that match all of the criteria (Boolean AND).

The table in the bottom half of the 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 Search window are monitored and removed from the window if they no longer meet the current search criteria.

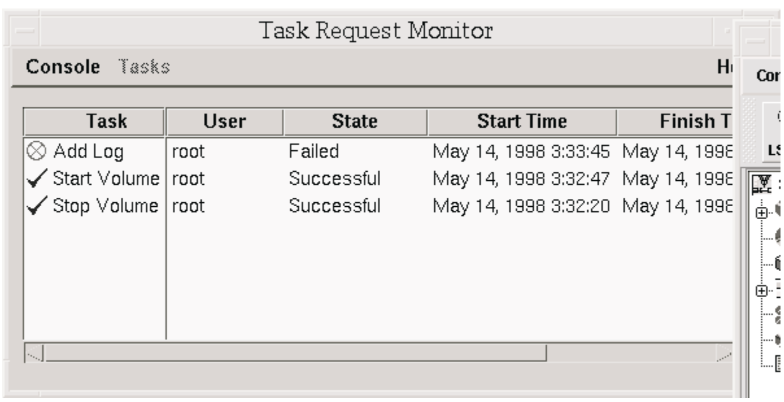
The Search window has menus similar to those in the Main window menu. The Window menu opens other windows or a copy of the current search-results table. The context-sensitive Selected menu accesses tasks or properties for a object selected in the table. You can right-click on an object to access a context-sensitive pop-up menu. To close the Search window, choose Close from the Console menu.

A.3.8 Task Request Monitor Window

The Task Request Monitor window shows LSM and other tasks that the Storage Administrator performed in the current session (and any other sessions running on the system) as shown in Figure A–7. Each task is listed with properties such as the user who performed the task, the task status, and the start and finish times. For failed tasks, the Task Properties window includes any relevant error messages.

To display the Task Request Monitor, click on the Task button (in the toolbar) or from the Window menu choose Tasks.

Figure A–7: Sample Task Request Monitor Window



To remove finished tasks and to close the window, from the Console menu choose Remove Finished Tasks.

To view the low-level commands used to perform a task, choose Properties from the Tasks menu. You can copy commands from the Executed Commands field of the Tasks Properties dialog box to the command line or to a script file.

A.4 Shortcuts and Other Operations

There are shortcuts that let you more efficiently perform operations with the GUI.

A.4.1 Sort

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 Search window, and the Task Request Monitor window.

A.4.2 Clearing an Alert

To acknowledge and clear an alert icon displayed on the Status bar, choose Clear Alert Status from the Options menu.

A.4.3 Keyboard Shortcuts

You can use the following keyboard shortcuts instead of menu commands:

- Ctrl-Shift-V

  Create a volume.
- Ctrl-G

  Create a disk group.
- Ctrl-F

  Create a file system.
- Ctrl-Z

  Resize an object.
- Ctrl-N

  Rename an object.
- Ctrl-Shift-R

  Remove an object.
- Ctrl-P

  Show the properties of an object.
- Ctrl-L

  Show the layout of a volume (graphical view)

The following shortcuts only work in the Main window:

- Ctrl-R

  Open the Task Request Monitor window.
- Ctrl-A

  Open the Alert Monitor window.
- Ctrl-S

  Open the Search window.
- Ctrl-C

Close the window.

A.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.

To separate or attach the Command Launcher and the Main window, choose Preferences from the Options menu. In the Preferences dialog box, choose the Main Window tab and click on Dock Command Launcher. To separate the Command Launcher, click on the Dock Command Launcher again to remove the check.

B

The Visual Administrator

The Visual Administrator, also called dxlsm, is the graphical user interface for the LSM software. This interface is designed primarily for disk and volume operations, but also provides a limited set of file system operations.

This appendix provides an overview of Visual Administrator features and use.

B.1 Starting the Visual Administrator

To start the LSM Visual Administrator, you must be logged into an account that has root privileges.

To start dxlsm, enter:

```
# dxlsm
```

The system displays the following message in a pop up window:

```
dxlsm is coming up, please wait.
```

When dxlsm starts, two LSM windows display:

- The main LSM Visual Administrator window
- The View of rootdg window

B.1.1 The Visual Administrator Main Window

The Visual Administrator main window contains a menu bar and a set of buttons as shown in Figure B–1. If you have RAID subsystems installed, the main window displays an additional Subsystems button.

Figure B–1: Visual Administrator Main Window

To display and manipulate different parts of the physical and logical storage systems, click on the View button. Each view window title includes the name of the machine on which the session is running. The main window contains a button for every view on the system.

The Visual Administrator has two types of views: default and user-created. You cannot remove or rename default views.

B.1.1.1 Default Views

From the main window, click on the following view buttons to access the default view window:

View Buttons	Window	Access
Disks	View of Disks	Displays all physical disks on the system
Volumes	View of Volumes	Displays all volumes, plexes, and associated subdisks on the system
World	View of World	Displays everything on the system including physical and LSM disks, volumes, and other objects
rootdg	View of rootdg	Displays everything in the default disk group, `rootdg`, including LSM disks, volumes, and other objects

B.1.1.2 User-Created Views

A user-created view is a view window defined to focus on a part of a system. For example, you can create a view window for each disk group. Create a new view window with the Views menu on the main window. Creating this window will place a new button on the window. Once the view is created, you can add icons by selecting an icon from another view and either using the Icon menu Create Icons option or dragging and dropping the selected icon.

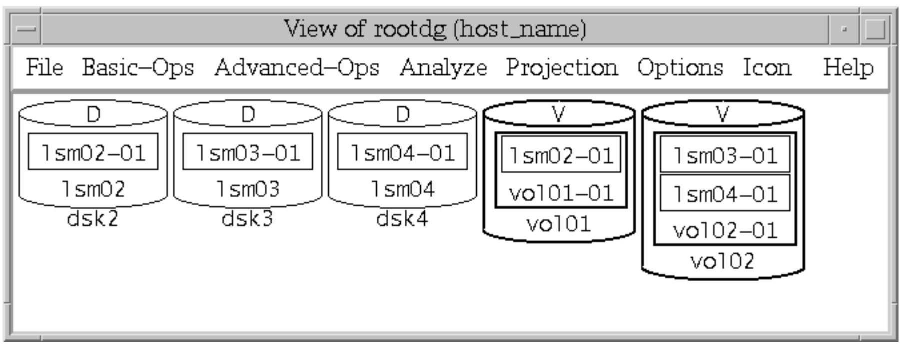
User-created views thus contain copies of icons from default views. Operations performed on these icon copies are reflected in the default views that display the affected icons. However, icons that appear in user-created views are not always updated when the icons are altered in the default view.

B.1.2 The View of rootdg Window

Immediately after the main window appears, the View of rootdg window appears. This view displays icons representing everything that is in the `rootdg` disk group. Whenever possible, perform operations in the View of rootdg window or in another disk group view.

Figure B–2 shows a view of a rootdg window that contains two volumes.

Figure B–2: View of rootdg



B.2 Mouse Buttons

A two- or three-button mouse is required to use the Visual Administrator. Table B–1 describes the mouse buttons. Unless otherwise stated, all directions to select or click on an item refer to the left mouse button (MB1). Right-click refers to the right mouse button for both two- and three- button access.

Table B–1: Default Mouse Buttons

Virtual Mouse Button	3-Button Access	2-Button Access	Function
MB1	Left	Left	Selects a single icon.
MB2	Middle	Ctrl -Left	Selects either one or multiple icons simultaneously.
MB3	Right	Right	Displays either the properties form or the analysis statistics form for that object, depending on whether the icon is undergoing analysis.
Shift -MB1	Shift -Left	Shift -Left	Toggles between minimizing and maximizing an icon.

Table B–1: Default Mouse Buttons (cont.)

Virtual Mouse Button	3-Button Access	2-Button Access	Function
Shift -MB2	Shift -Middle	Ctrl -Right	Toggles between starting and stopping projection on the selected icon.
Shift -MB3	Shift -Right	Shift -Right	Displays the properties form for the object, regardless of whether analysis is in effect.

An icon can be deselected by positioning the pointer over that icon and clicking MB2. This works regardless of which mouse button was used to select the icon.

Note

The examples in this document assume that you are using a three-button mouse, set up according to Table A-1. It is possible to redefine mouse buttons (using the `xmodmap` command, for example). See your X Window System documentation for details.

B.3 Icons for LSM Objects

The Visual Administrator interface uses icons to represent LSM objects (volumes, plexes, subdisks, and disks.)

Disk groups are represented as view windows rather than icons. The icons representing LSM disks, volumes, and other objects belonging to a particular disk group are all displayed within the view of the disk group.

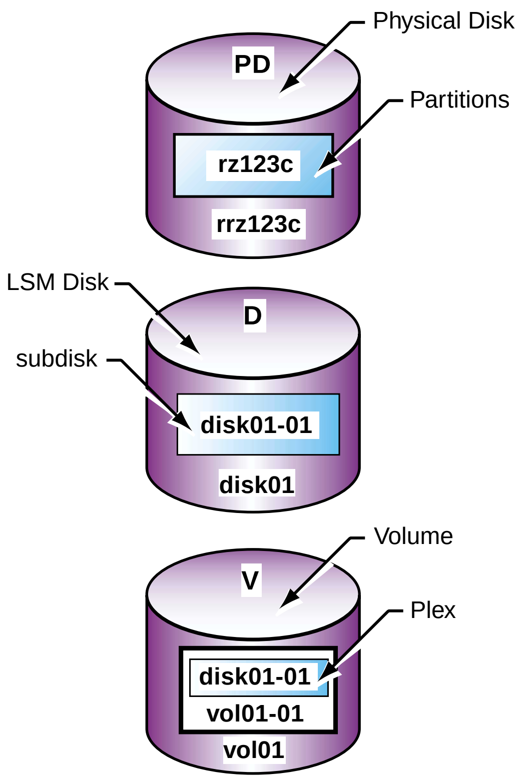
Note

Icons representing all elements of the system are displayed in the View of World window accessed with the World button on the Visual Administrator window.

B.3.1 Icon Characteristics

Figure B–3 shows how the Visual Administrator represents LSM objects.

Figure B–3: Icons That Represent LSM Objects



With some operations, icons are updated almost instantly to reflect the results of the operation just performed. During other operations, it may take time for a particular icon to update itself. While being updated, icons are prevented from accepting input or undergoing configuration changes. Inaccessible icons are grayed out.

B.3.2 Manipulating Icons

There are two ways to manipulate icons:

- Select-Operate. With this option you select an icon representing an LSM object (click on the icon or click MB2 on multiple icons) and perform the desired operation on that object by selecting from window menus.
- Drag and Drop. With this option you drag the icon of the selected object (hold down MB1 and move the mouse until the outline of the icon reaches the desired location) and drop it (release the mouse) elsewhere, such as on another object in a view window.

  When dropping an icon onto another icon, the dragged icon must be positioned so that the pointer (in the image of a hand) is directly over an unobscured portion of the icon on which it is to be dropped. Table B–2 describes drag and drop operations:

Table B–2: Drag and Drop Operations

Icon Type	Drop Location	Action
Free subdisk	View window	Creates a plex and associates the subdisk with the plex.
Free subdisk	LSM disk	Creates an identical subdisk on the disk.
Free subdisk	Plex	Associates the subdisk with the plex.
Associated subdisk	Free subdisk	Swaps the associated subdisk with the free subdisk. The free subdisk becomes associated and replaces the original subdisk, which is removed.
Associated subdisk	LSM disk	Creates an identical free subdisk on the LSM disk, then swaps the associated subdisk with the new free subdisk. The free subdisk becomes associated and replaces the original subdisk, which is removed.
Associated subdisk	View window	Dissociates the subdisk.
Associated plex	View window	Dissociates the plex.
Dissociated plex	User's view window	Copies the plex icon to the user's view.
Plex	Volume	Associates the plex to the volume.
Disk	User's view window	Copies the physical or LSM disk icon to the user's view.
Volume	User's view window	Copies the volume icon to the user's view.
LSM disk	Disk group view	Adds an LSM disk (corresponding to the slice, simple, or nopriv disk) to that disk group.

B.4 Pull-Down Menus

The Visual Administrator provides pull-down menus that provide access to various Visual Administrator features.

Menus are located in the menu bar just below the window's title.

B.5 Forms

The Visual Administrator uses forms to present textual information. These forms also provide useful information about existing objects and configurations.

There are two types of forms:

- General forms usually appear during menu-selected operations or setup requests and accept or require user input.
- Properties forms display detailed information about a specific object's characteristics, some of which can be modified directly. Access properties forms by clicking MB3 on the chosen icon. (If the icon is undergoing analysis, use Shift-MB3).

B.5.1 Fields

Many forms require information in order to proceed with an operation. If a required field in the form is either left blank or is incorrect, an error will result. Other fields already contain information (such as default values), which you can either alter or accept. Yet other fields are read-only; these fields beep if you attempt to change them.

B.5.2 Form Error Messages

Error messages are displayed if you select Apply with incorrect fields on a form. A message is printed at the bottom of the form just above the buttons, and you can correct the values for those fields. If the error cannot be corrected or the operation is no longer desired, select Cancel.

B.6 Error and Warning Messages

The Visual Administrator uses dialog boxes to present error or warning messages. When a message is displayed, you cannot proceed until you select one of the buttons displayed in the error dialog box. Some warning boxes announce that a prerequisite is not met and require you to acknowledge this by clicking the Continue button before reattempting the operation.

B.7 Help Windows

You can access online help text from the menu bar of the main window and from the view windows. Help text is also available through the Help option in submenus or through the Help button at the bottom of forms. The Help window contains information about the current window, menu, form, or operation.

At the bottom of each Help window is a SEE ALSO area that lists related Help topics. To access any of the listed Help topics, click on the appropriate words in the SEE ALSO list. The Help facility keeps track of the order in which Help topics are visited, so you can move between topics by selecting Previous or Next from the menu bar.

The Help menu in the menu bar of the Help window itself provides access to the following information:

General Help	Accesses Help text that includes general information on the Visual Administrator Help facility and how it is used.
Help Index	Accesses a complete listing of the available Help topics arranged in logical groupings. Once a topic is identified from this list, that topic can be directly accessed from the SEE ALSO section of this Help window, which lists all topics alphabetically.

To close the Help window, select the Close option from the File menu. The record of help topics visited is retained.

C

Using the Visual Administrator

This appendix shows you how to complete common LSM management tasks using the Visual Administrator.

C.1 Volume Management

The following sections provide information on menus and forms relating to volume management.

C.1.1 Volume Menus

Both the Basic-Ops and Advanced-Ops menus provide access to volume-related menus. Most menus provide a Help selection, which contains information about the items and operations listed in that particular menu.

C.1.1.1 Basic-Ops Menu

You access the Basic-Ops menu by selecting:

Basic-Ops → Volume Operations

This menu provides access to volume operations involving general volume maintenance. These operations use the automated approach to volume management.

The Volume Operations menu provides the following selections:

- Create
- Remove Volumes Recursively
- Add Plex
- Remove Plex
- Resize
- Snapshot
- Help

The following list describes these menu selections:

- Create

Basic-Ops → Volume Operations → Create

This operation creates a simple or striped volume on one or more disks. You can select one or more disks on which to create the volume (providing that there is sufficient space on the disks). If no disks are specified, the LSM software automatically determines the disks to use based on available free space.

From the Create menu, you select the type of volume to be created from a submenu listing two of the basic types of volumes:

Type	Description
Simple	Creates a simple, concatenated volume whose subdisks are arranged both sequentially and contiguously within a plex.
Striped	Creates a volume with data spread fairly evenly across multiple disks by way of striping. *Stripes* are relatively small, equally-sized fragments that are allocated alternately to the subdisks of each plex.

To create a mirrored volume, create a simple or striped volume, then mirror it using the Add Mirror option.

`Requirements:`

- Only disks in the same disk group can be selected.
- Only LSM disks (disks under LSM control and assigned to a disk group) can be selected.
- If striping is to be in effect, at least two disks are required in order for the operation to succeed.

`Forms:` Simple Volume/FS Create Form and Striped Volume/FS Create Form.

- Remove Volumes Recursively

Basic-Ops → Volume Operations → Remove Volumes Recursively

This operation removes the selected volumes and deallocates all of the disk space set aside for that volume. It automatically removes all underlying plexes and subdisks associated with the volume.

Note

This is a permanent operation and cannot be undone. If completed, it will be difficult or impossible to retrieve the data associated with that volume. For this reason, a confirmation window is presented if the selected volume is not ready for removal (that is, started or enabled).

`Requirements:`

– At least one volume icon must be selected.

– The selected volumes cannot contain a mounted file system.

- Add Plex

  Basic-Ops → Volume Operations → Add Mirror

  This operation adds a plex to the selected volume by associating a plex of the correct length to the volume. The plex effectively duplicates the information contained in the volume. Although a volume can have a single plex, at least two are required for mirroring.

  From the Add Mirror menu, you select the type of plex to be added from a submenu listing two of the basic types of plexes:

Type	Description
Simple	Adds a simple, concatenated plex whose subdisks are arranged both sequentially and contiguously.
Striped	Adds a plex whose data is allocated evenly across each of its subdisks in an alternating fashion. This is accomplished with *stripes*, which are relatively small, equally-sized fragments that are allocated alternately to each subdisk.

  You can select disks for this operation. However, the number of selected disks must be sufficient to accommodate the layout type of both the existing volume and the plex to be added. If no disks are selected, the free space for the plex is allocated by the LSM software.

  `Requirements:`

  – A volume icon must be selected.

  – For a striped plex, at least two disks other than those already in use by the volume must be available.

- Remove Plex

  Basic-Ops → Volume Operations → Remove Mirror

  This operation removes the selected plex, along with any associated subdisks.

  `Requirements:`

  – A plex icon must be selected.

  – The last valid plex in a started or enabled volume cannot be removed.

- `Resize`

  Basic-Ops → Volume Operations → Resize

This operation resizes the selected volume. The volume can be increased to, increased by, reduced to, or reduced by a given length. This involves adding or removing disk space to or from the plexes associated with the volume.

If new disk space is needed during the resize, it is allocated as necessary; if space becomes unused, it is added to the free space pool.

`Requirements:`

– A volume icon must be selected.

– A volume containing a mounted file system cannot be shrunk.

`Form:` Volume Resize Form

• Snapshot

Basic-Ops → Volume Operations → Snapshot

This operation backs up a volume by creating a snapshot image of that volume. This is a convenient way of performing backup with minimal interruption.

This operation invokes the LSM snapshot approach in which the snapshot operation creates a new volume that is a snapshot of an existing volume. This is done by creating a plex of the existing volume (creating and associating a plex) using disk space from the pool of free disk space. The plex is brought up to date (this may take some time) and a separate (snapshot) volume is then created for that plex. The snapshot volume represents a consistent copy of the original volume at the time the snapshot was begun. The snapshot volume can be used to make a backup of the original volume without stopping it. After the backup is made, you can remove the snapshot volume without losing any data.

Note

For UFS volumes, it is recommended that you unmount the file system briefly to ensure the snapshot data on disk is consistent and complete.

From the Snapshot menu, a submenu allows you to first create the snapshot plex and then the snapshot volume:

Option	Description
Snapstart	Start the snapshot procedure by creating a snapshot plex within the volume to be backed up. It takes a variable amount of time to update the new plex, during which time the snapshot plex icon is grayed out.
Snapshot	At a convenient time (preferably after warning users to suspend activity briefly), create another volume for the snapshot plex. This portion of the procedure should take only seconds to complete.

`Requirements:`

- A volume icon must be selected.
- There must be sufficient free disk space to accommodate the snapshot volume.

`Form:` Snapshot Form

C.1.1.2 Advanced-Ops Menu

You access the Advanced-Ops menu selections by selecting:

Advanced-Ops → Volume

This menu provides access to assorted volume operations. These volume operations use the manual approach to volume management. The Volume menu provides the following selections:

- Create
- Remove Volumes
- Initialize Volumes
- Start Volumes
- Stop Volumes
- Resynchronize Volumes
- Set to Maint State
- Recover Volumes
- Help

The following list describes these menu selections:

- Create

  Advanced-Ops → Volume → Create

  This operation creates a volume. You can select one or more plexes to be associated with the new volume after creation.

`Form`: Volume Create Form

- Remove Volumes

  Advanced-Ops → Volume → Remove Volumes

  This operation removes the selected volumes. If the selected volume is started, it must be stopped before it can be removed.

  Note

  This is a permanent operation and cannot be undone. Any plexes associated with the volume will be disassociated and left behind.

  `Requirements`:

  – At least one volume icon must be selected.

  – The volume must be stopped before it can be removed.

- Initialize Volumes

  Advanced-Ops → Volume → Initialize Volumes

  This operation initializes the selected volumes.

  From the Initialize volumes menu, you select the type of initialization from a submenu listing the following choices:

Option	Description
Active	This enables the selected volume and its associated plexes, and sets the state of all associated plexes to ACTIVE.
Enable	This enables the selected volume and its associated plexes, but leave the plex states as EMPTY.
Clean	This sets the state for all associated plexes of the selected volume to CLEAN. This can be applied only under limited circumstances.
Zero	This enables the selected volume and its associated plexes, then write zeroes over the entire volume. After the operation completes, all associated plexes are set to ACTIVE, assuming that there are no I/O errors.

  `Requirements`:

  – At least one volume icon must be selected.

  – The selected volume cannot have been previously initialized.

  – The selected volume should have at least one associated plex that is complete (or contiguous).

- Start Volumes

  Advanced-Ops → Volume → Start Volumes

  This operation starts the selected volumes. A volume must be started before it can be accessed.

  From the Start volumes menu, a submenu allows you to indicate whether all volumes or just those selected should be started:

Option	Description
Start	Start the selected volume, which must be startable.
Start All	Start all volumes in this disk group that can be started.

  Requirements:

  – At least one volume icon must be selected for the Start operation. No volume icons need to be selected for the Start All operation.

  – A volume should be initialized before it can be started.

- Stop Volumes

  Advanced-Ops → Volume → Stop Volumes

  This operation stops the selected volumes. A volume that is stopped is inaccessible.

  From the Stop volumes menu, a submenu allows you to indicate whether all volumes or just those selected should be stopped:

Option	Description
Stop	Stop the selected volume.
Stop All	Stop all volumes in this disk group.

  Requirements:

  – At least one volume icon must be selected for the Stop operation. No volume icons need to be selected for the Stop All operation.

  – A volume must be started before it can be stopped.

  – A volume that is in use or contains a mounted file system cannot be stopped.

- Resynchronize Volumes

  Advanced-Ops → Volume → Resynchronize Volumes

  This operation brings all plexes within the selected volumes up to date. Any plexes that are inconsistent are resynchronized to contain consistent data.

This operation may take some time depending on how large the plexes are and whether or not logging is enabled.

Requirements:

– At least one volume icon must be selected.

– The selected volumes must be started.

- Set to Maintenance State

  Advanced-Ops → Volume → Set to Maint State

  This operation sets the state of the selected volumes to a maintenance state. Refer to the volume(8) reference page for information on the maintenance state.

  Requirement: At least one volume icon must be selected.

- Recover Volumes

  Advanced-Ops → Volume → Recover Volumes

  This operation recovers the selected volumes. At least one volume icon must be selected.

C.1.2 Volume Forms

Some volume operations result in the appearance of forms, which must be completed in order for that operation to proceed. Most forms provide a Help button, which contains information relevant to the fields and other aspects of that particular form.

C.1.2.1 Basic-Ops Forms

The following forms are accessed via volume-related selections from the Basic-Ops menu:

- Simple Volume/FS Create Form

  Basic-Ops → Volume Operations → Create → Simple

  This form creates a concatenated volume and optionally creates a file system on the new volume. The form is divided into two sections, one for volume creation and the other for file system creation. Most of the form fields are already set to the defaults for the creation of a new volume; the file system fields are grayed out because the default is not to add a file system to the volume. The following tables describes the fields for this form.

  Most fields in this form are required; those that are optional are listed here. All fields in this form are read/write fields.

Field	Description
Volume name:	The name of the volume to be created. The name must be unique within this disk group. The maximum length of this field is 31 characters.
Volume size:	The desired volume size. The size should be entered as a number followed immediately by the letter `k`, `m`, or *s* to indicate kilobytes, megabytes, or sectors, respectively. If no unit is specified, the default is sectors. The volume size should be less than or equal to the available free space of the disks.
Usage Type:	The desired usage type. The `fsgen` type is the file system generic usage type, which assumes that the volume is being used by a file system. The `gen` type is the generic usage type, which makes no assumptions regarding the data content of the volume. The default is `fsgen`.
Create file system:	Indicates whether a file system is to be created. When you invoke this form from the Volume Operations menu, the default is not to create a file system (No). All fields below this field are only accessible when Yes is specified here.

The following fields only apply if the Create file system: field is set to Yes. Otherwise, these fields are inaccessible.

Field	Description
FS type:	UFS is the only currently supported files system type.
Mount file system:	Indicates whether the file system should be mounted after creation. If the answer is Yes (the default), a mount point must also be specified in the next field. All fields below this field are only accessible when Yes is specified here.
Mount point:	The desired mount point for the new file system. If the specified mount point does not already exist, the Visual Administrator automatically creates it. This field is required if the file system is to be mounted.
Mount automatically:	Indicates whether this file system should be mounted every time the system comes up (by placing an entry in `/etc/fstab`). The default is Yes .

- Striped Volume/FS Create Form

  Basic-Ops → Volume Operations → Create → Striped

  This form creates a concatenated volume and optionally creates a file system on the new volume. The form is divided into two sections, one for volume creation and the other for file system creation. Most of the form fields are already set to the defaults for the creation of a new volume; the file system fields are grayed out because the default is not to add a file system to the volume. The following table describes the fields for this form.

Most fields in this form are required; those that are optional are listed here. All fields in this form are read/write fields.

Field	Description
Volume name:	The name of the volume to be created. The name must be unique within this disk group. The maximum length of this field is 31 characters.
Volume size:	The desired volume size. The size should be entered as a number followed immediately by the letter `k`, `m`, or *s* to indicate kilobytes, megabytes, or sectors, respectively. If no unit is specified, the default is sectors. If the size is not wholly divisible by the stripe width, LSM will adjust the volume size up to the next even multiple in order to create the volume. For a striped volume, the volume size should be calculated as follows: *vol_size = stripe_width * number_of_stripes * n*, where *n* is a number greater than zero. The volume size should be less than or equal to the available free space of the disks.
Usage Type:	The desired usage type. The `fsgen` type is the file system generic usage type, which assumes that the volume is being used by a file system. The `gen` type is the generic usage type, which makes no assumptions regarding the data content of the volume. The default is `fsgen`.
Number of Stripes:	The number of stripes that the volume's plex is to have. This is effectively the number of disks on which the volume is to be created. If some number of disks have already been selected, that number of stripes appears in this field. This number corresponds to the number of disks across which data will be striped. If no number is specified, LSM selects an appropriate number (usually 2).
Stripe width:	The width of the stripes on the plex that this volume will have. The value specified may be optimized for the particular application. However, the default value for this field of 128 sectors is a good stripe width for most systems.
Create file system:	Indicates whether a file system is to be created. When you invoke this form from the Volume Operations menu, the default is not to create a file system (`No`). All fields below this field are only accessible when Yes is specified here.

The following fields only apply if you set the Create file system: field to Yes. Otherwise, these fields are inaccessible.

Field	Description
FS type:	UFS is the only currently supported file system type.
Mount file system:	Indicates whether the file system should be mounted after creation. If the answer is Yes (the default), you must also specify a mount point in the next field. All fields below this field are only accessible when Yes is specified here.
Mount point:	The desired mount point for the new file system. If the specified mount point does not already exist, the Visual Administrator automatically creates it. This field is required if the file system is to be mounted.
Mount automatically:	Indicates whether this file system should be mounted every time the system comes up (by placing an entry in `/etc/fstab`). The default is Yes.

- Volume Resize Form

Warning

File systems and other applications cannot currently resize their data when LSM resizes a volume, therefore shrinking a volume that contains data destroys the data. Therefore, only use this operation when a volume contains no valuable data.

Basic-Ops → Volume Operations → Resize

This form either grows or shrinks a volume using the Logical Storage Manager free space management resources. If new disk space is needed, it will be allocated as necessary; if space becomes unused, it will be added to the free space pool. The following table describes the fields in this form.

Most fields in this form are required; those that are optional are listed here. Fields in this form are read/write fields, unless they are listed as read only.

Field	Description
Selected Volume:	This field displays the name of the volume to be resized. This field is read only and cannot be changed.
Current size:	This field displays the current size of the volume to be resized. This field is read only and cannot be changed.

Field	Description
Option:	The type of resize operation to be performed. This will determine whether the volume is grown or shrunk to a certain size, or grown or shrunk by a given amount. The default is Grow To.
Size/Amount:	Enter either the length to which or the amount by which the volume is to be resized. If Grow To or Shrink To is selected, this field should reflect the final size. If Grow By or Shrink By is selected, this field should reflect the amount by which the size should change. The new volume size should be less than or equal to the available free space of the disks.

- Snapshot Form

  Basic-Ops → Volume Operations → Snapshot

  This form creates a snapshot of the selected volume for backup purposes. The following table describes the fields for this form. Fields in this form are required.

  Fields in this form are read/write fields, unless they are listed as read only.

Field	Description
Selected Volume:	The name of the volume to be used as the snapshot source. This field is read only and cannot be changed.
Snapshot name:	The name of the snapshot volume to be created as a backup. Although a default name appears in this field, a name that more closely resembles that of the selected volume should be used for easier association. The maximum length is 31 characters. The snapshot name must be unique.

`Requirement`: There must be sufficient free space to accommodate the snapshot volume.

C.1.2.2 Advanced-Ops Forms

The following forms are accessed via volume-related selections from the Advanced-Ops menu:

- Volume Create Form

  Advanced-Ops → Volume → Create

  This form creates a volume according to the user's specifications. The following table describes the fields for this form.

  Most fields in this form are required; those that are optional are listed here. Fields in this form are read/write fields, unless listed as read only.

Field	Description
Volume name:	The name of the volume to be created. The name must be unique within this disk group. The maximum length of this field is 31 characters. The name specified for the volume must be unique within this disk group.
Usage Type:	The desired usage type. The `fsgen` type is the file system generic usage type, which assumes that the volume is being used by a file system. The `gen` type is the generic usage type, which makes no assumptions regarding the data content of the volume. The default is `fsgen`. This field is optional.
User:	The name of the user who will be the owner of this volume. This must be a valid user name on the system. The maximum length of this field is 64 characters.
Group:	The name of the group that will own this volume. This must be a valid group name on the system. The maximum length of this field is 64 characters.
Mode:	The permissions mode for the new volume. Only numbers of the correct format are valid in this field. The maximum length of this field is 4 characters.
Length:	The length of the volume. If no unit is specified, the default is sectors. Only positive numbers greater than zero are valid. This field is optional.
Plexes:	This field displays the number of plexes associated with the volume. If no plexes were selected prior to invoking this form, this field displays 0. This field is read only and cannot be changed.
Read Policy:	The read policy that the volume adopts when deciding which plex to write to. These policies are distinguished as follows: Round Robin — All plexes are read equally, in turn. Preferred Plex — A particular plex is specified as the plex to be read whenever possible. The preferred plex will not be read in situations such as when that plex is detached due to I/O failure. Based on plex layouts — All plexes are read equally and in turn, unless a striped plex is present, in which case the striped plex becomes the preferred plex. This option is the default and it typically gives the best read performance.
Preferred Plex:	The name of the preferred plex if the Preferred Plex read policy has been specified. The string in this field must be the name of a valid plex that is associated with this volume. This field is required if Preferred Plex is specified in the Read Policy: field.
Comment:	An appropriate comment for this volume. The maximum length of the comment is 40 characters. This field is optional.

Field	Description
Startup:	This field may contain an arbitrary string that is reserved for the user by usage-type utilities. The intention is that this field be used to store options that apply to the volume, such as for the start volumes operation. This is normally a comma-separated list of flag names and *option=value* pairs. This field is optional.
Logging:	Indicates whether logging is defined and supported on this volume. An undefined log type is included to support old versions of the Logical Storage Manager. The default is Don't Log.
Writeback:	Indicates whether the volume is to write back on read failure. If set to Yes, an attempt will be made to fix a read error from a participating plex. The default is Yes.
Putil0:	Permanent utility field 0. This is reserved for Logical Storage Manager use, but can be changed. The maximum length of all Putil fields is 14 characters. This field is optional.
Putil1:	Permanent utility field 1. This field is reserved, but can be changed. This field is optional.
Putil2:	Permanent utility field 2. This field is reserved, but can be changed. This field is optional.

C.1.3 Volume Properties Form

The following is the properties form that reveals the properties of a particular volume:

- Volume Properties Form

  You can access this form by clicking the MB3 on the desired volume icon. (If the volume icon is undergoing analysis, press Shift-MB3 instead.)

  This form provides detailed information on the attributes of a particular volume. The following table describes the fields in this form.

  The fields in this form are read/write fields, unless listed as read only. Properties of the volume can be changed via this form by altering the current values in the appropriate read/write fields and then clicking on the Apply button.

Field	Description
Volume name:	The name of the volume. This name must be unique within this disk group. The maximum length of this field is 31 characters. This volume name can be changed by entering another name in this field.
Usage Type:	The volume usage type. The `fsgen` type is the file system generic usage type, which assumes that the volume is being used by a file system. The `gen` type is the generic usage type, which makes no assumptions regarding the data content of the volume.
Utility State:	The state that the volume is currently in. This should be either Started, Startable, or Unstartable. This field is read only and cannot be changed.
User:	The name of the user who owns this volume. This must be a valid user name. The maximum length of this field is 64 characters.
Group:	The name of the group that will own this volume. This must be a valid group name. The maximum length of this field is 64 characters.
Mode:	The permissions mode for the volume. Only numbers of the correct format are valid in this field. The maximum length of this field is 4 characters.
Length:	The length of the volume. If no unit is specified, the default is sectors. Only positive numbers greater than zero are valid.
Plexes:	This field displays the number of plexes associated with the volume. If no plexes were selected prior to invoking this form, this field displays 0. This field is read only and cannot be changed.
Read Policy:	The read policy that the volume adopts when deciding which plex to write to. These policies are distinguished as follows: Round Robin — All plexes are read equally, in turn. Preferred Plex — A particular plex is specified as the plex to be read whenever possible. The preferred plex will not be read in situations such as when that plex is detached due to I/O failure. Based on plex layouts — All plexes are read equally and in turn, unless a striped plex is present, in which case the striped plex becomes the preferred plex. This option is the default and it typically gives the best read performance.
Preferred Plex:	The name of the preferred plex if the Preferred Plex read policy has been specified. The string in this field must be the name of a valid plex that is associated with this volume. This field applies only if Preferred Plex is specified in the Read Policy: field.
Comment:	A comment relevant to this volume. The maximum length of the comment is 40 characters.

Field	Description
Startup:	This field may contain an arbitrary string that is reserved for the user by usage-type utilities. The intention is that this field be used to store options that apply to the volume, such as for the start volumes operation. This is normally a comma-separated list of flag names and *option=value* pairs.
Logging:	Indicates whether logging is defined and supported on this volume. An undefined log type is included to support old versions of the Logical Storage Manager.
Writeback:	Indicates whether the volume is to write back on read failure. If set to Yes, an attempt will be made to fix a read error from a participating plex.
Putil0:	Permanent utility field 0. This is reserved for Logical Storage Manager use, but can be changed. The maximum length of all Putil fields is 14 characters.
Putil1:	Permanent utility field 1. This field is reserved, but can be changed.
Putil2:	Permanent utility field 2. This field is reserved, but can be changed.
Tutil0:	Temporary utility field 0. This is reserved for LSM use, but can be changed. The maximum length of all Tutil fields is 14 characters.
Tutil1:	Temporary utility field 1. This field is reserved, but can be changed.
Tutil2:	Temporary utility field 2. This field is reserved, but can be changed.
Kernel State:	The kernel state of this volume. These states are distinguished as follows: Enabled — The volume device can be used. This is the default state. Detached — The volume device cannot be used, but ioctls will still be accepted. Disabled — The volume cannot be used for any operations.
Number of IO Failures:	The number of failed I/O operations on this volume since the last boot. This field cannot be changed.

C.2 Plex Management

The following sections provide information on menus and forms relating to plex management.

C.2.1 Plex Menus

Advanced-Ops → Plex

The Advanced-Ops menu provides access to the following plex-related menus:

- Create
- Remove Plexes
- Associate Plexes
- Disassociate Plexes
- Attach Plexes
- Detach Plexes
- Help

The Help selection accesses a Help window that displays information relevant to the plex operations.

The plex Advanced-Ops menus are described in the following list:

- Create

  Advanced-Ops → Plex → Create

  This operation creates a plex. You can select one or more subdisks to be associated with the new plex after creation.

  `Form`: Plex Create Form

- Remove Plexes

  Advanced-Ops → Plex → Remove plexes

  This operation removes the selected plexes. This is a permanent operation and cannot be undone. Any subdisks associated with the plex will be disassociated and left behind.

  `Requirements`:

  – At least one plex icon must be selected.

  – If the selected plex is associated with a volume, it must be disassociated before it can be removed.

- Associate Plexes

  Advanced-Ops → Plex → Associate Plexes

  This operation associates one or more selected plexes with the selected volume. If the volume is started, LSM begins to bring the plex up to date by copying all necessary data to the plex. This may take time.

  `Requirements`:

  – A volume icon and at least one plex icon must be selected.

  – Only nonassociated plexes can be associated.

- Disassociate Plexes

  Advanced-Ops → Plex → Disassociate Plexes

This operation disassociates one or more selected plexes from their parent volumes. This operation will fail if the plex cannot be disassociated. For example, the last plex in a started volume cannot be disassociated.

`Requirements:`

– At least one plex icon must be selected.

– Only associated plexes can be disassociated.

– Before the last plex in a volume can be disassociated, that volume must be stopped.

• Attach Plexes

Advanced-Ops → Plex → Attach Plexes

This operation attaches one or more selected plexes to their parent volumes. A plex must be detached but still associated with an enabled volume in order to be attached; the plex is actually being reattached with its parent volume.

`Requirements:`

– At least one plex icon must be selected.

– A plex must be detached before it can be attached.

– Only a plex associated with an enabled volume can be attached.

• Detach Plexes

Advanced-Ops → Plex → Detach Plexes

This operation detaches one or more selected plexes from their parent volumes. A detached plex is inaccessible for reads and writes, but is still associated with the volume.

`Requirements:`

– At least one plex icon must be selected.

– Only associated plexes can be detached.

– This operation is not permitted when the specified plex is the last valid plex on the volume.

C.2.2 Plex Forms

Some plex operations result in the appearance of forms, which must be completed in order for that operation to proceed. Most forms provide a Help button, which contains information relevant to the fields and other aspects of that particular form.

The following forms are accessed via plex-related selections from the Advanced-Ops menu:

- Plex Create Form

Advanced-Ops → Plex → Create

The following table describes the fields in this form. Most fields in this form are required; those that are optional are listed here. Fields in this form are read/write fields, unless listed as read only.

Field	Description
Plex name:	The name of the plex to be created. The name must be unique within this disk group. The maximum length of this field is 31 characters.
Plex state:	The plex utility state. This is reserved for use by usage types. This field is optional.
Volume:	The name of the volume that this plex should be associated with. The name must be a valid volume name in this disk group. The maximum length of this field is 31 characters. This field is optional.
Layout:	The desired layout for the plex. A concatenated plex is a plex with associated subdisks that are both sequentially and contiguously arranged. A striped plex is a plex that distributes data evenly across each of its associated subdisks. The default is Concatenated.
Stripe width:	The width of the stripes on the plex. The stripe width must be a number greater than 0. If no units are specified, sectors are assumed. The maximum length of this field is 14 characters. If a striped plex layout is specified, this field is required. This field must be blank if a concatenated plex layout is specified.
Subdisks:	The number of subdisks associated with the plex. This field is read only and cannot be changed.
Comment:	An appropriate comment for the plex. The maximum length of the comment is 40 characters. This field is optional.
Errors:	Indicates whether the plex should participate in LSM error policies. The default is Participate.
Putil0:	Permanent utility field 0. This is reserved for LSM use, but may be changed. The maximum length of all Putil fields is 14 characters. This field is optional.

Field	Description
Putil1:	Permanent utility field 1. This field is reserved, but can be changed. This field is optional.
Putil2:	Permanent utility field 2. This field is reserved, but can be changed. This field is optional.

C.2.3 Plex Properties Forms

The following list describes the properties form that reveals the properties of a particular plex:

- Plex Properties Form

  To access the plex properties form, click the MB3 mouse button on desired plex icon.

  This form provides detailed information on the attributes of a particular plex. The following table describes the fields in this form.

  The fields in this form are read/write fields, unless listed as read only. Properties of the plex can be changed via this form by altering the current values in the appropriate read/write fields and then clicking on the Apply button.

Field	Description
Plex name:	The name of the plex. The name must be unique within this disk group. The maximum length of this field is 31 characters. The plex name can be changed by entering another name in this field.
Plex state:	The plex utility state. This is reserved for use by usage types. This field is read only and cannot be changed.
Volume:	The name of the volume that this plex should be associated with. This field is read only and cannot be changed.
Layout:	The layout of the plex: concatenated or striped. A concatenated plex is a plex with associated subdisks that are both sequentially and contiguously arranged. A striped plex is a plex that distributes data evenly across each of its associated subdisks. This field is read only and cannot be changed.
Stripe width:	The width of the stripes on the plex. If Striped plex layout has been specified, this field indicates the stripe width. This field should be blank if Concatenated plex layout has been specified. This field is read only and cannot be changed.
Subdisks:	The number of subdisks associated with the plex. This field is read only and cannot be changed.

Field	Description
Log Subdisk:	This field shows the name of the subdisk that is being used for logging on this plex. If there is no associated Dirty Region Logging subdisk (no logging in effect), this field is blank. This field is read only and cannot be changed.
Comment:	An appropriate comment for the plex. The maximum length of the comment is 40 characters.
Errors:	Indicates whether the plex participates in LSM error policies. This field is read only and cannot be changed.
Putil0:	Permanent utility field 0. This is reserved for use, but can be changed. The maximum length of all Putil fields is 14 characters.
Putil1:	Permanent utility field 1. This field is reserved, but can be changed.
Putil2:	Permanent utility field 2. This field is reserved, but can be changed.
Tutil0:	Temporary utility field 0. This is reserved for LSM use, but can be changed. The maximum length of all Tutil fields is 14 characters.
Tutil1:	Temporary utility field 1. This field is reserved, but can be changed.
Tutil2:	Temporary utility field 2. This field is reserved, but can be changed.
Kernel State:	The accessibility of the plex. This field is read only and cannot be changed.
Length:	The length of the plex. This field is read only and cannot be changed.
Number of I/O failures:	The number of failed I/O operations on this plex since the last boot. This field is read only and cannot be changed.

C.3 Subdisk Management

The following sections provide information on menus and forms relating to subdisk management.

C.3.1 Subdisk Menus

You access the subdisk Advanced-Ops menu as shown here:

Advanced-Ops → Subdisk

This menu provides access to the following subdisk operations:

- Create

- Remove Subdisks
- Associate Subdisks
- Associate as Log Sd
- Disassociate Subdisks
- Join Subdisks
- Split the Subdisk
- Help

The Help selection accesses a Help window that displays information relevant to the subdisk operations.

The following list describes how to access the subdisk menus:

- Create

  Advanced-Ops → Subdisk → Create

  This operation creates a subdisk on the selected LSM disk. An LSM disk must be selected.

  `Form`: Subdisk Create Form (described in Section C.3.2).

- Remove Subdisks

  Advanced-Ops → Subdisk → Remove Subdisks

  This operation removes the selected subdisks. This is a permanent operation and cannot be undone.

  `Requirements`:

  – At least one subdisk icon must be selected.

  – If the selected subdisk is associated with a plex, it must be disassociated before it can be removed. Only free subdisks can be removed.

  Associate Subdisks

  Advanced-Ops → Subdisk → Associate Subdisks

  This operation associates one or more subdisks with the selected plex.

  `Requirements`:

  – A plex icon and at least one subdisk icon must be selected.

  – Only nonassociated (free) subdisks can be associated.

- Associate as Log Subdisk

  Advanced-Ops → Subdisk → Associate as Log Sd

  This operation associates the selected subdisk as a log subdisk with the selected plex. The resulting log subdisk icon has double borders to distinguish it from normal subdisks.

Requirements:

- A plex icon and a subdisk icon must be selected.
- Only nonassociated (free) subdisks can be associated.
- The selected plex cannot already have a log subdisk.
- Subdisks must be 2 or more sectors to enable logging in noncluster environments and 65 or more sectors for TruCluster environments.

• Disassociate Subdisks

Advanced-Ops → Subdisk → Disassociate Subdisks

This operation disassociates one or more selected subdisks from their parent plexes. Both log subdisks and normal subdisks can be disassociated.

Requirements:

- At least one subdisk icon must be selected.
- Only associated subdisks can be disassociated.
- The last subdisk associated with a plex that is currently associated with a volume cannot be disassociated. The plex must be disassociated from its volume first.

• Join Subdisks

Advanced-Ops → Subdisk → Join Subdisks

This operation joins the selected subdisks together to create a single subdisk. The resulting subdisk has the offset and name of the first subdisk (as arranged on the disk) and its length is the sum of the subdisk lengths.

Requirements:

- At least two subdisk icons must be selected.
- The subdisks must be contiguous on the disk.
- If the subdisks are associated, they must all be associated with the same plex and be contiguous on that plex.
- Logging subdisks and subdisks associated with striped plexes cannot be joined.

• Split a Subdisk

Advanced-Ops → Subdisk → Split the Subdisk

This operation splits the selected subdisk into either two or many parts. The resulting subdisks will occupy the same region on the disk that the previous subdisk occupied. If the subdisk is associated with a plex, the resulting subdisks will also be associated with that plex.

From the Split the subdisk menu, a submenu allows the user to indicate whether the subdisk is to be split into two or several parts:

Into 2 Subdisks	Split the selected subdisk into 2 subdisks.
Into More Than 2 Subdisks	Split the selected subdisk into several subdisks.

`Requirements:`

– Only one subdisk icon can be selected.

– Logging subdisks cannot be split.

`Forms:` Subdisk Split Into Two

Subdisk Split Into Many (described in Section C.3.2).

C.3.2 Subdisk Forms

Some subdisk operations result in the appearance of forms, which must be completed in order for that operation to proceed. Most forms provide a Help button, which contains information relevant to the fields and other aspects of that particular form.

The following forms are accessed via subdisk-related selections from the Advanced-Ops menu:

- Subdisk Create Form

  Advanced-Ops → Subdisk → Create

  This form creates a subdisk according to the user's specifications. The following table describes the fields in this form.

  Most fields in this form are required; those that are optional are listed here. Fields in this form are read/write fields, unless listed as read only.

Field	Description
Disk name:	The name of the LSM disk on which the subdisk is to be created. This field is read only and cannot be changed.
Subdisk name:	The name of the subdisk to be created. The name must be unique within this disk group. The maximum length of this field is 31 characters.
Disk offset:	The length into the disk where this subdisk should be located. If no units are specified, sectors are assumed. This offset should not place this subdisk within the bounds of another subdisk on the disk or past the end of the disk. Only valid positive numbers are allowed in this field.

Field	Description
Subdisk length:	The length of the subdisk to be created. If no units are specified, sectors are assumed. The length should not place this subdisk within the bounds of another subdisk on the disk or past the end of the disk. Only valid positive numbers are allowed in this field.
Plex name:	The name of the plex with which the subdisk is to be associated. This must be a valid plex that already exists in this disk group. The maximum length of this field is 31 characters. This field is optional.
Plex offset:	The offset of this subdisk into its associated plex. Only valid positive numbers are allowed in this field. This field is required only if a plex has been specified for association. If the subdisk is not to be associated with a plex, this field must be left blank.
Comment:	An appropriate comment for the subdisk. The maximum length of the comment is 40 characters. This field is optional.
Putil0:	Permanent utility field 0. This is reserved for Logical Storage Manager use, but may be changed. The maximum length of all Putil fields is 14 characters. This field is optional.
Putil1:	Permanent utility field 1. This field is reserved, but may be changed. The maximum length of this field is 14 characters. This field is optional.
Putil2:	Permanent utility field 2. This field is reserved, but may be changed. The maximum length of this field is 14 characters. This field is optional.

• Subdisk Split Into Two

Advanced-Ops → Subdisk → Split the Subdisk –> Into 2 Subdisks

This form is used to split the selected subdisk into exactly two subdisks. The first subdisk retains the name and size of the original one; the second subdisk adopts the name and size specified in this form. The following table describes the fields in this form.

Most fields in this form are required; those that are optional are listed here. Fields in this form are read/write fields, unless listed as read only.

Field	Description
Present size:	The size of the subdisk to be split. This field is read only and cannot be changed.
Name of new subdisk:	The name of the subdisk to be created from the original one. This must be a valid name and must be unique in this disk group.
Size of new subdisk:	The size of the subdisk to be created from the original one. This must be a valid number, greater than zero. The new subdisk size must be at least one sector less than the present subdisk size.

- Subdisk Split Into Many

  Advanced-Ops → Subdisk → Split the Subdisk –> Into More Than 2 Subdisks

  This form is used to split the selected subdisk into several subdisks of equal sizes. The first subdisk retains the name and size of the original one; the additional subdisks are automatically named by LSM. The following table describes the fields in this form.

  Most fields in this form are required; those that are optional are listed here. Fields in this form are read/write fields, unless listed as read only.

Field	Description
Original subdisk:	The name of the selected subdisk. This field is read only and cannot be changed.
Present size:	The size of the subdisk to be split. The original subdisk must contain enough sectors to accommodate the desired total number of subdisks for the split. This field is read only and cannot be changed.
Number of new subdisks:	The total number of subdisks to be created by the split. There must be a sufficient number of sectors in the original subdisk to accommodate this number. This number should be at least 2.

`Requirements:` The number of subdisks is limited by the amount of space left in the configuration database.

C.3.3 Subdisk Properties Forms

The following is the properties form that reveals the properties of a particular subdisk:

- Subdisk Properties Form

  To access the Subdisk Properties form, click MB3 on the desired subdisk icon. If the subdisk is undergoing analysis, press Shift-MB3 instead. This form provides detailed information on the attributes of a particular subdisk. The following table describes the fields in this form.

The fields in this form are read/write fields, unless listed as read only. Properties of the subdisk can be changed via this form by altering the current values in the appropriate read/write fields and then clicking on the Apply button.

Field	Description
Disk name:	The name of the disk where the subdisk resides. This field is read only and cannot be changed.
Subdisk name:	The name of the subdisk. The name must be unique within this disk group. The maximum length of this field is 31 characters. The subdisk name can be changed by entering another name in this field.
Disk offset:	The length into the disk where this subdisk is located, in sectors. This field is read only and cannot be changed.
Subdisk length:	The length of the subdisk. If no units are specified the number is assumed to be in sectors. This offset should not place this subdisk within the bounds of another subdisk on the disk or past the end of the disk. Only valid positive numbers are allowed in this field.
Plex name:	The name of the plex with which the subdisk is associated. This field is read only and cannot be changed.
Plex offset:	The offset of this subdisk into its associated plex. If the subdisk is not associated, this field contains a zero. This field is read only and cannot be changed.
Comment:	An appropriate comment for the subdisk. The maximum length of the comment is 40 characters.
Log Subdisk:	Indicates whether this subdisk is a Dirty Region Logging subdisk. This field is read only and cannot be changed.
Putil0:	Permanent utility field 0. This is reserved for LSM use, but may be changed. The maximum length of all Putil fields is 14 characters.
Putil1:	Permanent utility field 1. This field is reserved, but may be changed. The maximum length of this field is 14 characters.
Putil2:	Permanent utility field 2. This field is reserved, but may be changed. The maximum length of this field is 14 characters.
Tutil0:	Temporary utility field 0. This is reserved for LSM use, but may be changed. The maximum length of all Tutil fields is 14 characters.
Tutil1:	Temporary utility field 1. This field is reserved, but may be changed. The maximum length of this field is 14 characters.

Field	Description
Tutil2:	Temporary utility field 2. This field is reserved, but may be changed. The maximum length of this field is 14 characters.
Number of IO failures:	The number of failed I/O operations on this subdisk since the last boot. This field is read only and cannot be changed.

C.4 Disk Management

The following sections provide information on menus and forms relating to disk management.

C.4.1 Disk Menus

Both the Basic-Ops and Advanced-Ops menus provide access to disk-related operations. Most menus provide a Help selection, which contains information relevant to the items and operations listed in that particular menu.

C.4.1.1 Basic-Ops Menu

You access the Basic-Ops menu by selecting:

Basic-Ops → Disk Operations

This menu provides access to disk operations involving general disk maintenance. These operations use the automated approach to disk management.

The Disk Operations menu provides the following selections:

- Add Disks
- Evacuate Subdisks
- Replace Disks
- Remove Disks
- Help

The Help selection accesses a Help window, which displays information relevant to the basic disk operations.

The following list describes the menu selections you can access via the Basic-Ops menu:

- Add Disks

  Basic-Ops → Disk Operations → Add Disks

This operation adds a disk to the Logical Storage Manager, placing it under LSM control. This involves initializing, analyzing, and partitioning the raw disk; initializing the disk for LSM use; and adding the disk to a disk group (if requested).

`Form`: Add Disks Form (described in the Disk Forms section).

- Evacuate Disks

  Basic-Ops → Disk Operations → Evacuate Disks

  This operation moves all subdisks from the selected disk to another disk in the same disk group.

  `Requirements`: The disk from which subdisks are to be evacuated must be selected. Both disks must belong to the same disk group.

  `Forms`: Evacuate Subdisks Form (described in the Disk Forms section).

- Replace Disks

  Basic-Ops → Disk Operations → Replace Disks

  This operation replaces a disk. This is normally done when a failed disk needs to be replaced with a new one. This involves initializing and partitioning the raw disk; initializing the disk for LSM use; and replacing the old disk and associated disk media records with the new disk and its information.

  `Requirements`: A disk icon representing a failed disk must be selected.

  `Forms`: Replace Disks Form (described in the Disk Forms section).

- Remove Disks

  Basic-Ops → Disk Operations → Remove Disks

  This operation removes a disk from a disk group and then removes the disk from LSM control.

  `Requirements`: A disk icon must be selected.

C.4.1.2 Advanced-Ops Menu

You access the Advanced-Ops menu selections by selecting:

Advanced-Ops → Disk

This menu provides access to assorted disk operations using the manual approach to disk management.

The Disk menu provides the following selections:

- Initialize
- Define

- Remove
- Online
- Offline
- Help

The Help selection accesses a Help window, which displays information relevant to the advanced disk operations.

The following list describes the menu selections you can access via the Advanced-Ops menu:

- Initialize

  Advanced-Ops → Disk → Initialize

  This operation identifies a disk to LSM and initializes the disk for LSM use. This involves installing a disk header and writing an empty configuration on the disk. A disk access record is created for the disk, unless such a record already exists.

  `Requirement`: The disk should not already be initialized.

  `Form`: Disk Init Form (described in the Disk Forms section).

- Define

  Advanced-Ops → Disk → Define

  This operation defines a disk access record that enables LSM to scan the disk. This makes the disk accessible, but does not initialize the disk.

  `Form`: Define Disk Form (described in the Disk Forms section).

- Remove

  Advanced-Ops → Disk → Remove

  This operation removes the LSM disk associated with the selected partitions from LSM control by removing the associated disk access records. If all partitions on a given disk are selected for removal at once, the disk is effectively removed from LSM control.

  `Requirements`:

  – At least one partition icon corresponding to a LSM disk must be selected.

  – The LSM disks corresponding to the selected partitions cannot belong to a disk group at the time of removal.

- Online

  Advanced-Ops → Disk → Online

This operation places the disk access record on a specified partition in an online state. During searches for disk IDs or members of a disk group, online disks are checked.

`Form`: Disk Online Form (described in the Disk Forms section).

- Offline

  Advanced-Ops → Disk → Offline

  This operation places the disk access record on the selected partitions in an offline state. During searches for disk IDs or members of a disk group, offline disks are ignored.

  `Requirements`:

  – At least one partition icon must be selected.

  – The disks corresponding to the selected partitions must be initialized.

  – The selected partition icon cannot be in use (shaded and associated with a LSM disk).

C.4.2 Disk Forms

Some disk operations result in the appearance of forms. You must complete these forms in order for that operation to proceed. Most forms provide a Help button that provides access to information relevant to the fields and other aspects of that form.

C.4.3 Basic-Ops Forms

The following forms are accessed via disk-related selections from the Basic-Ops menu.

- Add Disks Form

  Basic-Ops → Disk Operations → Add Disks

  This form is used to place a disk under Logical Storage Manager control. The following table describes the fields in this form.

  Most fields in this form are required; those that are optional are listed here. Fields in this form are read/write fields, unless listed as read only.

Field	Description
New disk name:	The name of the new physical disk in the form `dsk`*nn*, for example, `dsk10`. The name must be unique within this disk group. You can also place specific partitions on a disk under LSM control. For example, `dsk3g` would put the `g` partition on `dsk3` under LSM control.
Disk group:	The name of the disk group to which this disk is to be added. The named disk group must exist. If no name is provided, it will not be added to a disk group. This field is optional.

- Replace Disks Form

  Basic-Ops → Disk Operations → Replace Disks

  This form is used to replace an existing LSM disk that has failed with another one. The following table describes the fields in this form.

  Most fields in this form are required; those that are optional are listed here. Fields in this form are read/write fields, unless listed as read only.

Field	Description
Old LSM disk name:	The name of the failed (collapsed or disconnected) LSM disk in this disk group. This field is read only and cannot be changed.
New physical disk name:	The name of the new physical disk that is to replace the existing one. The name should be in the form `dsk`*nn*, for example, `dsk10`. The new name must be unique in this disk group.

- Evacuate Subdisks Form

  Basic-Ops → Disk Operations → Evacuate Subdisks

  This form is used to transfer subdisks from one LSM disk to another. The following table describes the fields in this form.

  Most fields in this form are required; those that are optional are listed here. Fields in this form are read/write fields, unless listed as read only.

Field	Description
Disk group name:	The name of the disk group to which both disks belong. Both disks must share the same disk group.
Evacuate From:	The name of the LSM disk from which the subdisks are to be evacuated.
To:	The name of the LSM disk to which the subdisks are to be moved. This field is optional. However, if no target disk is specified, the subdisks are evacuated to one or more random disks (depending on disk space availability).

C.4.3.1 Advanced-Ops Forms

The following forms are accessed via disk-related selections from the Advanced-Ops menu:

- Disk Init Form

Advanced-Ops → Disk → Initialize

This form is used to initialize a disk for LSM use. The following table describes the fields in this form.

Most fields in this form are required; those that are optional are listed here. Fields in this form are read/write fields, unless listed as read only.

Field	Description
Public Device	The pathname of the device node that represents a partition available for use. This name must be a valid entry in `/dev/disk`. A name in the form `dsk`*nn* is used to assign the full disk under LSM control. The disk `dsk`*nn* would be added as a sliced LSM disk. Before a sliced disk can be defined, change the disk label to have LSM disk label tags. A name in the form `dsk`*nnp* is used to assign partition *p* on disk `dsk`*nn* under LSM control. The disk partition `dsk`*nnp* is added as a simple LSM disk.
Device Type	The desired disk type. The simple type (default) assumes that the public and private regions are stored on the same disk partition, with the public region following the private region. The sliced type assumes that the public and private regions are stored on different disk partitions. Before initializing the disk, change the disk label to have LSM disk label tags. The `nopriv` type has no private region and log and configuration copies cannot be written to the disk.
Public length (0 for whole device)	The length of the public section of the disk. If zero is provided as the length, the Logical Storage Manager computes a default value from available partition table information. This length must be valid and cannot exceed the length of the disk.

Field	Description
Private Length:	The length of the private region of the disk. When one is not specified, LSM chooses a default value. This length must be valid and cannot exceed the length of the disk. For a sliced disk, the length cannot exceed the size of the partition chosen for the private region. This field is optional.
Number of config copies:	The number of configuration copies to be stored in the private section of this disk. The default value is 1 copy.
Comment:	A comment appropriate for the LSM disk. The maximum length of the comment is 40 characters. This field is optional.

- Define Disk Form

  Advanced-Ops → Disk → Define

  This form is used to define a disk. The following table describes the fields in this form.

  Most fields in this form are required; those that are optional are listed here. Fields in this form are read/write fields, unless listed as read only.

Field	Description
Public Device	The pathname of the device node that represents a partition available for use. This name must be a valid entry in `/dev/disk`. A name in the form `dsk`*nn* is used to assign the full disk under LSM control. The disk `dsk`*nn* would be added as a sliced LSM disk. A name in the form `dsk`*nnp* is used to assign partition *p* on disk `dsk`*nn* under LSM control. The disk partition `dsk`*nnp* would be added as a simple LSM disk.
Device Type	The desired disk type. The simple type (default) assumes that the public and private regions are stored on the same disk partition, with the public region following the private region. The sliced type assumes that the public and private regions are stored on different disk partitions. The nopriv type has no private region and log and configuration copies cannot be written to the disk.
Public Length (0 for whole disk):	The length of the public section of the disk. If zero is provided as the length, LSM computes a default value from available partition table information. This length must be valid and cannot exceed the length of the disk.
Offline:	Indicates whether to initially place the disk in the offline state. The default is No.
Comment:	A comment appropriate for this Logical Storage Manager disk. The maximum length of the comment is 40 characters. This field is optional.

- Disk Online Form

Advanced-Ops → Disk → Online

This form is used to place a disk on line. The following table describes the fields in this form.

Field	Description
Device name:	The disk access name of the disk to be placed onlined. This must be a valid disk access name. This field is required.

- Free Space Form

  To access the free space form, click MB3 on a gap between subdisk icons in a LSM disk icon.

  This form provides information about a specific region of an LSM disk that contains free space.

  Free space results when subdisks are removed for some reason, making the space that they occupied available for use. Free space is visually represented as a gap or hole between subdisks that reside on a LSM disk icon. The following table describes the fields in the form. All fields in this form are read only and cannot be changed.

Field	Description
Device:	The name of the LSM disk where this free space resides.
Hole offset:	The offset into the LSM disk where this free space extent begins.
Hole size:	The size of this free space extent. The units used are specified by the user under the Options pull down menu.

C.4.4 Disk Properties Forms

Properties forms exist for LSM disks, physical disks, and partitions. The following list describes these forms:

- LSM Disk Properties Form

  To access the LSM disk properties form, click MB3 on desired LSM disk icon. (If the LSM disk icon is undergoing analysis, press Shift-MB3 instead.)

  This form provides detailed information on the attributes of a particular LSM disk that is under LSM control. The information displayed in this form actually corresponds to the disk media record associated with a disk. The following table describes the fields in this form.

  The fields in this form are read/write fields, unless listed as read only. Properties of the disk can be changed via this form by altering the current values in the appropriate read/write fields and then clicking on the Apply button.

Field	Description
LSM disk name:	The name of the LSM disk.
Disk Access:	The name of the disk access record that corresponds to this disk media record. This field is read only and cannot be changed.
Disk Type:	The type with which this disk media record was created. This field is read only and cannot be changed.
Public Region:	The name of the public region of this disk. This field is read only and cannot be changed.
Private Region:	The name of the private region of this disk. If there is no private region then this field will be blank. This field is read only and cannot be changed.
Public Region Offset:	The offset, in sectors, of the public region on the disk. This field is read only and cannot be changed.
Private Region Offset:	The offset, in sectors, of the private region on the disk. If there is no public region, then this field will display zero. This field is read only and cannot be changed.
Public Region Length:	The length, in sectors, of the public region on the disk. This field is read only and cannot be changed.
Private Region Length:	The length, in sectors, of the private region on the disk. If there is no private region, this field will display zero. This field is read only and cannot be changed.
Disk Attributes:	The attributes of this LSM disk. This field is read only and cannot be changed.
Comment:	The user-specified comment for this LSM disk. The maximum length of the comment is 40 characters.
Putil0:	Permanent utility field 0. This is reserved for LSM use, but may be changed. The maximum length of all Putil fields is 14 characters.
Putil1:	Permanent utility field 1. This field is reserved, but may be changed.
Putil2:	Permanent utility field 2. This field is reserved, but may be changed.
Tutil0:	Temporary utility field 0. This field is reserved, but may be changed. The maximum length of all Tutil fields is 14 characters.
Tutil1:	Temporary utility field 1. This field is reserved, but may be changed. The maximum length of this field is 14 characters.

Field	Description
Tutil2:	Temporary utility field 2. This field is reserved, but may be changed. The maximum length of this field is 14 characters.
Maximum Free Space:	The maximum amount of free space available on this LSM disk. This does not take disk extents into account. This number assumes every free sector on the LSM disk is usable. This field is read only and cannot be changed.

- Physical Disk Properties Form

  To access the physical disk properties form, click MB3 on the desired physical disk icon.

  This form provides detailed information on the attributes of a particular physical disk. The following table describes the fields in this form.

  All fields in this form are read only and cannot be changed.

Field	Description
Device:	The raw device node for this physical disk.
Device Type:	A brief description the device type. Possible device types include SCSI hard drive and Floppy.
Cylinders:	The number of cylinders on this disk.
Tracks:	The number of tracks per cylinder.
Sectors:	The number of sectors per track.
Sector Size:	The size, in bytes, of each sector on this disk.
Total Size:	The total size of the disk, in sectors.

- Partition Properties Form

  To access the partition properties form, click MB3 on the desired partition icon.

  This form provides detailed information on the attributes of a particular partition. The following table describes the fields in this form.

  All fields in this form are read only and cannot be changed.

Field	Description
Device:	The device node that the LSM Visual Administrator uses to communicate with this disk.
Start Sector:	The sector on the physical disk where this partition begins.
Size:	The length of this partition.

Field	Description
Type:	The identification tag associated with this partition.
Disk Media:	The disk media record that corresponds to this partition. If this field is empty, the partition has not been initialized with a disk media record.

C.5 Disk Group Management

The following sections provide information on menus and forms relating to disk group management.

Note

With the Visual Administrator, partition icons represent partitions described by disk access records.

C.5.1 Disk Group Menus

You access disk group operations via the Advanced-Ops menu, as shown here:

Advanced-Ops → Disk Group

The Advanced-Ops menu provides access to the following disk-related menus.

- Initialize
- Import Disk Groups
- Deport Disk Groups
- Add Disk
- Remove Disks
- Disconnect Disks
- Reconnect Disks
- Help

The Help selection accesses a Help window that displays information relevant to the disk group operations.

The following list describes the disk group menu options:

- Initialize

  Advanced-Ops → Disk Group → Initialize

This operation defines a new disk group with a name you specify. The new disk group contains one or more LSM disks corresponding to the partitions you select.

`Requirements`: At least one partition icon must be selected.

`Form`: Initialize Disk Group Form (described in the Disk Group Forms section).

- Import Disk Group

  Advanced-Ops → Disk Group → Import Disk Group

  This operation imports a disk group to make that disk group available on the local machine. If the name of a deported disk group is known, this operation can be used to make that disk group accessible again.

  `Form`: Import Disk Group Form (described in the Disk Group Forms section).

- Deport Disk Group

  Advanced-Ops → Disk Group → Deport Disk Group

  This operation disables access to a disk group. A deported disk group is no longer accessible and its view window disappears. Once deported, a disk group can be reimported.

  `Requirements`: A disk group cannot be deported if any volumes in that disk group are currently open.

  `Form`: Deport Disk Group Form (described in the Disk Group Forms section).

- Add Disk

  Advanced-Ops → Disk Group → Add Disk

  This operation adds an LSM disk corresponding to the selected partition icon to a disk group. This involves creating a disk media record for the disk to be added. Partitions representing disks that already belong to disk groups cannot be added to disk groups.

  `Requirements`:

  – One partition icon must be selected.

  – The selected partition cannot already belong to a disk group.

  – Only one disk can be added to a disk group at a time.

  `Form`: Add Disk Form (described in the Disk Group Forms section).

- Remove Disks

  Advanced-Ops → Disk Group → Remove Disks

This operation removes the selected LSM disks from a disk group. Disks are removed from the disk group in which they reside. Any subdisks that exist on the selected disks must be removed before the disk can be removed.

`Requirements:`

- At least one LSM disk icon must be selected.
- Only disks associated with the specified disk group can be removed.
- Disks containing any subdisks cannot be removed.
- Only disks in the same disk group can be selected for removal in a single operation.
- The last disk in a disk group cannot be removed. The disk group itself must be deported in order for its last disk to be removed.

• Disconnect Disks

Advanced-Ops → Disk Group → Disconnect Disks

This operation disables the selected LSM disk, making it unavailable for use within its disk group. This involves disassociating the disk media record from its disk access record.

`Requirements:`

- At least one LSM disk icon must be selected.
- The LSM disk icons must contain a disk media record at the time of selection.

• Reconnect Disks

Advanced-Ops → Disk Group → Reconnect Disks

This operation enables a LSM disk that has previously been disconnected. This involves connecting the selected LSM disk’s disk media record with the selected disk access record. Although the LSM disk must be disconnected, it does not necessarily have to be reconnected to its former partition (disk access record).

`Requirements:`

- One LSM disk icon and one partition icon must be selected.
- Neither the LSM disk icon nor the partition icon can already be connected.

C.5.2 Disk Group Forms

Some disk group operations result in the appearance of forms, which must be completed in order for that operation to proceed. Most forms provide a

Help button, which contains information relevant to the field and other aspects of that particular form.

The following forms are accessed via disk group-related selections from the Advanced-Ops menu:

- Initialize Disk Group Form

  Advanced-Ops → Disk Group → Initialize

  This form is used to define a new disk group consisting of selected disks.

  The following table describes the field in this form.

Field	Description
Disk group:	The name of the new disk group. This must be a valid and unique name. This field is required. This is a read/write field.

- Import Disk Group Form

  Advanced-Ops → Disk Group → Import Disk Group

  This form is used to make the specified disk group available to the system. The following table describes the field in this form.

Field	Description
Disk group:	The name of the disk group to be imported and made available to the system. This must be a valid and unique disk group name. This field is required. This is a read/write field.

- Deport Disk Group Form

  Advanced-Ops → Disk Group → Deport Disk Group

  This form is used to make the specified disk group inaccessible to the system. The following table describes the fields in this form.

Field	Description
Disk group:	The name of the disk group to be deported and made inaccessible to the system. This must be a valid disk group.

  `Requirements:` The root disk group (rootdg) cannot be deported.

- Add Disk Form

  Advanced-Ops → Disk Group → Add Disk

  This form is used to add an LSM disk to a disk group. The following table describes the fields in this form.

  Most fields in this form are required; those that are optional are listed here. Fields in this form are read/write fields, unless listed as read only.

Field	Description
Disk group:	The name of the disk group to which the LSM disk is to be added. This must be a valid disk group. This field is required.
Disk media name:	The name of the LSM disk to be created. The disk media name must be unique. By default, a unique name is generated. If this field is left blank, then the disk access name is used.

C.6 Projection Analysis

The following sections provide information on menus and forms relating to projection and analysis. In addition, tables are provided to summarize various aspects of projection and analysis behavior. You can access these operations as follows:

- Projection

  Projection operations are accessed via the Projection menu. This menu is located in view windows such as View of the `rootdg` disk group. The Projection menu starts or stops projection, and highlights any free subdisk icons.

  Projection can also be started or stopped by pressing Shift-MB2 with the pointer positioned on the desired icon.

- Analysis

  Analysis operations are accessed via the Analyze menu. This menu is located in view windows such as View of the `rootdg` disk group. The Analyze menu can be used to starts or stops analysis and sets analysis-related preferences.

C.6.1 Projection

Icon projection provides the user with visual information about the relationships between icons. When projection is started for an icon, all other icons (representing LSM objects) associated with that particular one are highlighted, no matter which views they occupy. Icons can be placed under projection either individually or in multiples. Projection highlighting can accumulate on a given icon when that icon is is undergoing projection from more than one source.

C.6.2 Projection Menus

The following list describes the menus, submenus, and menu selections you can access via the Projection menu:

- Icon Projection

Projection → Icon Projection

This menu provides access to projection options used to start or stop projection for icons.

- Start

  Projection → Icon Projection → Start

  This option starts projection for the selected icons. When projection is started, all icons related to the selected icons are highlighted. Highlighting occurs for related icons in any view windows. If the selected icon has no associated objects, the Visual Administrator issues a warning to this effect.

  `Requirements:`

  – At least one icon must be selected.

  – Physical disk and partition icons cannot be selected for projection.

  – The selected icons must be associated with at least one other icon in order for projection to take effect.

- Stop

  Projection → Icon Projection → Stop

  This options stops projection for the selected icons. When projection is stopped, all icons related to the selected icons lose their projection highlighting.

  `Requirement:` At least one icon must be selected. If the selected icon is not undergoing projection, the Visual Administrator ignores the stop request.

- Stop All

  Projection → Icon Projection → Stop All

  This options stops projection for all icons that are currently undergoing selection.

- Show Free Subdisks

  Projection → Show Free Subdisks

  This menu selection determines whether free subdisks should be highlighted or not. When Show Free Subdisks is turned on, the Visual Administrator highlights all unassociated subdisks (representing unallocated disk space). Once turned on, any future free subdisks are automatically highlighted. Free subdisk icons can be used by designating them to objects, but the LSM Visual Administrator interface cannot automatically use free subdisks as free space. Free subdisk projection is either started or stopped across all Visual

Administrator views. The start or stop preference is also retained for a particular user in future sessions.

From the Show Free Subdisks menu, a submenu allows you to indicate whether or not to highlight free subdisks:

Option	Description
Start	Start highlighting free subdisks immediately and continue to do so until instructed to stop.
Stop	Stop highlighting free subdisks.

C.6.3 Projection Relationships

Table C–1 summarizes the projection relationships that are highlighted for particular icon types. If no icons of the correct type are associated with the selected icon, then nothing is highlighted.

Table C–1: Projection Table

Icon Selected	Icons Highlighted
Volume	All subdisks associated with any plex associated with the volume
Plex	All subdisks associated with the plex
Subdisk	Associated plex and volume, and all other subdisks associated with the plex
LSM Disk	All plexes associated with the subdisks that reside on the disk

C.7 Analysis

Analysis is the LSM Visual Administrator's way of displaying statistics on the performance of various LSM objects.

Statistics are displayed both visually (via color or pattern) and numerically (via pop-up statistics forms).

C.7.1 Analysis Menus

The following menu selections are accessed via the Analyze menu:

- Start

  Analyze → Start

  This menu selection begins analysis of the selected icons. These icons are added to the list of objects being analyzed. Only volume and LSM disk icons can be analyzed. Once analysis is activated, the selected icons begin to display information about their performance characteristics.

`Requirement`: At least one volume or LSM disk icon must be selected.

- Stop

  Analyze → Stop

  This menu selection terminates analysis of the selected icons. These icons are removed from the list of objects being analyzed. When analysis stops, the selected icons return to their preanalysis states. When analysis is stopped for one icon, other icons undergoing analysis are not affected.

  `Requirements`:

  – At least one volume or LSM disk icon must be selected.

  – The selected icons must be undergoing analysis.

- Stop All

  Analyze → Stop All

  This menu selection automatically terminates analysis of all icons in all views. All icons return to their preanalysis states.

  `Requirements`: Analysis must be in effect.

- `Parameters`

  Analyze → Parameters

  This menu selection accesses the Analysis Parameters form, which is used to set user preferences for how analysis is to be conducted.

  `Form`: Analysis Parameters Form (described in the Analysis Forms section).

C.7.2 Analysis Forms

The following forms are accessed via the Analyze menu:

- Analysis Parameters Form

  Analyze → Parameters

  This form is used to set user preferences for conducting analysis. The following table describes the fields in this form.

  Most fields in this form are required; those that are optional are listed here. Fields in this form are read/write fields, unless listed as read only.

Field	Description
Sample Rate:	Determines the time interval between data samples. This field is divided into two sections: the slider bar is used to select the interval (1-60) and the menu to the right is used to select units of time (seconds or minutes). The default is 5 seconds. A shorter interval means the data will be updated more often, but is also a higher load on the system.
Volume Parameters:	Specifies the high and low values that decide the coloring (or pattern) of the volume icons.
Disk Parameters:	Specifies the high and low values that decide the coloring (or pattern) of the LSM disk icons.
Subdisk Parameters:	Specifies the high and low values that decide the coloring (or pattern) of the subdisk icons.
Log File:	The name of the file for the statistics log. If the file does not exist, it is binary file created. The file name is taken to be relative unless a path name is given. To stop logging to the file, delete the file name text in this field. This field is optional. To view the log file, you must run `/usr/bin/lsmlog2text` *filename* on the file to process it for viewing.

`Requirements:`

- For each set of high/low parameters, the high parameter must be greater than the low parameter.
- The user must have access to the specified log file.

• Analysis Statistics Form

To access the analysis statistics form, click MB3 on desired the icon that is being analyzed.

This form displays analysis statistics relevant to the selected volume or LSM disk icon. This form applies only to volume or disk icons that are undergoing analysis. The following table describes the fields in this form. All fields in this form are read only and cannot be changed.

Field	Description
Reads:	The number of times the object was read from during the last interval.
Writes:	The number of times the object was written to during the last interval.
Total R/W:	The total number of reads and writes during the last interval.
Blocks Read:	The number of disk blocks read from the object during the last interval.

Field	Description
Blocks Written:	The number of disk blocks written to the object during the last interval.
Total Blocks:	The total number of blocks read from or written to the object during the last interval.
Avg Read Time:	The average time, in milliseconds, that it took for a read operation to complete. This is equal to the number of number of reads during the last interval divided by the total time spent on reads.
Avg Write Time:	The average time, in milliseconds, that it took for a write operation to complete. This is equal to the number of writes during the last interval divided by the total time spent on writes.
Interval:	The actual time, in seconds, since the last data was sampled. This may vary slightly from the specified interval time due to uncontrollable variances from system to system.

`Requirements:` The icon selected by clicking MB3 must be undergoing analysis.

C.7.3 Analysis Table

Table C–2 summarizes the default colors and patterns associated with the various levels of analysis. These defaults can be changed using the dxlsm-related X resources. See the `dxlsm`(8) reference page for more information.

Table C–2: Analysis Table

Analysis Level	Color	Bitmap Pattern
low	green	cross_weave
medium	yellow	root_weave
high	red	wide_weave

C.8 UFS Management

This section provides information on Visual Administrator UFS file system operations. You access UFS operations via the Basic-Ops menu. This menu is located in view windows, such as View of the `rootdg` disk group . This menu provides access to UFS operations involving general file system maintenance, and is accessed by selecting:

Basic-Ops → UFS Operations

You can access the following menu selections via the Basic-Ops menu.

- Create
- Make File System
- Mount
- Unmount
- Check File System (fsck)
- Display Properties
- Help

The Help selection accesses a Help window which displays information relevant to the available file system operations.

C.8.1 File System Menus

The following list describes the file system operations menu items:

- Create

  Basic-Ops → UFS Operations → Create

  This operation creates a file system on an underlying volume. This is done by creating a volume on one or more disks and then creating the file system on that volume.

  You can select one or more disks on which to create the volume (providing that there is sufficient space on the disks). If you do not specify any disks, LSM automatically determines which disks to use based on available free space.

  From the Create menu, select the type of volume to be created from a submenu listing two of the basic types of volumes:

Type	Description
Simple	Creates a simple, concatenated volume whose subdisks are arranged both sequentially and contiguously within a plex.
Striped	Creates a volume with data spread fairly evenly across multiple disks by way of striping. Stripes are relatively small, equally sized fragments that are allocated alternately to the subdisks of each plex.

  If a mirrored volume is desired, a simple or striped volume must be created and then mirrored using the Add Mirror option from the Volume Operations menu.

  `Requirements:`

  – Only disks in the same disk group can be selected.

– Only LSM disks (disks under LSM control) can be selected.

– If striping is to be in effect, at least two disks are required in order for the operation to succeed.

`Forms`: Simple Volume/FS Create Form and Striped Volume/FS Create Form.

• Make File System

Basic-Ops → UFS Operations → Make File System

This operation is used to make a file system on an existing volume. The user selects the volume on which to place the new file system, and specifies the mount point if the file system is to be mounted immediately.

`Requirements:`

– A volume icon must be selected.

– The selected volume must be enabled.

– Only one mounted file system can exist on each volume.

`Form`: Make File System Form (described in Section C.8.2).

• Mount

Basic-Ops → UFS Operations → Mount

This operation mounts the file system that resides on the selected volume. This operation assumes that the selected volume already contains a valid file system. The Visual Administrator has no way of knowing whether a valid, unmounted file system already exists on a given volume. You must make sure of the existence of an unmounted file system on a volume, as well as that file system's type.

`Requirements:`

– A volume icon must be selected.

– A valid, unmounted file system must already exist on the selected volume.

`Form`: Mount File System Form (described in Section C.8.2).

• Unmount

Basic-Ops → UFS Operations → Unmount

This operation is used to unmount the file systems that resides on the selected volumes. The file system can be unmounted only if the mount point is not busy.

`Requirements:`

– At least one volume icon must be selected.

– The selected volume must contain a mounted file system.

- Check File System

  Basic-Ops → UFS Operations → Check File System (fsck)

  This operation checks the file systems on the selected volumes for consistency (using `fsck`). The file system to be checked must currently be unmounted.

  `Requirements:`

  – At least one volume icon must be selected.

  – The selected volumes must contain an unmounted file system.

  `Form:` File System Check Form (described in Section C.8.2).

- Display Properties

  Basic-Ops → UFS Operations → Display Properties

  Display information for file systems mounted on the system. You can select the file system for which information is to be displayed from a list of all mounted file systems. If a volume is selected, the properties for the file system that resides on that volume is displayed by default.

C.8.2 File System Forms

Some file system operations result in the appearance of forms that you must complete in order for that operation to proceed. Most forms provide a Help button that provides access to information relevant to the fields and other aspects of that particular form.

C.8.2.1 Basic-Ops Forms

The following list describes how to access forms via file system-related selections from the Basic-Ops menu:

- Simple Volume/FS Create Form

  Basic-Ops → UFS Operations → Create → Simple

  This form creates a concatenated volume and then creates a file system on the new volume. The form is divided into two sections, one for volume creation and the other for file system creation. Most of the form fields are already set to the defaults for the creation of a new volume and file system. The following table describes the fields for this form.

  Most fields in this form are required; those that are optional are listed here. Fields in this form are read/write fields, unless listed as read only.

Field	Description
Volume name:	The name of the volume to be created. The name must be unique within this disk group. The maximum length of this field is 31 characters.
Volume size:	The desired volume size. The size should be entered as a number followed immediately by the letter `k`, `m`, or *s* to indicate kilobytes, megabytes, or sectors, respectively. If no unit is specified, the default is sectors. The volume size should be less than or equal to the available free space of the disks.
Usage Type:	The desired usage type. The `fsgen` file system is the generic usage type, which assumes that the volume is being used by a file system. The `gen` file system is the generic usage type, which makes no assumptions regarding the data content of the volume. The default is `fsgen`.
Create file system:	Indicates whether a file system is to be created. When this form is invoked from the UFS Operations menu, the default is to create a file system (Yes). All fields below this field are only accessible when Yes is specified here.
FS type:	UFS is the only currently supported file system type.
Mount file system:	Indicates whether the file system should be mounted after creation. If the answer is Yes (the default), a mount point must also be specified in the next field. All fields below this field are only accessible when Yes is specified here.
Mount point:	The desired mount point for the new file system. If the specified mount point does not already exist, the Visual Administrator automatically creates it. This field is required if the file system is to be mounted.
Mount automatically:	Indicates whether this file system should be mounted every time the system comes up (by placing an entry in `/etc/fstab`). The default is Yes.

- Striped Volume/FS Create Form

  Basic-Ops → UFS Operations → Create → Striped

  This form creates a striped volume and creates a file system on the new volume. The form is divided into two sections, one for volume creation and the other for file system creation. Most of the form fields are already set to the defaults for the creation of a new volume. The following table describes the fields for this form.

  Most fields in this form are required; those that are optional are listed here. Fields in this form are read/write fields, unless listed as read only.

Field	Description
Volume name:	The name of the volume to be created. The name must be unique within this disk group. The maximum length of this field is 31 characters.
Volume size:	The desired volume size. The size should be entered as a number followed immediately by the letter `k`, `m`, or *s* to indicate kilobytes, megabytes, or sectors, respectively. If no unit is specified, the default is sectors. If the size is not wholly divisible by the stripe width, LSM adjusts the volume size up to the next even multiple in order to create the volume. For a striped volume, the volume size should be calculated as follows: *vol_size* = *stripe_width* * *number_of_stripes* * *n*, where *n* is a number greater than zero. The volume size should be less than or equal to the available free space of the disks.
Usage Type:	The desired usage type. The `fsgen` type is the file system generic usage type, which assumes that the volume is being used by a file system. The `gen` type is the generic usage type, which makes no assumptions regarding the data content of the volume. The default is `fsgen`.
Number of Stripes:	The number of stripes that the volume's plex is to have. This is effectively the number of disks on which the volume is to be created. If some number of disks have already been selected, that number of stripes appears in this field. This number corresponds to the number of disks across which data will be striped. If no number is specified, an appropriate number (usually 2) is used.
Stripe width:	The width of the stripes on the plex that this volume will have. The value specified may be optimized for the particular application. However, the default value of 128 sectors is as a good stripe width for most systems.
Create file system:	Indicates whether a file system is to be created. When this form is invoked from the UFS Operations menu, the default is to create a file system (Yes). All fields below this field are only accessible when Yes is specified here.
FS type:	UFS is the only currently supported file system type.
Mount file system:	Indicates whether the file system should be mounted after creation. If the answer is Yes (the default), a mount point must also be specified in the next field. All fields below this field are only accessible when Yes is specified here.

Field	Description
Mount point:	The desired mount point for the new file system. If the specified mount point does not already exist, the Visual Administrator automatically creates it. This field is required if the file system is to be mounted.
Mount automatically:	Indicates whether this file system should be mounted every time the system comes up (by placing an entry in `/etc/fstab`). The default is Yes.

- Make File System Form

Basic-Ops → UFS Operations → Make

This form is used to make a file system (using `newfs`) according to your specifications. The following table describes the fields for this form.

Most fields in this form are required; those that are optional are listed here. Fields in this form are read/write fields, unless they are listed as read only.

Field	Description
Device name:	Displays the block device on which to make the file system, which corresponds to the name of the selected volume. This field is read only and cannot be changed.
File system size:	Displays the length of the file system to be made. If no units are specified, sectors are assumed. This length should typically correspond to the length of the volume on which the file system is to be made, although it can be altered for special circumstances.
FS Type:	UFS is the only currently supported file system type.
Mount file system:	Indicates whether the file system should be mounted after creation. If the answer is Yes (the default), a mount point must also be specified in the next field. All fields below this field are only accessible when Yes is specified here.
Mount point:	The desired mount point for the new file system. If the specified mount point does not already exist, the Visual Administrator automatically creates it. This field is required if the file system is to be mounted.
Mount automatically:	Indicates whether this file system should be mounted every time the system comes up (by placing an entry in `/etc/fstab`). Yes is the default.

- Mount File System Form

Basic-Ops → UFS Operations → Mount

This form is used to mount a file system that already exists on a selected volume. The following table describes the fields for this form.

Most fields in this form are required; those that are optional are listed here. Fields in this form are read/write fields, unless listed as read only.

Field	Description
Device name:	Displays the block device on which to make the file system, which corresponds to the name of the selected volume. This field is read only and cannot be changed.
FS Type:	UFS is the only currently supported file system type.
Mount point:	The desired mount point for the file system. If the specified mount point does not already exist, the Visual Administrator automatically creates it. The Visual Administrator attempts to provide a default mount point, which it obtains by scanning `/etc/fstab`.
Mount automatically:	Indicates whether this file system should be mounted every time the system comes up (by placing an entry in `/etc/fstab`). No is the default.

- File System Check Form

Basic-Ops → UFS Operations → Check File System (fsck)

This form is used to check a file system that exists on a volume but is not currently mounted. The following table describes the fields for this form.

The fields in this form are required. Fields in this form are read/write fields, unless listed as read only.

Field	Description
Volume:	Displays the name of the volume containing the file system to be checked (with `fsck`). This field is read only and cannot be changed.
FS type:	Indicates the type of the file system to be checked.

C.8.2.2 File Systems Properties Form

The following discussion describes the properties form. This form reveals the properties of a particular file system:

- File System Properties Form

Basic-Ops → UFS Operations → Display Properties

This form provides detailed information on the attributes of a particular file system. This properties form contains a list of mounted file systems, from which you can select the file system whose properties are to be displayed. The following table describes the fields for this form.

All fields in this form are read only and cannot be changed.

Field	Description
Mount Point:	The mount point of this file system.
Device:	The block device on which this file system resides.
Block Size:	The block size of this file system.
Default block size:	Fundamental file system block size.
Total disk space:	Number of megabytes of disk storage on this file system available on the disk.
Disk space available:	Number of megabytes of disk storage on this file system that are available for use.
Capacity:	Percentage of the total disk storage space still available for use. This is the free space available divided by the total disk space.
Total files:	The maximum number of files allowed on this file system.
Free files available:	The number of files that still may be created on this file system.
FS type:	The file system type (such as `UFS`).
Max file name length:	The maximum number of characters for a file name on this file system. This restriction is imposed by the file system.
FS attributes:	Indicates attributes associated with this file system: `Read Only` indicates a file system that cannot be written to.

D

The voldiskadm Menu Interface

This appendix describes the `voldiskadm` menu interface that you can use to perform LSM disk and disk group operations. The menus are easy to use and provide information about each step to help you decide the correct response for each prompt.

D.1 Starting the voldiskadm Menu Interface

To start the `voldiskadm` menu interface, enter:

```
# voldiskadm
```

The following output is displayed:

```
1       Add or initialize one or more disks
2       Encapsulate one or more disks
3       Remove a disk
4       Remove a disk for replacement
5       Replace a failed or removed disk
6       Mirror volumes on a disk
7       Move volumes from a disk
8       Enable access to (import) a disk group
9       Remove access to (deport) a disk group
10      Enable (online) a disk device
11      Disable (offline) a disk device
12      Mark a disk as a spare for a disk group
13      Turn off the spare flag on a disk
14      Recover plexes and volumes after disk replacement
list    List disk information
?       Display help about menu
??      Display help about the menuing system
q       Exit from menus
```

D.2 Disk Management

This section describes the disk management available with the `voldiskadm` menu interface.

D.2.1 Initializing a Disk

Disk initialization identifies a disk to LSM and prepares the disk for LSM use. This operation involves installing a disk header and writing an empty configuration on the disk. A disk access record is created for the disk, unless such a record already exists.

Note

A disk must contain a disklabel before you can initialize it for LSM use.

To initialize a disk for use with the LSM software, perform the following steps:

1. At the main menu prompt, select menu item 1 to enable the “Add or initialize one or more disk” operation.
2. At the prompt on the following “Add or initialize one or more disk” screen, enter the address of the disk to be added. If you do not know the address of the disk you want to add, enter the letter `l` or type the word `list` at the prompt. LSM displays a list of the disks that are on the system. For example:

```
Select disk devices to add:

[<space-separated disk list>,<disk>,list,q,?] list

DEVICE       DISK         GROUP        STATUS
dsk2         dsk2         rootdg       online
dsk3         dsk3         rootdg       online
dsk4         dsk4         rootdg       online
dsk5         dsk5         rootdg       online
dsk6         -            -            online
dsk7         -            -            online
dsk8         -            -            unknown
  :
```

3. Once you have entered the disk name, LSM displays this screen that asks you to supply the name of the disk group you want the disk to be a part of:

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

You can:

- Press the Return key to accept the default disk group name, `rootdg`.
- Specify the name of a disk group to add the disk to an existing disk group.
- Create a new disk group and add the disk to it. To create a new disk group, enter a disk group name that does not yet exist.
- Do not specify a disk group and leave the disk available for use by future add or replacement operations. To leave the disk available for future use, enter a disk group name of none. Enter none if:
  – The disk group you want the disk to be a part of does not exist yet.

– You want to keep this disk available as a spare to be used as a replacement disk.

4. Depending on your response to the “Which disk group...” prompt, LSM displays one of the following screens.

   – If you entered none, LSM displays the following:

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

The disk will be initialized and left free for use as a replacement
disk.

dsk8

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

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.

dsk8
Initialize this device? [y,n,q,?] (default: y)
Initializing device dsk8.
```

   – If you selected `rootdg` as the disk group, LSM displays the following screen:

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

The default disk name that will be assigned is:

disk01

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

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

The selected disks will be added to the disk group rootdg with the
default disk names.

dsk8

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

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.

dsk8

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

Initializing device dsk8.

 Adding disk device dsk8 to disk group rootdg with disk
 name disk01.
```

   Press the Return key to continue.

5. If LSM successfully completes the disk initialization, the following appears:

   ```
   Add or initialize other disks? [y,n,q,?] (default: n)
   ```

   If the `fstype` in the disk label of the specified partition or an overlapping partition is set, LSM displays a warning message to inform you that initializing the disk might destroy existing data.

   If you are sure that the disk partition has no valid data and that the partition can be added to LSM, you can ignore the warning message and answer `y` to the prompt. The `voldiskadm` utility then proceeds to initialize the disk partition and add it to LSM.

   If the disk cannot be initialized because the specified partition or an overlapping partition on the disk is open (that is, a partition is actively in use by UFS, AdvFS, LSM or swap), the initialization process fails and `voldiskadm` issues an error message informing you of the problem.

D.2.2 Displaying Disk Information

The following steps describe how to find information about disks available on the system.

1. From the main menu, enter the letter `l` or type the word `list` to display a list of disks available on the system.

   LSM displays a list of devices similar to the following screen, and prompts you to enter the address of the disk for which you want to obtain detailed information.

   ```
   Select an operation to perform: list

   List disk information
   Menu: VolumeManager/Disk/ListDisk

   Use this menu operation to display a list of disks.  You can
   also choose to list detailed information about the disk at
   a specific disk device address.

   Enter disk device or "all" [<address>,all,q,?] (default: all)

   DEVICE       DISK         GROUP        STATUS
   dsk2         dsk2         rootdg       online
   dsk3         dsk3         rootdg       online
   dsk4         dsk4         rootdg       online
   dsk5         dsk5         rootdg       online
   dsk6         -            -            online
   dsk7         -            -            online
   dsk8         disk01       rootdg       online
    :
   ```

2. The following screen displays information for the disk called `dsk8`:

   ```
   Device to list in detail [<address>,none,q,?] (default: none) dsk8
   ```

```
Device:    dsk8
devicetag: dsk8
type:      sliced
hostid:    rio.dec.com
disk:      name=disk01 id=922907065.1771.rio.dec.com
group:     name=rootdg id=921709207.1025.rio.dec.com
flags:     online ready autoimport imported
pubpaths:  block=/dev/disk/dsk8g char=/dev/rdisk/dsk8g
privpaths: block=/dev/disk/dsk8h char=/dev/rdisk/dsk8h
version:   2.1
iosize:    min=512 (bytes) max=32768 (blocks)
public:    slice=6 offset=16 len=4106368
private:   slice=7 offset=0 len=4096
update:    time=922907069 seqno=0.5
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

List another disk device? [y,n,q,?] (default: n)
```

3. Press the Return key to return to the main menu.

D.2.3 Adding a Disk to a Disk Group

You may want to add a new disk to an already established disk group. Perhaps the current disks have insufficient space for the project or work group requirements, especially if these requirements have changed.

Follow these steps to add a disk to a disk group:

1. Follow the instructions documented in steps 1 and 2 in Section D.2.1.
2. When the add disk operation adds a disk to a disk group, LSM checks to see if the disk is already initialized. If the disk is initialized, LSM displays the following screen and asks whether or not you want to reinitialize the disk:

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

    The default disk name that will be assigned is:

    dg101

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

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

    The selected disks will be added to the disk group dg1 with the
    default disk names.

    dsk8

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

      The following disk device appears to have been initialized already.
```

```
  The disk is currently available as a replacement disk.

 dsk8

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

The following disk you selected for use appears to already have
been initialized for the Logical Storage Manager.  If you are
certain the disk has already been initialized for the Logical
Storage Manager, then you do not need to reinitialize the disk
device.

dsk8

  Reinitialize this device? [y,n,q,?] (default: y)
```

Use the information in the following table to determine whether or not you should reinitialize the disk.

If...	Then...
The disk is new	Initialize the disk before placing it under the control of LSM.
The disk was previously in use and contains useful data	Do not initialize the disk. Instead, use the LSM encapsulation function to add the disk to the LSM system while still preserving the existing data.
The disk was previously in use but it does not contain useful data	Initialize the disk before placing the disk under LSM control.

D.2.4 Moving Volumes from a Disk

Before you disable or remove a disk, you may want to move the data from that disk to other disks on the system. Use this operation immediately prior to removing a disk, either permanently or for replacement (described in Section D.2.5).

Note

Simply moving volumes off of a disk without also removing the disk, does not prevent volumes from being moved onto the disk by future operations. For example, two consecutive move operations could move volumes from one disk to another and then back. Also note that you need to make sure the other disks in the disk group have sufficient space available.

To move volumes from a disk, do the following:

1. Select menu item 7 from the main menu.

2. From the “Move volumes from a disk” screen, enter the name of the disk whose volumes you want to move:

```
Use this menu operation to move any volumes that are using a
disk onto other disks.  Use this menu immediately prior to
removing a disk, either permanently or for replacement.  You can
specify a list of disks to move volumes onto, or you can move the
volumes to any available disk space in the same disk group.

    NOTE:  Simply moving volumes off of a disk, without also removing
           the disk, does not prevent volumes from being moved onto
           the disk by future operations.  For example, using two
           consecutive move operations may move volumes from the
           second disk to the first.

  Enter disk name [<disk>,list,q,?] dsk5
```

3. Enter the name of the disk that the volumes should be moved to:

```
You can now specify a list of disks to move onto.  Specify a list
of disk media names (e.g., rootdg01) all on one line separated by
blanks.  If you do not enter any disk media names, then the volumes
will be moved to any available space in the disk group.

Enter disks [<disk ...>,list,q,?] dsk4

    Requested operation is to move all volumes from disk dsk5 in
    group rootdg.

      NOTE: This operation can take a long time to complete.

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

4. As LSM moves the volumes from the disk, it displays the status of the operation:

```
Move volume v1 ...
```

5. When the volumes have all been moved, LSM displays the following:

```
Evacuation of disk dsk5 is complete.
Move volumes from another disk? [y,n,q,?] (default: n)
```

D.2.5 Removing a Disk from a Disk Group

This operation involves removing the LSM disk associated with the selected partitions from LSM control by removing the associated disk access records. The `voldiskadm` menu interface provides two methods for removing disks. These two operations remove a disk as follows:

- Menu item 3, Remove a disk — Removes a disk completely from LSM control and does not retain the disk name.
- Menu item 4, Remove a disk for replacement — Removes a failed disk and retains the disk name so it can be replaced with another disk.

See Section D.2.4, which describes how to move data from a disk to another disk on the system, and see Section D.2.6, which describes how to replace a failed or removed disk.

Note

You must disable the disk group before you can remove the last disk in that group. Disabling a disk group, also referred to as deporting a disk group, is described in Section D.3.2.

D.2.5.1 Removing a Disk Without Replacement

To remove a disk from its disk group, perform the following steps:

1. Select menu item 3 from the main menu.
2. LSM displays the following "Remove a disk" screen and prompts you to enter the disk name of the disk to be removed.

   ```
   Use this operation to remove a disk from a disk group.  This
   operation takes, as input, a disk name.  This is the same name
   that you gave to the disk when you added the disk to the disk
   group.

     Enter disk name [<disk>,list,q,?] dsk5
   ```

   The example removes a disk called dsk5.
3. LSM displays a verification screen and asks whether or not to continue:

   ```
   Requested operation is to remove disk dsk5 from group rootdg.

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

   Press the Return key to continue.
4. LSM removes the disk from the disk group and then displays the following screen when the operation has completed:

   ```
   Removal of disk dsk5 is complete.

     Remove another disk? [y,n,q,?] (default: n)
   ```

   Press the Return key to return to the main menu.

D.2.5.2 Removing a Disk for Replacement

You may occasionally need to replace a disk in a disk group. This operation involves initializing the disk for LSM use, and replacing the old disk and associated disk media records with the new disk and its information. Perform the following steps to replace a disk while retaining the disk name:

1. Select menu item 4 from the main menu. LSM displays the "Remove a disk for replacement" screen.
2. Enter the name of the disk to be replaced if you know it. Otherwise, enter the letter `l` for a list of disks. LSM displays a screen similar to the following:

```
Use this menu operation to remove a physical disk from a disk
group, while retaining the disk name.  This changes the state
for the disk name to a "removed" disk.  If there are any
initialized disks that are not part of a disk group, you will be
given the option of using one of these disks as a replacement.


  Enter disk name [<disk>,list,q,?]
```

3. If this operation does not need to disable a volume in order to replace the disk, the following is displayed:

```
The following volumes will lose mirrors as a result of this
operation:

v1

 No data on these volumes will be lost.

If this operation must disable a volume in order to replace the disk,
the following will be displayed:

The following volumes will be disabled as a result of this
operation:

v1

These volumes will require restoration from backup.

Are you sure you want to do this? [y,n,q,?] (default: n)

Warning: Disabling a volume that is currently open and in use
will return I/O errors back to the file system or
application using that volume.  Disabling a volume
that contains a mounted file system may crash the
system.  Refer to section C.2.4 on how to move
volumes off a disk before removing the disk.
```

4. If there are any initialized disks available that are not part of a disk group, LSM displays the following screen and gives you the option of using one of these disks as a replacement. Select the replacement disk from the list provided. Press the Return key if you want to use the default disk.

```
The following devices are available as replacements:

            dsk5

      You can choose one of these disks now, to replace dsk4.
      Select "none" if you do not wish to select a replacement disk.

    Choose a device, or select "none"
    [<device>,none,q,?] (default: dsk5)
```

5. LSM then displays the following verification screen:

```
Requested operation is to remove disk dsk4 from group rootdg.
The removed disk will be replaced with disk device dsk5.

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

Press the Return key to continue.

6. When LSM successfully replaces the disk, LSM displays the following screen:

```
Removal of disk dsk4 completed successfully.

Proceeding to replace dsk4 with device dsk5.

Disk replacement completed successfully.

Remove another disk? [y,n,q,?] (default: n)
```

Press the Return key to return to the main menu.

D.2.6 Replacing a Failed or Removed Disk

Use this menu operation to specify a replacement disk for a disk that you removed with the "Remove a disk for replacement" menu operation (see Section D.2.5.2), or for a disk that failed during use. To replace a disk, use the following instructions:

1. Select menu item 5 from the main menu.
2. The following screen asks you to enter the name of the disk to be replaced. You can choose an uninitialized diskto be used to replace the failed or removed disk:

```
replace the failed or removed disk:

    Use this menu operation to specify a replacement disk for a disk
    that you removed with the "Remove a disk for replacement" menu
    operation, or that failed during use.  You will be prompted for
    a disk name to replace and a disk device to use as a replacement.
    You can choose an uninitialized disk, in which case the disk will
    be initialized, or you can choose a disk that you have already
    initialized using the Add or initialize a disk menu operation.


  Select a removed or failed disk [<disk>,list,q,?] list

  Disk group: rootdg

  DM NAME         DEVICE       TYPE     PRIVLEN  PUBLEN   STATE

  dm dsk5         -            -        -        -        NODEVICE

  Select a removed or failed disk [<disk>,list,q,?] dsk5

  Select disk device to initialize [<address>,list,q,?] list

  DEVICE       DISK         GROUP        STATUS
  dsk2         dsk2         rootdg       online
  dsk3         dsk3         rootdg       online
  dsk4         dsk4         rootdg       online
  dsk5         -            -            online
  dsk10        -            -            unknown


  Select disk device to initialize [<address>,list,q,?] dsk10
```

```
  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.

  dsk10

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

  The requested operation is to initialize disk device dsk10 and
  to then use that device to replace the removed or failed disk
  dsk5 in disk group rootdg.

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

3. LSM displays the following success screen:

```
Replacement of disk dsk5 in group rootdg with disk device
    dsk10 completed successfully.

 Replace another disk? [y,n,q,?] (default: n)
```

Press Return to return to the main menu.

D.2.7 Disabling a Disk

This operation places the disk access record in an offline state. During searches for disk IDs or members of a disk group, offline disks are ignored. To disable a disk, perform the following steps:

1. Select menu item 11 from the main menu to disable (off line) a removable disk.
2. On the next screen, select the disk you want to disable:

```
Use this menu operation to disable all access to a disk device
by the Logical Storage Manager.  This operation can be applied
only to disks that are not currently in a disk group.  Use this
operation if you intend to remove a disk from a system without
rebooting.

   NOTE:  Many systems do not support disks that can be removed from
          a system during normal operation.  On such systems, the
          offline operation is seldom useful.

 Select a disk device to disable [<address>,list,q,?] dsk3
```

This example shows that disk `dsk3` has been selected.

3. LSM disables disk `dsk3` and then asks if you want to disable another device:

```
Disable another device? [y,n,q,?] (default: n)
```

Press the Return key to return to the main menu.

D.3 Disk Group Management

This section describes the disk group management that you can perform with the the `voldiskadm` menu interface.

D.3.1 Importing a Disk Group

Use this menu operation to enable access by this system or cluster to a disk group. This operation can be used to move a disk group from one system to another. If you want to move a disk group from one system to another you must first disable (deport) it on the original system (see Section D.3.2), then move the disk between systems and enable (import) the disk group.

Note

If two hosts share a SCSI bus, make sure that the other host really failed or deported the disk group. If two hosts import a disk group at the same time, the disk group will be corrupted and become unusable.

To import a disk group, do the following:

1. Select menu item 8 from the main menu. From the “Enable access to (import) a disk group” menu, select the name of the disk group to import:

```
Use this operation to enable access to a disk group.  This can be
used as the final part of moving a disk group from one system to
another.  The first part of moving a disk group is to use the
"Remove access to (deport) a disk group" operation on the
original host.

A disk group can be imported from another host that failed without
first deporting the disk group.  Be sure that all disks in the disk
group are moved between hosts.

If two hosts share a SCSI bus, be very careful to ensure that the
other host really has failed or has deported the disk group.  If
two active hosts import a disk group at the same time, the disk
group will be corrupted and will become unusable.

Select disk group to import [<group>,list,q,?] (default: list) list

        GROUP: dg1 (id: 921709259.1071.rio.dec.com)
        DEVICES:
                 dsk7

        GROUP: dg1 (id: 922382892.1625.rio.dec.com)
        DEVICES:
                 dsk9

        GROUP: dg1 (id: 922908695.1779.rio.dec.com)
        DEVICES:
                 dsk6
```

```
                dsk8

 Select disk group to import [<group>,list,q,?] (default: list) dg1
```

2. Once the import is complete, LSM displays the following success screen:

   ```
   The import of dg1 was successful.

   Select another disk group? [y,n,q,?] (default: n)
   ```

   Press Return to return to the main menu

D.3.2 Deporting a Disk Group

Use this operation to disable access to a disk group that is currently enabled (imported) by this system. Deport a disk group if you intend to move the disks in a disk group to another system. Also, deport a disk group if you want to use all of the disks remaining in a disk group for some new purpose.

Note

For removable disk devices on some systems, it is important to disable all access to the disk before removing the disk.

To deport a disk group, perform the following steps:

1. Select menu item 9 from the main menu.
2. From the following “Remove access to (deport) a disk group” menu, enter the name of the disk group to be deported:

   ```
   Use this menu operation to remove access to a disk group that
   is currently enabled (imported) by this system.  Deport a disk
   group if you intend to move the disks in a disk group to another
   system.  Also, deport a disk group if you want to use all of the
   disks remaining in a disk group for some new purpose.

   You will be prompted for the name of a disk group.  You will
   also be asked if the disks should be disabled (offlined).  For
   removable disk devices on some systems, it is important to
   disable all access to the disk before removing the disk.

   Enter name of disk group [<group>,list,q,?] (default: list) list

   GROUP        DISK/VOLUME         DEVICE/STATE LENGTH

   dg1          disk   dg101        dsk8         4106368
   dg1          disk   dsk6         dsk6         4109440

   Enter name of disk group [<group>,list,q,?] (default: list) dg1
   ```

3. Verify that you want LSM to disable the disk group on the displayed confirmation screen:

```
The requested operation is to disable access to the removable
disk group named dg1.  This disk group is stored on the
following disks:

          dsk6 on device dsk6
          dg101 on device dsk8

You can choose to disable access to (also known as "offline")
these disks.  This may be necessary to prevent errors if
you actually remove any of the disks from the system.

Disable (offline) the indicated disks? [y,n,q,?] (default: n)

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

Press Return to deport the disk group.

4. Output similar to the following is displayed:

```
Removal of disk group dg1 was successful.

Disable another disk group? [y,n,q,?] (default: n)
```

D.4 Mirror Volume Management

You can only use the vvoldiskadm menu interface to mirror a volume. You cannot use the voldiskadm menu interface to mirror volumes that are already mirrored, or that are comprised of more than one subdisk.

To mirror volumes on a disk, make sure that the target disk has an equal or greater amount of space as the originating disk and then do the following:

1. Select menu item 6 from the main menu.
2. On the “Mirror volumes on a disk” menu, enter the name of the disk whose volumes you want to mirror:

```
This operation can be used to mirror volumes on a disk.  The volumes
can be be mirrored onto another disk or onto any available disk space.
Volumes will not be mirrored if they are already mirrored or contain
more than one subdisk.

Mirroring the root and swap volumes from the boot disk will produce a
disk that can be used as an alternate boot disk.

At the prompt below, supply the disk media name containing the
volumes to be mirrored.

Enter disk name [<disk>,list,q,?] dsk5
```

3. Select the target disk name (this disk must be the same size or larger than the originating disk). Volumes can be mirrored onto another disk or onto any available disk space.

```
You can choose to mirror volumes from disk dsk5 onto any
available disk space, or you can choose to mirror onto a specific
disk.  To mirror to a specific disk, select the name of that disk.
To mirror to any available disk space, select "any".
```

```
Enter destination disk [&lt;disk&gt;,list,q,?] (default: any) dsk4
```

4. LSM displays the verification screen. Press Return to make the mirror.

```
The requested operation is to mirror all volumes on disk dsk5
in disk group rootdg onto available disk space on disk dsk4.

There is space already allocated on disk dsk4.  If you don’t
want to mirror onto this disk, enter "n" at the next prompt and
restart this operation from the beginning.

NOTE: This operation can take a long time to complete.

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

5. LSM displays the status of the operation as it performs the mirroring,

```
 Mirroring of disk dsk5 is complete.
```

6. Once LSM has completed the mirroring operation, it asks if you want to mirror volumes on another disk. Press Return to go back to the main menu.

```
Mirror volumes on another disk? [y,n,q,?] (default: n)
```

D.5 Exiting the voldiskadm Menu Interface

When you have completed all of your disk administration activities, exit the `voldiskadm` menu interface by selecting menu option `q` from the main menu.

E

LSM Error Messages

LSM is fault-tolerant and resolves most problems without system administrator intervention. If the volume daemon (`vold`) recognizes what actions are being taken, it can roll a transaction forward or back. In the event that `vold` is unable to recognize and fix system problems, you need to handle the task of problem solving.

This appendix describes the majority of informational, failure, and error messages displayed by `vold` and the kernel driver. These sections include some errors that are infrequently encountered and difficult to troubleshoot. Clarifications are included to elaborate on the situation or problem that may have generated a particular message. Wherever possible, a recovery procedure (user action) is provided to locate and correct potential problems.

Should it be necessary to contact your customer support organization, these messages are numbered for ease of reference.

E.1 Volume Daemon Error Messages

The following list contains the error messages associated with the volume daemon.

1. `Message:`

   -r must be followed by 'reset'

   `Clarification:` This message is caused by a usage error.

   `User Action:` Correct the usage and try again.

2. `Message:`

   -x [*argument* : prefix too long]

   `Clarification:` The stub-mode device path prefix name supplied exceeded the maximum of 32 characters.

   `User Action:` Select an alternate path for device files and retry the command.

3. `Message:`

   -x [*string* : invalid debug string]

   `Clarification:` An unknown argument string was given to the `-x` option to `vold`.

User Action: Select a valid string from the reference page for vold and try again.

4. Message:

Usage: vold [-dkf] [-r reset] [-m mode] [-x level] For detailed help use: vold help

Clarification: vold was invoked with an invalid set of arguments.

User Action: Correct the usage and try again or type vold help for more help. This is the full usage message from entering vold help:

```
Usage:  vold [-dkf] [-r reset] [-m mode] [-x level]
Recognized options:
 -d       set initial mode to disabled for transactions
 -k       kill the existing configuration daemon process
 -f       operate in foreground; default is background
 -r reset reset kernel state; requires 'reset' option argument
 -m mode  set vold's operating mode
          modes: disable, enable, bootload, bootstart
 -x level set debugging level to <debug>, 0 turns off debugging
 -R file  set filename for client request rendezvous
 -D file  set filename for client diag request rendezvous
```

5. Message:

lsm:vold: Error: volume [*volume_name* : Logging daemon killed by \ ] **signal** [*signal_number* [core dumped]]

Clarification: Someone killed the logging daemon.

User Action: If required, restart the daemon with a call to voliod logio.

6. Message:

lsm:vold: Error: /dev/volevent: [*error_message*] **lsm:vold: Error: cannot open /dev/volconfig:** [*error_message* lsm:vold: Error: Cannot kill existing daemon, pid= *process_id*]

Clarification: An attempt to kill an existing vold process with a SIGKILL signal has failed. This might be due to the process being in an unkillable kernel state perhaps because of a hung I/O or a missing I/O interrupt. There may be disk driver error messages in the /dev/osm buffer.

User Action: Try typing cat /dev/osm to see if any other messages have been output to the console device. If possible, use crash to determine the state of the process. If the process is asleep waiting for an I/O completion, then any disk driver error messages that have occurred might point to the solution. Failing this, a reboot is recommended.

7. Message:

lsm:vold: Error: /dev/voliod: VOL_LOGIOD_CHECK failed

`Clarification:` Some inconsistency between `vold` and the kernel has caused an ioctl to fail. This could be caused by the use of older versions of `vold` or the kernel, or it could be due to a bug in LSM.

`User Action:` Try stopping and restarting `vold`. If this fails then a reboot of the system should be attempted, possibly followed by a reinstall of the LSM package. If this fails, contact Customer Support.

8. `Message:`

   lsm:vold: Error: /dev/voliod: VOL_LOGIOD_KILL failed

   `Clarification:` Some inconsistency between `vold` and the kernel has caused an ioctl to fail. This could be caused by the use of older versions of `vold` or the kernel, or it could be due to a bug in LSM.

   `User Action:` Try stopping and restarting `vold`. If this fails then a reboot of the system should be attempted, possibly followed by a reconfiguration of LSM. If this fails, contact Customer Support.

9. `Message:`

   lsm:vold: Error: All transactions are disabled

   `Clarification:` This message may appear with the message `Disk group disabled by errors` if the disk group to be disabled is the root disk group. The continued use of the system could be dangerous because any configuration changes required (including error handling cases) could cause the loss of ability to perform I/O to a volume. Because this includes the root volume, this situation could, if uncorrected, cause the system to hang.

   `User Action:` This is a fatal error. All copies of the bootable root disk have failed. Recovery from this situation will require booting from floppy or from a disk unconnected with the LSM software. It may then be necessary to remove the LSM rootable disk configuration by using the `volunroot` command. See the LSM installation instructions for details. Once this has been achieved, the root disk group can be reinitialized to reestablish the database and log areas.

10. `Message:`

    lsm:vold: Error: Cannot get all disk groups from the kernel

    `Clarification:` Some inconsistency between `vold` and the kernel has caused an `ioctl` to fail. This could be caused by the use of older versions of `vold` or the kernel, or it could be due to a bug in LSM.

    `User Action:` Try stopping and restarting `vold`. If this fails then a reboot of the system should be attempted, possibly followed by a reinstall of the LSM package. If this fails, contact Customer Support.

11. `Message:`

lsm:vold: Error: Cannot get all disks from the kernel

Clarification: Some inconsistency between vold and the kernel has caused an ioctl to fail. This could be caused by the use of older versions of vold or the kernel, or it could be due to a bug in LSM.

User Action: Try stopping and restarting vold. If this fails then a reboot of the system should be attempted, possibly followed by an LSM reconfiguration. If this fails, contact Customer Support.

12. Message:

lsm:vold: Error: Cannot get kernel transaction state

Clarification: Some inconsistency between vold and the kernel has caused an ioctl to fail. This could be caused by the use of older versions of vold or the kernel, or it could be due to a bug in LSM.

User Action: Try stopping and restarting the vold daemon. If this fails then a reboot of the system should be attempted, possibly followed by a reconfiguration LSM. If this fails, contact Customer Support.

13. Message:

lsm:vold: Error: Cannot get private storage from kernel

Clarification:

User Action:

14. Message:

lsm:vold: Error: Cannot get private storage from kernel

Clarification: Some inconsistency between vold and the kernel has caused an ioctl to fail. This could be caused by the use of older versions of vold or the kernel, or it could be due to a bug in LSM.

User Action: Try stopping and restarting vold. If this fails then a reboot of the system should be attempted, possibly followed by a reconfiguration LSM. If this fails, contact Customer Support.

15. Message:

lsm:vold: Error: Cannot get private storage size from kernel

Clarification: Some inconsistency between vold and the kernel has caused an ioctl to fail. This could be caused by the use of older versions of vold or the kernel, or it could be due to a bug in LSM.

User Action: Try stopping and restarting vold. If this fails then a reboot of the system should be attempted, possibly followed by a reconfiguration LSM. If this fails, contact Customer Support.

16. Message:

lsm:vold: Error: Cannot get record [*name* from the kernel: *error_message*]

`Clarification:` Some inconsistency between `vold` and the kernel has caused an `ioctl` to fail. This could be caused by the use of older versions of `vold` or the kernel, or it could be due to a bug in LSM.

`User Action:` Try stopping and restarting `vold`. If this fails then a reboot of the system should be attempted, possibly followed by a reconfiguration LSM. If this fails, contact Customer Support.

17. `Message:`

    lsm:vold: Error: Cannot not make directory [*directory_path*]

    `Clarification:` When trying to create the specified directory, `vold` got a failure.

    `User Action:` Try creating the directory manually and then issue the command `voldctl enable`.

18. `Message:`

    lsm:vold: Error: Cannot recover operation in progress Failed to get group [*group_name* from the kernel]

    `Clarification:` Some inconsistency between `vold` and the kernel has caused an `ioctl` to fail. This could be caused by the use of older versions of `vold` or the kernel, or it could be due to a bug in LSM.

    `User Action:` Try stopping and restarting `vold`. If this fails then a reboot of the system should be attempted, possibly followed by a reconfiguration of LSM. If this fails, contact Customer Support.

19. `Message:`

    lsm:vold: Error: Cannot start [*usage_type* volume, no valid complete plexes]

    `Clarification:` No usable plexes remain for either the root or swap volume. This error is fatal and will result in the message `System startup failed` also appearing and the system being shutdown.

    `User Action:` This is generally an unrecoverable error and will likely require a reload of the system from backups.

20. `Message:`

    lsm:vold: Error: Cannot start [*usage_type* volume, no valid plexes]

    `Clarification` No usable plexes remain for either the root or swap volume. This error is fatal and will result in the message `System startup failed` also appearing and the system being shutdown.

    `User Action:` This is generally an unrecoverable error and will likely require a reload of the system from backups.

21. Message

lsm:vold: Error: Cannot start [*usage_type* volume, volume state is invalid]

Clarification: The volume is not in a state that can be recovered from. This might be because of corruption of the databases or because of an invalid use of the vold interfaces without the use of the utilities.

User Action: This is generally an unrecoverable error and will require reloading of the system from backups.

22. Message:

lsm:vold: Error: Cannot store private storage into the kernel

Clarification: Some inconsistency between vold and the kernel has caused an ioctl to fail. This could be caused by the use of older versions of vold or the kernel, or it could be due to a bug in LSM.

User Action: Try stopping and restarting vold. If this fails then a reboot of the system should be attempted, possibly followed by a reconfiguration of LSM. If this fails, contact Customer Support.

23. Message:

lsm:vold: Error: Differing version of vold installed

Clarification: Some inconsistency between vold and the kernel has caused an ioctl to fail. This could be caused by the use of older versions of vold or the kernel, or it could be due to a bug in LSM.

User Action: Try stopping and restarting vold. If this fails then a reboot of the system should be attempted, possibly followed by a reconfiguration of LSM. If this fails, contact Customer Support.

24. Message:

lsm:vold: Error: Disk [*disk_name* , group *group_name* , device *device_name* : \ not updated with new host ID] **Error:** [*error_message*]

Clarification: If the host ID for a system is changed using the voldctl init command then all disks in all imported disk groups will need to have the host ID changed to the new ID. If the host ID for a disk cannot be changed, then this message will be displayed. Other problems might also exist for this disk.

User Action: The contents of the disk should be evacuated elsewhere and the disk should be reinitialized.

25. Message:

lsm:vold: Error: Disk group [*group_name* , Disk *disk_name* : Cannot auto-import group: *error_message*]

Clarification: The disk group *group_name*, could not be reimported after a system restart. The reason is given as part of the error message. Other error messages may appear which provide more information on what went wrong. Any volumes in the disk group will be unavailable until the error condition is fixed and the disk group is reimported.

User Action: Clear the error condition, if possible, and then import the disk group by hand with `voldg import`. After importing, you should restart all volumes with `voldg -g` *groupname* `-sb`.

26. Message:

    lsm:vold: Error: Disk group [*group_name* , Disk *disk_name*] **: Group name collides with** [\ record in rootdg]

    Clarification: The disk group name *group_name*, for the disk group being imported from the named disk, collides with a configuration record in the `rootdg` disk group. Disk groups must have names that do not match any records in the root disk group.

    User Action: If you want to import the disk group, you will have to rename the conflicting record in `rootdg` to some other name.

27. Message:

    lsm:vold: Error: Disk group [*group_name* : Cannot recover temp database] **error_message**

    Clarification: The temp database stored in the root file system could not be opened or read. Other messages will detail the error. This may happen because of an I/O error or a problem in the file system.

    User Action: The system should be rebooted and the operation retried.

28. Message:

    lsm:vold: Error: Disk group [*group_name* : Disabled by errors]

    Clarification: This message can appear if the last configuration database or last kernel log area for a disk group became disabled. This could have been due to an I/O error or some other condition. Other messages preceding this one are likely to highlight the root cause.

    User Action: Any remaining active volumes should be backed up. The disk group will have to be reinitialized and the disks added again to the group to recover.

29. Message:

    lsm:vold: Error: Disk group [*group_name* : Errors in some configuration copies:]

`Clarification:` One or more on-disk database copies were found to contain errors. As a result, the disk group could not be imported. This is probably due to a disk I/O error, or to blocks of a configuration copy being overwritten within invalid contents. Check for messages from the disk driver. Errors pertaining to specific configuration copies are listed on successive lines. These lines can be in either of the following forms:

File [*filename* : *error_message* : Block *number* : *error_message*] **Disk** [*diskname* , copy *copy_number* : *error_message* : Block *number* : *error_message*]

Lines beginning with `File` indicate an error in the special configuration copy file used for storing nonpersistent disk group information. Lines beginning with `Disk` indicate failure of a persistent configuration copy stored on a disk. The copy number indicates which of the disk's configuration copies contains the error.

`User Action:` If one or more disks for the disk group are currently inaccessible (such as due to a cabling error), make the disks accessible and try to import the disk group again with `voldg import`. Otherwise, the disk group is probably no longer usable and will have to be recreated. All volume configuration information for the disk group is lost.

30. `Message:`

    lsm:vold: Error: Disk group [*group_name* : Reimport of disk group failed: *error_message*]

    `Clarification:` The reload of a disk group into the kernel failed. This could be because the log size for the kernel may not be set or because of some other error in the import procedure. Other messages should indicate the true cause of the failure.

    `User Action:` The operation should be retried unless some other error message leads to a suggested course of action. If this fails, the system should be rebooted.

31. `Message:`

    lsm:vold: Error: Disk group [*group_name* : update failed: *error_message*]

    `Clarification:` This message occurs because a database update failed completely. No complete copy of the database could be written for the disk group. The disk group will be disabled and further access for configuration changes will be disallowed. If this error occurs for the root disk group, it will probably be necessary to reinstall the system.

    `User Action:` Any volumes still active in the disk group should be backed up. The disk group will then have to be reinitialized and the disks added again to it.

32. Message:

lsm:vold: Error: Exec of /sbin/voliod failed

Clarification: An exec of /sbin/voliod failed.

User Action: Check the existence and permissions of the /sbin/voliod command. Try executing the command manually to ensure that it can be run.

33. Message:

lsm:vold: Error: Failed to store commit status list into kernel: [*error_message*]

Clarification: Some inconsistency between vold and the kernel has caused an ioctl to fail. This could be caused by the use of older versions of vold or the kernel, or it could be due to a bug in LSM.

User Action: Try stopping and restarting vold. If this fails then a reboot of the system should be attempted, possibly followed by a reconfiguration of LSM. If this fails, contact Customer Support.

34. Message:

lsm:vold: Error: Fork of logio daemon failed

Clarification: The creation of a process that could then be used as a logging daemon failed.

User Action: Check for messages explaining the reason that a fork(2) call failed. Retry the operation.

35. Message:

lsm:vold: Error: GET_VOLINFO ioctl failed: [*error_message*] **lsm:vold: Error: Version number of kernel does not match vold**

Clarification: Some inconsistency between vold and the kernel has caused an ioctl to fail. This could be caused by the use of older versions of vold or the kernel, or it could be due to a bug in LSM.

User Action: Try stopping and restarting vold. If this fails then a reboot of the system should be attempted, possibly followed by a reconfiguration of LSM. If this fails, contact Customer Support.

36. Message:

lsm:vold: Error: Get of current rootdg failed

Clarification: An attempt to retrieve the rootdg from the kernel failed. This might be because of a kernel vold inconsistency or could also be because of a version difference between vold and the kernel.

User Action: Check that the correct version of vold and the kernel are installed. Other messages might suggest other problems in a prior attempt at loading a configuration and possible courses of action. If this fails, contact Customer Support.

37. `Message:`

lsm:vold: Error: No convergence between root disk group and disk list Disks in one version of rootdg: [*disk_name* type= *disk_type* info= *disk_info* Disks in alternate version of rootdg:] [*disk_name* type= *disk_type* info= *disk_info*]

`Clarification:` This message can appear when `vold` is not running in autoconfigure mode (see the `vold`(8) reference page) and when, after several retries, it cannot resolve the set of disks belonging to the root disk group. The algorithm for non-autoconfigure disks is to scan disks listed in the `/etc/vol/volboot` file and then examine the disks to find a database copy for the `rootdg` disk group. The database copy is then read to find the list of disk access records for disks contained in the group. These disks are then examined to ensure that they contain the same database copy. As such, this algorithm expects to gain convergence on the set of disks and the database copies contained on them. If a loop is entered and convergence cannot be reached, then this message will appear and the root disk group importation will fail.

`User Action:` Reorganizing the physical locations of the devices attached to the system may break the deadlock. If this fails, contact Customer Support.

38. `Message:`

lsm:vold: Error: Open of directory [*directory_path* failed]

`Clarification:` When `vold` was trying to create node files for the volumes, it was unable to open the directory in which the nodes were to be created.

`User Action:` Check for other errors that suggest why the directory might be missing or if the permissions might be incorrect. Fix the condition to allow `vold` to open or create the directory, then issue the command `voldctl enable`.

39. `Message:`

lsm:vold: Error: Read of directory [*directory_path* failed]

`Clarification:` The node directory could not be read when `vold` was trying to scan for volume nodes.

`User Action:` Check for other messages that might suggest why the directory is inaccessible. Try reading the directory manually if the directory is corrupted, then try removing and recreating it and then restarting `vold`.

40. `Message:`

lsm:vold: Error: Unexpected configuration tid for group [*group_name* found in kernel]

`Clarification:` Some inconsistency between `vold` and the kernel has caused an `ioctl` to fail. This could be caused by the use of older versions of `vold` or the kernel, or it could be due to a bug in LSM.

`User Action:` Try stopping and restarting `vold`. If this fails then a reboot of the system should be attempted, possibly followed by a reconfiguration of LSM. If this fails, contact Customer Support.

41. `Message:`

    lsm:vold: Error: Unexpected error during [*usage_type* volume reconfiguration: *error_message*]

    `Clarification:` A record lock for the volume could not be acquired as part of the initial volume setup for either a root or swap volume. This is most likely to occur under low memory conditions.

    `User Action:` Other messages may suggest an alternate course of action. Otherwise, this is generally an unrecoverable error and will require either the boot of an alternate root device or reloading of the system from backups.

42. `Message:`

    lsm:vold: Error: Unexpected error fetching disk for [*usage_type* volume: *error_message*]

    `Clarification:` Some inconsistency between `vold` and the kernel has caused an `ioctl` to fail. This could be caused by the use of older versions of `vold` or the kernel, or it could be due to a bug in LSM.

    `User Action:` Try stopping and restarting `vold`. If this fails then a reboot of the system should be attempted, possibly followed by a reconfiguration of LSM. If this fails, contact Customer Support.

43. `Message:`

    lsm:vold: Error: Unexpected values stored in the kernel

    `Clarification:` Some inconsistency between `vold` and the kernel has caused an `ioctl` to fail. This could be caused by the use of older versions of `vold` or the kernel, or it could be due to a bug in LSM.

    `User Action:` Try stopping and restarting `vold`. If this fails then a reboot of the system should be attempted, possibly followed by a reconfiguration of LSM. If this fails, contact Customer Support.

44. `Message:`

    lsm:vold: Error: VOL_RESET_KERNEL failed: a volume or plex device is open

or

lsm:vold: Error: VOL_RESET_KERNEL failed: [*error_message*]

`Clarification`: An attempt at resetting the kernel state with a `vold -r reset` command failed because all the LSM objects in the kernel were not closed. If any volumes are in use, then the reset cannot be performed. This may also happen if a reset was requested on a system with root volumes. Root volumes are, by definition, never closed and so a reset cannot be performed.

`User Action`: If a reset is really desired, then checking the state of the volumes and any mounted file systems should result in information about who might have them open. Unmounting all volumes and killing any processes accessing the volumes should allow the reset to occur.

45. `Message`:

    lsm:vold: Error: [*mode* : Unrecognized operating mode]

    `Clarification`: An unknown mode string was entered following a `-m` option.

    `User Action`: Select a valid mode from the `vold`(8) reference page and try again.

46. `Message`:

    lsm:vold: Error: cannot open /dev/voliod: [*error_message*]

    `Clarification`: The open of the `/dev/voliod` file can only fail if the device node is missing or has an incorrect major or minor number.

    `User Action`: Check the existence and values of the file and make sure that the LSM software was correctly installed.

47. `Message`:

    lsm:vold: Error: cannot open [*argument* : *error_message*]

    `Clarification`: The tracefile specified on the command line could not be opened in append mode. The error message supplied should explain the reason.

    `User Action`: Select an alternate tracefile name that can be created or appended to.

48. `Message`:

    lsm:vold: Error: cannot open [*volconfig_device* : Device is already open]

    or

    lsm:vold: Error: cannot open [*volconfig_device* : *error_message*]

Clarification The exclusive open device (/dev/volconfig) is already open. Only one vold process can be active on the system at one time. Subsequent attempts at starting vold or opening the device will result in this message.

User Action: Check for other running vold processes. The voldctl mode will report if vold is currently active.

49. Message:

    lsm:vold: Error: enable failed: [*error_message*]

    Clarification: This message may occur during an initial startup of vold. If changing to enabled mode when this error occurs, failures could be due to problems with the creation of the portal or with connection to the kernel. If changing from an enabled state to a disabled state, then problems could occur with removing the disk groups from the kernel because of such things as volumes in use.

    User Action: Evaluate other error messages occurring with this one to determine the root cause of the problem. Make changes suggested by the other errors and then retry the command.

50. Message:

    lsm:vold: Error: failed to create daemon: fork failed: [*error_message*]

    Clarification: The call to fork(2) to generate a background vold process failed.

    User Action: Check for messages explaining the reason that a fork(2) call failed. Retry the operation.

51. Message:

    lsm:vold: Error: volume [*volume_name* : Wait for logging daemon failed]

    Clarification: The wait called to wait for the existence of the daemon process did not execute correctly. This can only happen if the ioctl does not correctly match the command required, perhaps because of a mismatch between the voliod command and the kernel versions or perhaps because of an incorrect minor number for the /dev/voliod device.

    User Action: Check the existence and permissions of the /dev/voliod device.

52. Message:

    lsm:vold: FATAL Error: Disk group rootdg: Inconsistency – Not loaded into kernel

    Clarification: Some inconsistency between vold and the kernel has caused an ioctl to fail. This could be caused by the use of older versions of vold or the kernel, or it could be due to a bug in LSM.

User Action: Try stopping and restarting vold. If this fails then a reboot of the system should be attempted, possibly followed by a reconfiguration of LSM. If this fails, contact Customer Support.

53. Message:

lsm:vold: FATAL Error: Group [*group_name* : Cannot update kernel]

Clarification: Some inconsistency between vold and the kernel has caused an ioctl to fail. This could be caused by the use of older versions of vold or the kernel, or it could be due to a bug in LSM.

User Action: Try stopping and restarting vold. If this fails then a reboot of the system should be attempted, possibly followed by a reconfiguration of LSM. If this fails, contact Customer Support.

54. Message:

lsm:vold: FATAL Error: Interprocess communication failure: [*error_message*]

Clarification: The portal to client utilities has returned a failure. This is a fatal error because without a portal to clients, vold cannot do anything useful.

User Action: Check for other errors suggesting the reason for portal failure. Restart vold. If problems persist, reboot the system.

55. Message:

lsm:vold: FATAL Error: Invalid status stored in kernel

Clarification: Some inconsistency between vold and the kernel has caused an ioctl to fail. This could be caused by the use of older versions of vold or the kernel, or it could be due to a bug in LSM.

User Action: Try stopping and restarting vold. If this fails then a reboot of the system should be attempted, possibly followed by a reconfiguration of LSM. If this fails, contact Customer Support.

56. Message:

lsm:vold: Warning: Cannot create device [*path* : *error_message*]

Clarification: The mknod(2) call made by vold to create a device node failed. The reason for the error should be displayed.

User Action: Fix the reason indicated for node creation failure and then issue the command voldctl enable.

57. Message:

lsm:vold: Warning: Cannot exec /sbin/rm to remove [*directory_path* : *error_message*]

Clarification: An exec of /sbin/rm failed.

User Action: Ignore the error. It is not serious if the directory could not be removed.

58. Message:

    lsm:vold: Warning: Cannot fork to remove directory [*directory_path* : *error_message*]

    Clarification: The call to fork(1) to generate a process that could then exec rm(2) failed.

    User Action: Ignore the error. It is not serious if the directory could not be removed.

59. Message:

    lsm:vold: Warning: Disk [*device_name* in kernel not a recognized type]

    Clarification: The disk type of a disk in the kernel does not match any known disk type. This can only occur if vold and the kernel are in an inconsistent state.

    User Action: Try stopping and restarting vold. If this fails then reconfigure LSM. If this fails, contact Customer Support.

60. Message:

    lsm:vold: Warning: Disk [*disk_name* names group *group_name* , but group ID differs]

    Clarification: As part of a disk group import, a disk was discovered that had a mismatched disk group name and disk group ID. This disk will not have been imported. This can only happen if two disk groups of the same name exist that have different disk group ID values. In that case, one group will be imported along with all its disks and the other group will not. This message will appear for disks in the un-selected group.

    User Action: If it turns out that the disk should be imported into the group, then this will have to be done by adding the disk to the group at a later stage. It will not happen automatically as part of the import. All configuration information for the disk will also be lost.

61. Message:

    lsm:vold: Warning: Disk group [*group_name* is disabled, disks not updated with \ new host ID]

    Clarification: If the host ID for a system is changed using the voldctl init command then all disks in all imported disk groups will need to have the host ID changed to the new ID. If a disk group is found in the imported but disabled state, then the host ID will not be changed.

`User Action`: The host ID will need to be cleared using the `voldisk clearimport` command for each disk, and then the disk group should be reimported.

62. `Message`:

    lsm:vold: Warning: Disk group [*group_name* : Disk group log may be too small] **Log size should be at least** [*number* blocks]

    `Clarification`: The log areas for the disk group have become too small for the size of configuration currently in the group. This should usually never happen without first displaying a message about the database area size. This message only occurs during disk group import; it occurs if the disk was inaccessible while new database objects were added to the configuration, and the disk was then made accessible and the system restarted.

    `User Action`: If this situation does occur, then the disks in the group will have to be explicitly reinitialized with larger log areas. See the `voldisk`(8) reference page for more information. To reinitialize all the disks, they must be detached from the group with which they are associated and then reinitialized and readded. The group should then be deported and reimported for the changes to the log areas for the group to take effect.

63. `Message`:

    lsm:vold: Warning: Disk group [*group_name* : Errors in some configuration copies:]

    `Clarification`: One or more on-disk database copies were found to contain errors. As a result, the disk group could not be imported. This is most likely to be due to a disk I/O error, or to blocks of a configuration copy being overwritten within invalid contents. Check for messages from the disk driver. Providing that other copies of the database can be successfully read, the system will continue and the disk group import or initial `vold` enable operation should succeed. If the database copy can subsequently be written to, then this message will not recur. Errors pertaining to specific configuration copies are listed on successive lines. These lines can be in either of the following forms:

    File [*filename* : *error_message* : Block *number* : *error_message*] **Disk** [*diskname* , copy *copy_number* : *error_message* : Block *number* : *error_message*]

    Lines beginning with `File` indicate an error in the special configuration copy file used for storing nonpersistent disk group information. Lines beginning with `Disk` indicate failure of a persistent configuration copy stored on a disk. The copy number indicates which of the disk's configuration copies contains the error.

User Action: This message is likely to occur once due to an I/O failure and then not reoccur. If it does reoccur, then it may be necessary to remove the disk and reinitialize it to clear the condition. If all configuration copies for a disk group become unusable, then the disk group itself becomes unusable and must be recreated. If the `rootdg` disk group becomes unusable, LSM may need to be reconfigured. In this case, if root file system is on a volume, then the operating system itself may need to be reinstalled.

64. Message:

    lsm:vold: Warning: Error in volboot file: [*error_message*] **Entry: disk** [*disk_name disk_type disk_info*]

    Clarification: This message occurs when an entry in the `volboot` file does not contain the correct information to define a valid disk access record.

    User Action: Remove the entry using the `voldctl rmdisk` command and add it again using `voldctl adddisk`.

65. Message:

    lsm:vold: Warning: Failed to update voldinfo area in kernel: [*error_message*]

    Clarification: Some inconsistency between `vold` and the kernel has caused an ioctl to fail. This could be caused by the use of older versions of `vold` or the kernel, or it could be due to a bug in LSM.

    User Action: Try stopping and restarting `vold`. If this fails then a reboot of the system should be attempted, possibly followed by a reconfiguration of LSM. If this fails, contact Customer Support.

66. Message:

    lsm:vold: Warning: Field too long in volboot file: Entry: disk [*disk_name disk_type disk_info*]

    Clarification: The `volboot` file is maintained by `vold` and `voldctl` and should never normally exhibit this problem. This problem might indicate some corruption of the `volboot` file or could also be the result of manual editing of the file.

    User Action: The offending entry could try to be removed by use of the `voldctl rmdisk` command. If this fails, `volboot` may have to be reinitialized using a `voldctl init` command.

67. Message:

lsm:vold: Warning: Get of record [*record_name* from kernel failed: *error_message*]

Clarification: Some inconsistency between vold and the kernel has caused an ioctl to fail. This could be caused by the use of older versions of vold or the kernel, or it could be due to a bug in LSM.

User Action: Try stopping and restarting vold. If this fails then a reboot of the system should be attempted, possibly followed by a reconfiguration of LSM. If this fails, contact Customer Support.

68. Message:

lsm:vold: Warning: Plex [*plex_name* for *usage_type* volume is stale or unusable]

Clarification: This message is output to alert the user to the failure of one or more plexes of either the root or swap volume. The system may be able to continue depending on the existence of other usable plexes for the volume.

User Action: The failed plex should be repaired by either reattaching the plex to the volume once the system is booted, or by evacuating and replacing the disk on which the failed plex resides if it is thought that the disk is going bad.

69. Message:

lsm:vold: Warning: cannot remove group [*group_id* from kernel: *error_message*]

Clarification: Some inconsistency between vold and the kernel has caused an ioctl to fail. This could be caused by the use of older versions of vold or the kernel, or it could be due to a bug in LSM.

User Action: Try stopping and restarting vold. If this fails then a reboot of the system should be attempted, possibly followed by a reconfiguration of the LSM package. If this fails, contact Customer Support.

70. Message:

lsm:vold: Warning: response to client [*client number* failed: *error_message*]

Clarification: The portal to client utilities has returned a failure. This is a fatal error because without a portal to clients, vold cannot do anything useful. This could be caused by a STREAMS error or some other communications problem with the client.

User Action: Check for other errors suggesting the reason for portal failure. Restart vold. If problems persist, reboot the system.

E.2 Kernel Error Messages

The following are the kernel level error messages.

1. `Message:`

   NOTICE: [*message* on volume device *hex_device_number* (*volume_name*)in diskgroup *group_name*]

   `Clarification:` This is caused by a driver above the LSM level calling the LSM `volprint`() function. This usually happens when a driver detects some error condition in LSM and want to display the error.

   `User Action:` No action necessary, unless specified in a supplied string.

2. `Message:`

   NOTICE: io/vol.c(volerror): Correctable [*type* error on volume *volume_name* ,e] **plex** [*plex_name* , block *block_number*]

   `Clarification:` A correctable I/O error was detected and corrected. A correctable I/O error is one where a read error from an underlying device driver could be corrected by reading the data from an alternate mirror copy and then writing it back to the failed mirror.

   `User Action:` If the I/O could have been completed by reading from an alternate mirror but the writeback to the failed mirror still failed, the mirror will be detached. This failure will cause the exception handling code to be entered, which will result in the volume's error recovery policy being followed. This usually results in either a mirror or the volume becoming detached. The user must intervene to reattach the mirror (`volplex att`), to bring back the failed mirror copy. If the volume was detached, then the data contained on it is unrecoverable and will have to be restored from backups.

3. `Message:`

   NOTICE: io/vol.c(volerror): Uncorrectable [*type* error on volume *volume_name* , \ plex *plex_name* , block *block_number*]

   `Clarification:` Following an I/O error from one mirror, an attempt to reread the data from an alternate mirror failed. This could be because no other mirrors exist or could be because the other mirrors also had I/O failures.

   `User Action:` This failure will cause the exception handling code to be entered, which will result in the volume's error recovery policy being followed. This can have effects ranging from detaching a mirror to disabling the volume. The user must intervene to reattach the mirror (`volplex att`), to bring back the failed mirror copy. If the volume was detached, then the data contained on it is unrecoverable and will have to be restored from backups.

4. `Message:`

   NOTICE: lsm: Can't close disk [*disk_name* in group *disk_group* .] **If it is removable media (like a floppy), it may have been removed.** [Otherwise, there may be problems with the drive. Kernel error code *error_number/error_number*]

   `Clarification:` This is unlikely to happen; closes cannot fail.

   `User Action:` None.

5. `Message:`

   NOTICE: lsm: Can't open device [*disk_name* , device busy or inaccessible.]

   `Clarification:` The named disk cannot be accessed.

   `User Action:` Turn on the drive.

6. `Message:`

   WARNING: io/vol.c(volexcept): No volume error daemon - Cannot Log plex detach, [\ detaching volume]

   `Clarification:` No `voliod` process was running and able to log a detach record for a mirror that is being detached due to an I/O error. This is a fatal error that causes future access to the volume to be rejected, because any system failure coming after additional I/O would not be able to detect the failure of the mirror and mirror inconsistencies might then occur.

   `User Action:` Although it is too late to rescue this volume, at least one `voliod` process should be started as soon as possible (using `voliod set 2`). The failed volume will have to be stopped and restarted, then reloaded from backups. Mirrors will have become inconsistent and so any attempt at using the data on the volume could prove disastrous.

7. `Message:`

   WARNING: volklog_dgfree: Can't clear group commit log record for group [*disk_group*]

   `Clarification:` This can occur if a log flush to disk could not be performed because no valid log copies remained. This is likely to compromise the ability of the LSM to recover from any further I/O errors.

User Action: Disks should be added to the system such that new viable logging areas can be generated. Alternatively, failed disks should be removed and replaced with working devices.

8. Message:

   WARNING: volklog_dgfree: Can't free kernel logging area for vol_reset_kernel of group [*disk_group*]

   Clarification: A free of the logs for a disk group failed because either no valid log areas remained for flushing or some log records remained in the log before the clear operation was requested.

   User Action: No user action can be taken here; this is a LSM internal error. Contact Customer Support.

Glossary

The following are LSM terms and definitions.

concatenated plex

A plex whose subdisks are associated at specific offsets within the address range of the plex, and extend in the plex address range for the length of the subdisk. This layout allows regions of one or more disks to create a plex, rather then a single big region.

description set

A set of files that are saved using the `volsave`(8) command and can be used to restore an LSM configuration. By default, an LSM description set is saved in a timestamped directory under the `/usr/var/lsm/db` directory.

disk

Disks exist as two entities:

1. A physical disk on which all data is ultimately stored and which exhibits all the behaviors of the underlying technology.
2. An LSM representation of disks which, while mapping one-to-one with the physical disks, are just presentations 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 while the LSM disk is simply a unit of allocation with a name and a size.

disk access record

A configuration record that defines the path to a disk. Disk access records most often name 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 allows you to move disks between controllers, or to different locations on a controller. When you move a disk, a different disk access record 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 all devices attached to the system. On these systems, it is not necessary 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.

Disk access records are identified by their disk access names (also known as DA names).

disk group

A group of disks that share a common configuration. A configuration consists of a set of records describing objects including disks, volumes, plexes, and subdisks that are associated with one particular disk group. Each disk group has an administrator-assigned name that you use to to reference that disk group. Each disk group has an internally defined unique disk group ID, which is used to differentiate 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 configurations are stored on disks. All disks in a disk group are stamped with a disk group ID, which is a unique identifier for naming disk groups. Some or all disks in a disk group also store copies of the configuration of the disk group.

disk group configuration

A disk group configuration is a small database that contains all volume, plex, subdisk, and disk media records. These configurations are replicated onto some or all disks in the disk group, often with two copies on each disk. Because these databases are stored on disk groups, 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.

disk group ID

A 64 byte universally unique identifier that is assigned to a disk group when the disk group is created with `/sbin/voldg init` command. This identifier is in addition to the disk group name, which you assigned. The disk group ID is used to check for disk groups that have the same administrator-assigned name but are actually different.

disk group records

Disk group records define several different types of names for a disk group. The different types of names are as follows:

- The *real name* is the name of the disk group, as the name is defined on disk. This name is stored in the disk group configuration, and is also stored in the disk headers of all disks in the disk group.
- The *alias name* is the standard name that the system uses when referencing the disk group. References to the disk group name usually mean the alias name. Volume and plex device directories are structured into subdirectories based on the disk group alias name. Typically, the disk group's alias name and real name are identical. A local alias is useful for gaining access to a disk group with a name that conflicts with other disk groups in the system, or that conflicts with records in the `rootdg` disk group.
- The *disk group ID* is a 64-byte identifier that represents the unique ID of the disk group. All disk groups on all systems should have a different disk group ID, even if they have the same real name. This identifier is stored in the disk headers of all disks in the disk group. It is used to ensure that the LSM software does not confuse two disk groups which were created with the same name.

disk header

A block stored in a private region of a disk and that defines several properties of the disk. The disk header defines the:

- 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

disk ID

A 64 byte universally unique identifier that is 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.

disk media record

A reference to a physical disk, or possibly a disk partition. This record can be thought of as a physical disk identifier for the disk or partition. Disk media records are configuration records that provide a name (known as the disk media name or DM name) that you use to reference a particular disk independent of its location on the system's various disk controllers. Disk media records reference particular physical disks through a disk ID, which is a unique identifier that is assigned to a disk when it is initialized for use with the LSM software.

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, with any associated subdisks being removed or replaced along with the disk.

host ID

A name, usually assigned by you, that identifies a particular host. 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's host ID. If another system attempts to access the disk, it detects that the disk has a non-matching host ID and disallows access until the first system discontinues use of the disk. Use the `/sbin/voldisk clearimport` command to clear the host ID stored on a disk for system failures that do not clear the host ID,

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.

kernel log

A log kept in the private region on the disk and that is written by 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 is guaranteed to detect the state changes even in the event of a system failure.

plex

A copy of a volume's logical data address space, also sometimes known as a *mirror*. A volume can have up to eight plexes associated with it. Each plex is, at least conceptually, a copy of the volume that is maintained consistently in the presence of volume I/O and reconfigurations. Plexes represent the primary means of configuring storage for a volume. Plexes can have a striped or concatenated organization (layout).

plex consistency

If the plexes of a volume contain different data, then the plexes are said to be inconsistent. This is only a problem if LSM is unaware of the inconsistencies, as the volume can return differing results for consecutive reads.

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, if parts of the write complete on one plex, but not the other. If the plexes are not first synchronized to contain the same data, plexes are inconsistent after creation of a mirrored volume. An important part of LSM is to ensure that consistent data is returned to any application that reads a volume. This may 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 such that reads from one plex are automatically written back to the other plexes, making the data consistent for that volume offset.

private region

Disks used by LSM contain two special regions: a private region and a public region. Usually, each region is formed from a complete partition of the disk; however, the private and public regions can be allocated from the same partition.

The private region of a disk contains various on-disk structures that are used by LSM for various internal purposes. Each private region 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, and copies of the disk group's kernel log.

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 particular disk. Only one contiguous region of disk can form the public region for a particular disk.

read policy

A configurable policy for switching between plexes for volume reads. 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. Read policy choices include:

- round-robin

  For every other read operation, switch to a different plex from the previous read operation. Given three plexes, this will switch between each of the three plexes, in order.

- preferred plex

  This 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 round-robin.

- select

  This read policy is the default policy, and adjusts to use an appropriate read policy based on the set of plexes associated with the volume. If exactly 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 non-striped plex, preferring the striped plex often yields better throughput.

root disk group

Each system requires one special disk group, called `rootdg`. This group is generally the default for most utilities. In addition to defining the regular disk group information, the configuration for the root disk group contains local information that is specific to a disk group and that is not intended to be movable between systems.

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 much 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.

subdisk

A region of storage allocated on a disk for use by a volume. Subdisks are associated to volumes through plexes. You organize one or more subdisks to form plexes based on a plex layout: striped or concatenated. Subdisks are defined relative to disk media records.

`volboot` file

The `volboot` file is a special file (usually stored in `/etc/vol/volboot`) that is used to bootstrap the root disk group 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 disks 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 is read and is used to get a more complete list of disk access records that are used as a second-stage bootstrap of the root disk group, and to locate all other disk groups.

volume

A virtual disk device that looks to applications and file systems like a regular disk partition device. Volumes present block and raw device interfaces that are compatible in their use. A volume can be mirrored, spanned across disk drives, moved to use different storage, and striped. You can change the configuration of a volume without causing disruption to applications or file systems that are using the volume.

volume records

Volume records define the characteristics of particular volume devices. The name of a volume record defines the node name used for files in the `/dev/vol` and `/dev/rvol` directories. The block device for a particular volume, which can be used as an argument to the mount command has the path /dev/vol/*groupname/volume*.

See the mount(8) reference page for more information on the mount command.

Index

T

U

V

How to Order Tru64 UNIX Documentation

You can order documentation for the Tru64 UNIX operating system and related products at the following Web site:

http://www.businesslink.digital.com/

If you need help deciding which documentation best meets your needs, see the Tru64 UNIX *Documentation Overview* or call **800-344-4825** in the United States and Canada. In Puerto Rico, call **787-781-0505**. In other countries, contact your local Compaq subsidiary.

If you have access to Compaq's intranet, you can place an order at the following Web site:

http://asmorder.nqo.dec.com/

The following table provides the order numbers for the Tru64 UNIX operating system documentation kits. For additional information about ordering this and related documentation, see the *Documentation Overview* or contact Compaq.

Name	Order Number
Tru64 UNIX Documentation CD-ROM	QA-6ADAA-G8
Tru64 UNIX Documentation Kit	QA-6ADAA-GZ
End User Documentation Kit	QA-6ADAB-GZ
Startup Documentation Kit	QA-6ADAC-GZ
General User Documentation Kit	QA-6ADAD-GZ
System and Network Management Documentation Kit	QA-6ADAE-GZ
Developer's Documentation Kit	QA-6ADAG-GZ
Reference Pages Documentation Kit	QA-6ADAF-GZ

Reader’s Comments

Tru64 UNIX
Logical Storage Manager
AA-RH9BA-TE

Compaq welcomes your comments and suggestions on this manual. Your input will help us to write documentation that meets your needs. Please send your suggestions using one of the following methods:

- This postage-paid form
- Internet electronic mail: `readers_comment@zk3.dec.com`
- Fax: (603) 884-0120, Attn: UBPG Publications, ZKO3-3/Y32

If you are not using this form, please be sure you include the name of the document, the page number, and the product name and version.

Please rate this manual:

	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	☐	☐	☐	☐
Clarity (easy to understand)	☐	☐	☐	☐
Organization (structure of subject matter)	☐	☐	☐	☐
Figures (useful)	☐	☐	☐	☐
Examples (useful)	☐	☐	☐	☐
Index (ability to find topic)	☐	☐	☐	☐
Usability (ability to access information quickly)	☐	☐	☐	☐

Please list errors you have found in this manual:

Page Description

________ ____
________ ____
________ ____
________ ____

Additional comments or suggestions to improve this manual:

____
____
____
____

What version of the software described by this manual are you using? ____________

Name, title, department ____________________________
Mailing address ____________________________
Electronic mail ____________________________
Telephone ____________________________
Date ____________________________

Do Not Cut or Tear - Fold Here and Tape

COMPAQ

NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS MAIL PERMIT NO. 33 MAYNARD MA

POSTAGE WILL BE PAID BY ADDRESSEE

COMPAQ COMPUTER CORPORATION
UBPG PUBLICATIONS MANAGER
ZKO3-3/Y32
110 SPIT BROOK RD
NASHUA NH 03062-9987

Do Not Cut or Tear - Fold Here

Cut on Dotted Line