Logical Storage Manager

Release Notes

Order Number: AA-Q3NHA-TE

April 1994

Product Version: Logical Storage Manager
Version 1.0

Operating System and Version:
DEC OSF/1 Version 2.0

This document includes release notes for the DEC OSF/1 Logical Storage Manager Version 1.0. These notes describe supported features and products and known problems and fixes.

digital equipment corporation
Maynard, Massachusetts

This document contains release notes for the Logical Storage Manager Version 1.0.

1.1 Accessing LSM Volumes

You can create an LSM volume using the LSM command interface or the visual administrator graphical interface tool, called dxlsm. LSM volumes can be accessed the same way a disk device is accessed. The block-special files for LSM volumes are located in the /dev/vol directory. The character-special files are located in the /dev/rvol directory.

To create and mount a new file system on an LSM volume, use the newfs command with the disk type argument specifying any known disk type. LSM uses the disk type to obtain the sector and track size information for newfs. For example, to create a new UFS file system on LSM volume vol01, do the following:

```
# newfs /dev/rvol/rootdg/vol01 rz26
# mount /dev/vol/rootdg/vol01 /mnt
```

1.2 Requirements

The following notes describe the hardware and software requirements, and configuration limitations for LSM.

1.2.1 Hardware Requirements

LSM does not depend on specific hardware in order to operate. All functions can be performed on any supported AXP computer running DEC OSF/1 Version 2.0 or later. There are no restrictions on the devices supported beyond the valid configurations defined in the DEC OSF/1 Version 2.0 Software Product Descriptions (SPD 41.51).

All SCSI and DSA disks supported by DEC OSF/1 Version 2.0 are supported by LSM. Redundant arrays of independent disks (RAID) hardware devices are supported as standard disks, with each RAID device-logical unit viewed as a physical disk.

In order to help the user configure logical volumes properly, LSM returns warning messages if the user attempts to mirror or stripe data on the same physical device.

1.2.2 Software Requirements

LSM requires that the system run DEC OSF/1 software Version 2.0 or later.

The visual administrator is a Motif-based application that requires you have Motif installed on the system.

1.2.3 Configuration Limitations

The maximum configuration supported by the DEC OSF/1 Logical Storage Manager is defined as follows:

- Maximum of 250 volumes

- Maximum of 8 plexes per volume

  A plex is a copy, also referred to as a mirror, of data. Volumes that are not mirrored have one plex, while mirrored volumes can have anywhere from two to eight plexes.

- Maximum of 256 subdisks per volume
  A subdisk is a basic unit of disk space allocation for logical volumes.

- Maximum volume size of 128 Gbytes

- Maximum of 128 disks per system

  Disks are organized into disk groups. A disk group is a named collection of disks that share the same management database. Volumes are created within a disk group and are restricted to using disks within that disk group (for example, logical volumes cannot span disk groups).

1.3 Starting LSM

You can start LSM at machine boot time prior to the execution of bcheckrc so that the fsck command can automatically analyze and mount file systems on LSM volumes. The /sbin/volinstall utility inserts an entry in the /etc/inittab file before the bcheckrc entry to start LSM at machine boot time. The entry in the /etc/inittab file executes the /sbin/lsmbstartup script to start LSM.

1.4 LSM Visual Administrator

The LSM visual administrator (dxlsm) interface is the easiest way to start using LSM. After LSM is started, use the /usr/bin/X11/dxlsm command to invoke the visual administrator.

The dxlsm interface changes the shape of the mouse pointer to an X. To change the mouse pointer, you need to explicitly reset its shape by picking the "Pointer" option from the Session Manager Customize menu.

Note: The Visual Administrator Demo Mode is not supported.

1.5 LVM Interface Retirement

DEC OSF/1 Version 1.2 (released in March 1993) included the DEC OSF/1 Logical Volume Manager (LVM). LVM expands and enhances the standard UNIX system mechanism for data storage, retrieval, and protection.

Because LSM provides concatenation, data mirroring, data striping, and a comprehensive online data storage management interface in a single integrated component, Digital plans to retire LVM in a future release of the DEC OSF/1 system.

To help you prepare for the LVM retirement, migration tools are provided with LSM to enable migration by encapsulating current LVM volumes and UNIX partition devices into LSM volumes.

1.6 LSM Encapsulation Tools

LSM provides a set of tools that you can use to migrate existing user data into LSM volumes, without physically moving the data. This process is referred to as encapsulation. The following tools are provided to help you encapsulate user data on UNIX style-partitions, LVM volume group, or ADVfs storage domains. The list briefly describes the capabilities of these tools. For complete encapsulation information, refer to the DEC OSF/1 Logical Storage Manager System Administrator's Guide.

Note

> You should back up all the data you plan to encapsulate before performing the encapsulation.

- /usr/sbin/volencap

  This is a UNIX disk (partitioned device) encapsulation tool. This program encapsulates complete disks or specific partitions on a UNIX disk into LSM logical volumes. When encapsulating a disk, each partition that is in use is converted to an LSM volume. A partition on a disk is determined to be in use when it has an entry in the /etc/fstab file or the partition has a disklabel tag other than "unusued." Applications that access user data directly using the device interface usually do not set the disklabel tags for the partition in use. Any partition being used in this way should run the disklabel utility to change the disklabel tag for the partition. The volencap utility automatically changes the /etc/fstab file entries for any partition that has an entry in the /etc/fstab file.

- /usr/sbin/vollvmencap
  This encapsulation tool converts all LVM volumes in a volume group (including volumes that span multiple disks) into LSM volumes. The conversion of LVM volumes is at the LVM volume group granularity such that all LVM volumes in a volume group are converted to LSM volumes. Any /etc/fstab file entries associated with those LVM volumes are converted to LSM volume names.

- /usr/sbin/voladvdomencap

  This encapsulation tool converts the storage devices under an ADVfs domain into LSM volumes. The domain is redefined to consist of LSM volumes.

These encapsulation tools examine the UNIX device, LVM volume group, or ADVfs domain the user specifies as input, and generate files containing instructions that actually implement the encapsulation changes.

To make the encapsulation changes take effect, execute the following commands:

- /sbin/vol-reconfig

  This command executes the instruction files generated by the volencap and voladvdomencap tools.

- /sbin/vol-lvm-reconfig

  This command executes the instruction files generated by the vollvmencap tool.

You can enter the vol-reconfig command at the command line prompt, or you can insert the command in the /etc/inittab file to process all the disk and ADVfs encapsulation instructions at machine reboot time. An example of an entry in the /etc/inittab file to encapsulate LVM volumes is as follows:

```
lsm:23:wait/sbin/lsmbstartup</dev/console>/dev/console 2>&1
encap:23:wait:/sbin/vol-reconfig</dev/console>/dev/console 2>&1
fs:23:wait:/sbin/bcheckrc</dev/console>/dev/console 2>&1
```

Note

> You must execute the vol-lvm-reconfig command at the command line prompt only. Do not insert the vol-lvm-reconfig command in the /etc/inittab file.

1.7 Problems Writing to Physical Disk Block 0

Because physical block 0 on Digital disks is write-protected by default, under some conditions, a device that has been encapsulated as an LSM disk can incur write failures unpredictably when physical disk block 0 is accessed for write operations.

This problem occurs when a disk that was previously added to LSM by one of the encapsulation methods, is reused for another application or purpose. When an LSM volume that has user data in block 0 is dissolved and its disk space is reused for a new purpose, neither the new application nor LSM know that a read-only physical disk block 0 is in the newly created LSM volume. When the application or LSM tries to write to the read-only physical disk block 0, a write failure can occur.

To fix this problem, you must remove the read-only physical disk block 0 from the LSM disk before it can be assigned to the new volume.

To remove block 0 from an LSM disk, perform these steps:

1. Remove the LSM disk using the voldg command.

2. If the LSM disk covers the entire disk, use the voldiskadd command to create the new LSM disk.

Note

This problem does not occur under the following conditions:

- When the volencap and voldiskadd utilities are used to encapsulate new disks. In this case, the encapsulation process does not include block 0 in the new LSM disk, thus, a write failure cannot occur.

  When volencap or voldiskadd are used to create a simple LSM disk on partition a or partition c of a disk, the starting offset of the partition is changed to 1, avoiding block 0. However, when the offset for partition a has been changed, then the disklabel utility fails to report disklabel information for that disk.

- When encapsulating a disk that has existing user data on a partition, block 0 is preserved so that the existing file system can continue to be accessed. As long as the volume is used for the same application, write failures will not occur.

1.8 Disk Device Naming Conventions

LSM Version 1.0 supports disk devices that are named in the format: ccN[p], where:

- cc is a two-character code designating the device type

- N is the device unit number

- [p] is the optional partition letter

For example, rz in the device name rz9 represents a pseudonym for SCSI disks. Similarly, ra in the device name ra9 represents a pseudonym for DSA disks.

In the case of a simple disk or nopriv disk, a partition letter must be specified. For example, rz9g. In the case of a sliced disk, a partition letter must not be specified. For example, rz9.

1.9 Real-Time Support

The LSM distribution kit includes LSM real-time objects that enable you to build LSM into a real-time kernel. If you have real-time installed on your system and you want to enable real-time support, you do so when installing LSM. During the installation of LSMBIN100, you are asked whether you want to build a real-time kernel or a regular kernel. If you want to change the kernel type at a later time, you can do so by changing the path of the LSM entry in the file /usr/sys/conf/.product.list to point to the location of the LSM kernel object.

The regular kernel objects are located in: /usr/opt/LSM100/sys/BINARY.

The real-time kernel objects are located in: /usr/opt/LSM100/sys/BINART.rt.

1.10 Layered Product Support

The following sections describe layered products that are newly supported by the LSM software.

1.10.1 Availability Server Environment (ASE)

The Availability Server Environment (ASE) failover capability is supported for LSM in the ASE kit. You can use ASE services to fail over LSM logical volumes among ASE hosts to provide highly available data storage.

1.10.2 Prestoserve Support

Prestoserve is supported with LSM.

1.10.3 RAID Support

Hardware RAID devices are supported as standard disks. LSM views each RAID device logical unit as a physical disk.

1.11 Problems Fixed in LSM

The following section describes a known LSM software problem that has been fixed.

1.11.1 Dxlsm Advanced Operations

Dxlsm advanced operations that use the length field (Advanced Operations in the dxlsm pull-down menu to create volumes, plexes, or subdisks) now work properly.

1.12 Documentation

The LSM documentation provides a complete introduction to the LSM storage concepts and reference information for managing LSM volume. Besides these release notes, the LSM documentation set includes the following manuals in both hardcopy and Bookreader formats:

- DEC OSF/1 Logical Storage Manager Install Guide
- DEC OSF/1 Logical Storage Manager System Administration Guide
- DEC OSF/1 Logical Storage Manager Basic Us Guide
- DEC OSF/1 Logical Storage Manager Visual Administration User's Guide
- DEC OSF/1 Logical Storage Manager Visual Administration Reference Guide

In addition, Reference Pages are provided. Enter the man command to display LSM command Reference Pages.

The following chapter provides a quick reference for some common LSM operations.

2.1 Initial LSM Set Up

To perform initial LSM setup that includes starting the LSM daemon processes, creating the rootdg disk group, and adding devices to the disk group, execute the /sbin/volsetup interactive utility. This utility performs the following tasks:

1. Starts two volume manager kernel error-log daemons to handle errors and recovery for LSM volumes. This step is equivalent to the voliod set 2 command.

2. Starts the volume manager configuration daemon by first killing any previously started daemon. The volume daemon (vold) handles configuration changes requested by the LSM utilities with the kernel. This step is equivalent to the vold -k -m disable command.

3. Reinitializes the /etc/vol/volboot file with a new host ID and an empty list of disks. The /etc/vol/volboot file is used on system startup to locate copies of the rootdg disk-group configuration. This file is critical for starting LSM and should not be deleted. This step is equivalent to the voldctl init command.

4. Sets the disklabel tags for the type of disk you wish to configure, as shown in the following table.

Type of Disk	Tags
Sliced	LSMpubl for public region LSMpriv for private region
Simple	LSMsimp

5. Initializes the rootdg disk group. At least one disk group, the rootdg disk group, must exist for LSM to start. This step is equivalent to the voldg init rootdg command.

6. Initializes the regions of the disk you want to configure. This step is equivalent to the voldisk -f init rz0 command.

7. Adds the disk you want to configure (for example, rz0) to the rootdg disk group. This command causes the disk group's configuration to be copied onto the disk. This step is equivalent to the voldg adddisk rz0 command.

8. Adds the disk you want to configure to the list of disks in the /etc/vol/volboot file. This step is equivalent to the voldctl add disk rz0 command.

9. Requests the volume daemon (vold) to enter enabled mode which causes all disk groups to be imported and the volume and plex device-special files to be created. This step is equivalent to the voldctl enable command.

   The block device-special files for LSM volumes exist in the /dev/vol directory, and the character-special files exist in the /dev/rvol directory. For example, the block device-special file for a volume named vol01 in the rootdg disk group is named /dev/vol/rootdg/vol01 and the character device special file is named /dev/rvol/rootdg/vol01.

2.2 Adding a Disk to a Disk Group

Once LSM has been initialized with the /sbin/volsetup utility, you can add more physical disks or disk partitions to the rootdg disk group by executing the interactive voldiskadd utility. This utility requires that a disklabel already exist on the device. Refer to the disklabel(8) reference page for complete information.

To initialize a disk without adding it to a disk group, use the voldisksetup(8) command.

1. Adding a sliced disk.
   A sliced disk is configured with the following partitions:

   - A private region that holds administrative data. It defaults to 256KB storage.

- A public region which is a large partition that holds user data.

In order to add a sliced disk to a disk group, the name of the disk you want to add should not include a partition. The voldiskadd utility modifies the disklabel on the disk to contain the label "LSMpriv" for the private region, and "LSMpubl" for the public region.
Adding a disk as a sliced disk is advantageous because it allows both LSM and utilities other that LSM to access the partition in the public region. This step is equivalent to the voldiskadd rz1 command.

2. Adding a simple disk.

   A simple disk has both the private and the public region in the same partition. The only way to get at the user data on a simple disk is to use LSM. To create a simple disk, use the voldiskadd utility to enter the partition name. This step is equivalent to the voldiskadd rz1g command.

3. Adding a no private-region disk.
   Refer to Appendix B in the Logical Storage Manager System Administrator's Guide for information about using the LSM encapsulation tools.

The following list shows individual commands that are equivalent to the actions performed by the voldiskadd utility:

1. Edit disklabel

2. voldisk -f init rz3c

3. voldg adddisk rz3c

To add a physical disk to LSM with a specific private region size, use the voldisksetup(8) command. For example, use the following command to initialize a sliced LSM disk with a private region size of 2048 sectors:

voldisksetup -i rz3 privlen=2048

The LSM disk can be added to a disk group later by using the voldg command.

2.3 Creating a Volume from a Disk Group

Once you have created a disk group and added disks, you can create volumes with the volassist command. For example:

```
# volassist -g disk group make volume length
attribute=value
```

Note that you can create a volume in a disk group using the instructions in the following list, or by using the dxlsm graphical user interface.

The following list contains examples of commands to create LSM volumes.

1. To create a 10MB volume using non-reserved disks in the rootdg disk group, enter the following command:

   ```
   # volassist -g rootdg make vol01 10m
   ```

2. To create a 1024KB volume using nonreserved disks in the dg1 disk group, enter the following command:

   ```
   # volassist -g dg1 make vol02 1024k
   ```

3. To create a volume on a specified disk in the rootdg disk group, enter the following command:

   ```
   # volassist -g rootdg make vol03 200000s rz7
   ```

4. To create a volume with 200000 sectors using nonreserved disks in the rootdg disk group excluding the rz9 disk, enter the following command:

   ```
   # volassist -g rootdg make vol03 200000s !rz
   ```

5. To create a 20MB striped volume from the rootdg disk group using three LSM disks with a stripe width of 64K (the default), enter the following command:

   ```
   # volassist -g rootdg make vol04 20m layout=
   nstripe=3
   ```

2.4 Mirroring a Volume

Once a volume is created and enabled, you can create and attach new mirrors to the volume using the volassist utility. You can create and attach new mirrors to a volume using the examples in the following list as a guide, or by using the dxlsm graphical user interface.

1. The following command creates three mirrors of the vol02 volume in the dg1 disk group. The command is executed in the background because it may take a long time for the command to complete:

   ```
   # volassist -g dg1 mirror vol02 nmirror=3 &
   ```

2. The following command creates a 30MB mirrored volume named vol05 from the rootdg disk group. The mirror=yes option specifies the number of mirrors as two. This is the default.

   ```
   # volassist -g rootdg make vol05 30m mirror=
   ```

2.5 Changing the Size of a Volume

You can increase or decrease the size of a volume using the volassist utility.

Note

> A volume containing one or more striped mirrors or plexes cannot grow in size.

You can change the size of a volume using the following examples as guidelines, or using the dxlsm graphical user interface.

1. Enter the following command to increase the size of the vol01 volume by 2 MBs:

   ```
   # volassist growby vol01 2m
   ```

2. Enter the following command to decrease the size

of the vol01 volume by 1024KBs:

```
# volassist shrinkby vol01 1024k
```

Note

> UFS file systems cannot take advantage of the extra space in a grown LSM volume. Also, shrinking an LSM volume with a UFS file system will cause loss of data.

2.6 Removing a Volume From a Disk Group

A volume can be removed from a disk group using the voledit utility. The volume you intend to remove must not be mounted when you enter the voledit command. You can remove a disk from a disk group using the following command, or using the dxlsm graphical user interface.

Enter the following command to remove vol01 from the rootdg disk group:

```
# voledit -g rootdg -rf rm vol01
```

2.7 Removing a Disk or Partition From a Disk

A disk or partition of a disk can be removed from a disk group using the voldg command as shown in the following examples, or using the dxlsm graphical user interface.

1. Enter the following command to remove the LSM disk rz0 from the dg1 disk group:

   ```
   # voldg -g dg1 rmdisk rz0
   ```

2. Enter the following command to remove a disk from the LSM configuration that does not belong to any disk group:

   ```
   # voldisk rm rz0
   ```

2.8 Replacing a Failed Disk Drive

If a disk needs to be replaced due to a disk failure, use the voldg command to remove the bad disk. You can replace a bad disk using the command as shown in the following examples, or using the dxlsm graphical user interface.

1. Enter the following command to replace the failed disk disk01:

   ```
   # voldg -g rootdg -k rmdisk disk01
   ```

   In the example, the -k option saves the disk-media records so that when the disk is replaced using the -k option, the disk is usable just as it was before the disk failure.

2. Replace the disk01 disk using the following command. This command gives the rz0 disk the same media name (disk01) as the previously failed disk:

   ```
   # voldg -g rootdg -k adddisk disk01=rz0
   ```

2.9 Adding Another Disk Group

Once the rootdg disk group exists, you can add another disk group with the voldg utility, or using the dxlsm graphical user interface. The following shows the syntax for the voldg command:
voldg init disk_group_name [medianame=] accessname..

For example, the following command creates the dg1 disk group and places disks rz10 and rz11 in the disk group:

```
# voldg init dg1 rz10 rz11
```

2.10 Obtaining Status Information

The following list describes several ways in which you can obtain status information about LSM volumes:

1. To obtain disk group, disk media, volume, plex, and subdisk records, use the volprint command. Using this command without any arguments provides information about all of the LSM entities. The

syntax for the volprint command is as follows:

volprint -htA

In the above example:

-h lists complete hierarchies below selected records. For volumes, this list includes all associated plexes and subdisks. For plexes, this list includes all associated subdisks.
-t provides single-line output records that depend upon the configuration record type. See the volprint(8) reference page for complete information about the types of output supplied for disk groups, subdisks, disk media records, plexes, and volumes.

-A prints records from all active (imported) disk groups. Each disk group represented in the output is separated from other disk groups by blank lines.

2. To obtain disk access records, or physical disk information. use the voldisk command, as shown here:

   # voldisk list

3. To obtain free disk space in all disk groups, enter the following command:

   # voldg free

Logical Storage Manager

Installation Guide

April 1994

Product Version: Logical Storage Manager Version 1.0

Operating System and Version: DEC OSF/1 Version 2.0

This document describes how to install DEC OSF/1 Logical Storage Manager Version 1.0.

digital equipment corporation
Maynard, Massachusetts

This guide describes installing Version 1.0 of the DEC OSF/1 Logical Storage Manager (LSM).

1.1 Pre-Installation

The following sections describe the LSM software requirements and the contents of the LSM distribution kit.

1.1.1 Software Requirements

The LSM Version 1.0 kit can be installed only on DEC OSF/1 systems running Version 2.0 or later. See the DEC OSF/1 Logical Storage Manager Release N additional information and known problems with LSM software.

1.1.2 Distribution Kit

The LSM Version 1.0 distribution media comprises the following items:

- Software Product Description (SPD)
- Software subsets, including:
  - LSMBASE100 subset contains the LSM base components
  - LSMBIN100 subset contains the LSM kernel objects and LSM real-time kernel objects
  - LSMBINCOM100 subset contains the LSM kernel include files
  - LSMMAN100 subset contains the LSM reference pages
  - LSMX11100 subset contains the LSM Motif-based GUI management tool
- Logical Storage Manager documentation set, including:

- *The Logical Storage Manager Release Note*

- *The Logical Storage Manager Basic User's*

- *The Logical Storage Manager System Administrator's Guide*

- *The Logical Storage Visual Administrato User's Guide*

- *The Logical Storage System Administrato Reference Guide*

1.2 Installing the LSM Version 1.0 Kit

Use the instructions in the following list to install the LSM Version 1.0 kit from the distribution media.

1. Mount the LSM Version 1.0 distribution media.

2. Use the *setld -l* command to install the LSM subsets. The LSM subsets have the following installation dependencies:

   - Install LSMBINCOM100 first

   - LSMBIN100 depends on LSMBASE100

   - LSMBASE100 depends on LSMBIN100

   - LSMX11100 depends on LSMBASE100

   - LSMMAN100 depends on OSFDCMT200

The *setld* installation procedure:

- Automatically configures and builds a new kernel with LSM devices. If you have real-time objects installed, LSM asks whether to build regular or real-time kernels.

Note

> During LSM layered-product installation, the installation procedure asks whether the installation procedure should automatically build the kernel. For

example:

Do you want setld to automatically configure and build the kernel
n):

You must enter 'y' in response to this question to ensure that LSM will be included as part of any future kernel builds. If you enter an 'n', you can correct the situation by entering the following command:

```
# /usr/sbin/setld -c LSMBIN100 INSTALL
```

This command reconfigures the LSMBIN100 subset that contains the LSM kernel objects and real-time kernel objects. Then, when the installation procedure asks the question about automatically configuring and building the kernel, enter 'y'. Refer to the setld(8) reference page for more details about the -c option to the setld command.

- Dynamically configures the LSM drivers into the system's new kernel. The host-specific kernel configuration file is not modified by the LSM installation.

Once the setld installation has completed, you must reboot the system using the shutdown command. This causes the LSM drivers to become a part of the running kernel.

When the reboot finishes and the system enters multiuser mode, you must execute the /sbin/volinstall utility which adds the following line to the /etc/inittab file:

```
lsm:23:wait:/sbin/lsmbstartup < /dev/console > /dev/console 2>&1
```

This command is necessary so that, at machine boot time, LSM is started prior to the execution of the bcheckrc command. This causes the fsck utility to analyze file systems on LSM volumes and mount the volumes automatically.

1.3 Configuring LSM

You must install an LSM license before LSM can be configured to run. Refer to the DEC OSF/1 System

Administration Guide for information about using LMF to install the LSM license.

The following subsections describe the concept of LSM disks and how to setup the initial LSM configuration.

1.3.1 Introduction to LSM Storage Concepts

An LSM logical volume constitutes up to eight mirrors, also referred to as plexes. Each plex is a copy of the data which, in normal circumstances, is a duplicate copy of all other plexes in the volume. A volume that is not mirrored has only one plex.

A plex is formed by a collection of contiguously allocated disk spaces, called subdisks. Subdisks, in turn, are allocated by LSM from LSM disks. An LSM disk corresponds to one or two disk partitions. The following list describes the types of LSM disks:

- Physical disks

  Must be added to the LSM environment as LSM disks before they can be used to create LSM volumes. Refer to the voldiskadd(8) reference page for information about adding physical disks to LSM.

- Sliced disks
  Corresponds to two disk partitions:

  - A small partition called the "private region" that is used to keep administrative data

  - A large partition called the "public region" that is used to store user data.

  There can be only one sliced disk per physical disk. The disklabel tags for the partitions of a sliced disk are set to "LSMpriv" and "LSMpubl."

- Simple disks
  Includes both public and private regions in the same partition. The disklabel tag of a simple disk partition is set to "LSMsimp."

- Nopriv disks

  Nopriv LSM disks do not have a private region; they rely on the private regions of other LSM disks in the disk group.

A sliced disk and a simple disk can reside on the same disk. The disklabel tags identify the partitions to LSM as LSM disks.

A disk group is a named collection of LSM disks that share a common configuration. LSM volumes cannot span different disk groups.

The volintro(8) reference page provides a quick reference of common LSM terminology and command usage. Refer to the Logical Storage Manager Administrator's Guide for a complete introduction and description of LSM concepts.

1.3.2 Configuring the Root Disk Group and Start

At least one disk group, the rootdg root disk group, must exist in order for LSM to start. The procedure to create the rootdg disk group is as follows:

1. Run the /sbin/volinstall utility to create the LSM daemon special-device files. This utility optionally adds an entry to the /etc/inittab file to start LSM automatically during machine reboot.

2. Select the physical disks to serve as members of the root disk group rootdg. For example, /dev/rz8.

3. Execute the /sbin/volsetup command. This command invokes an interactive utility that guides you through the steps necessary to set up the rootdg disk group. The volsetup command also starts the LSM daemons.

4. Once LSM is operational, you can run the /usr/bin/X11/dxlsm command to invoke the LSM Visual Administrator (dxlsm).

5. Use the dxlsm interface or the command interface to create LSM volumes.

The Logical Storage Manager Basic User's Guid provides basic knowledge about managing LSM volumes using dxlsm and the command interface. The basic dxlsm operations allow you to create simple, mirrored, and striped volumes.

1.4 Accessing LSM Volumes

Once you create LSM volumes using the LSM command interface or the dxlsm graphical interface tools, users and applications can access LSM volumes in the same way that they access any disk device. The block-special files for LSM volumes are located in the /dev/vol directory. The character-special files are located in the /dev/rvol directory.

To create a new file system on an LSM volume, use the newfs command with the disk type argument specifying any known disk type. The disk type is used to provide the sector and track size information for the newfs command. For example, to create a new UNIX file system (UFS) file system on the LSM volume vol01, enter the following commands:

```
# newfs /dev/rvol/rootdg/vol01 rz26
# mount /dev/vol/rootdg/vol01 /mnt
```

/

/etc/exports

cd1

cd2

77

DEC OSF/1

Logical Storage Manager

Order Number: AA-Q3NCB-TE

August 1994

Product Version: DEC OSF/1 Logical Storage Manager Version 1.1

Operating System and Version: DEC OSF/1 Version 3.0

This manual provides an overview of the DEC/OSF1 Logical Storage Manager (LSM) concepts and describes how to configure and maintain systems under the control of LSM. It also describes advanced volume management concepts including striping, mirroring, and disk group management. Reference information is included to help you perform common disk management operations using the LSM interfaces.

Digital Equipment Corporation
Maynard, Massachusetts

Contents

About This Guide

1 Introduction to the Logical Storage Manager

2 LSM Configurations

3 Setting Up LSM

4 Graphical User Interface

5 Advanced Visual Administrator Interface Operations

6 Menu Interface

7 Command Line Interface

8 Advanced Volume Operations

C Visual Administrator Reference

Figures

Tables

About This Guide

The DEC OSF/1 Logical Storage Manager (LSM) provides high data availability for disk storage devices on Alpha AXP systems. This guide explains LSM concepts and the system administrator tools and techniques you need to implement and manage LSM on Alpha AXP systems.

LSM is intended to replace the Logical Volume Manager (LVM). For LVM users who have not migrated to LSM yet, see Chapter 3 for information about the LSM encapsulation tools to help you migrate UNIX style partitions, LVM volumes, and AdvFS storage domains to LSM.

Audience

The *Logical Storage Manager* is intended for system administrators who need to configure and maintain systems under the control of LSM. This guide assumes that the reader has a:

- Working knowledge of the DEC OSF/1 operating system
- Basic understanding of system administration
- Basic understanding of volume management
- Basic understanding of disk structures

Organization

This manual consists of ten chapters and three appendixes:

Chapter 1	Contains an overview of the Logical Storage Manager, including a discussion of LSM fundamentals, components of an LSM configuration, LSM interfaces, and the LSM system architecture.
Chapter 2	Describes LSM configurations, including concatenated disks, mirroring, and striping.
Chapter 3	Provides information to help you start LSM and perform initial LSM set up, and encapsulate existing user data.
Chapter 4	Illustrates how to perform basic disk and volume operations using the Visual Administrator graphical user interface (dxlsm).

Chapter 5	Describes advanced usages of the dxlsm interface and how you can configure the Visual Administrator to your personal preferences and system requirements.
Chapter 6	Contains information about the LSM Support Operations menu interface (voldiskadm).
Chapter 7	Contains information about the command line interface available with LSM.
Chapter 8	Contains information about performing advanced LSM management operations using the command line interface.
Chapter 9	Discusses advanced LSM volume management concepts.
Chapter 10	Suggests performance priorities and guidelines for use with LSM.
Appendix A	Contains a list of error messages generated by LSM, a description of what the messages mean, and user actions the system administrator should take should these messages occur.
Appendix B	Describes the procedures used to preserve data. This appendix discusses ways to prevent data loss due to disk failure and to prevent loss of system availability due to failure of a key disk (a disk involved with system operation).
Appendix C	Contains reference information for using the Visual Administrator (dxlsm) interface.

Related Documents

The following documents provide information related to the Logical Storage Manager:

- The *Installation Guide* describes how to install the Logical Storage Manager Version 1.1.
- The *Release Notes* describes supported LSM features and products, and known problems and fixes.

The printed version of the DEC OSF/1 documentation set is color coded to help specific audiences quickly find the books that meet their needs. (You can order the printed documentation from Digital.) This color coding is reinforced with the use of an icon on the spines of books. The following list describes this convention:

Audience	Icon	Color Code
General Users	G	Teal
System Administrators	S	Red
Network Administrators	N	Yellow
Programmers	P	Blue
Reference Page Users	R	Black

Some books in the documentation set 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 DEC OSF/1 documentation set. The *Documentation Overview* provides information about all of the books in the DEC OSF/1 documentation set.

Reader's Comments

Digital welcomes your comments on this or any other DEC OSF/1 manual. A Reader's Comment form is located in the back of each printed DEC OSF/1 manual and on line in the following location:

`/usr/doc/readers_comment.txt`

You can send your comments in the following ways:

- Internet electronic mail: `readers_comment@zk3.dec.com`
- Fax: 603-881-0120 Attn: UEG Publications, ZK03-3/Y32
- Mail:

  Digital Equipment Corporation
  UEG Publications Manager
  ZK03-3/Y32
  110 Spit Brook Road
  Nashua, NH 03062-9987

  The Reader's Comment form located in the back of each printed manual is postage paid if you mail it in the United States.

If you have suggestions for improving particular sections or find any errors, please indicate the manual title, order number, and section numbers. Digital also welcomes general comments.

Conventions

The following conventions are used in this manual:

`%` `$`	A percent sign represents the C shell system prompt. A dollar sign represents the system prompt for the Bourne and Korn shells.
#	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.
Return	In an example, a key name enclosed in a box indicates that you press that key.
Ctrl/*x*	This symbol indicates that you hold down the first named key while pressing the key or mouse button that follows the slash. In examples, this key combination is enclosed in a box (for example, Ctrl/C).
mouse	The term mouse refers to any pointing device, such as a mouse, a puck, or a stylus.
MB1, MB2, MB3	On a three-button mouse, MB1 indicates the left mouse button. MB2 indicates the middle mouse button, and MB3 indicates the right mouse button. (You can redefine the buttons by using platform-specific window management facilities.)
Menu → Option → Submenu Option	The right arrow indicates an abbreviated instruction for choosing a menu option or submenu option. The following example means pull down the Modify menu, move the pointer to pull down the Image submenu, and choose the Clear option: Choose Modify → Image → Clear

Introduction to the Logical Storage Manager 1

This chapter introduces the DEC OSF/1 Logical Storage Manager (LSM), its features and capabilities, concepts, and terminology. The `volintro`(8) reference page also provides a quick reference of LSM terminology and command usage.

1.1 What Is LSM?

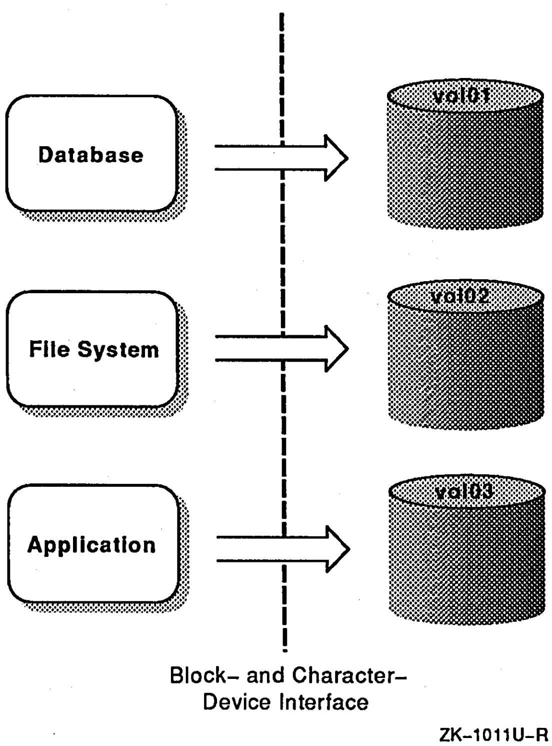
The DEC OSF/1 Logical Storage Manager (LSM) is an integrated, host-based disk storage management tool that protects against data loss and improves disk input/output (I/O) performance. System administrators use LSM to perform disk management functions without disrupting users or applications accessing data on those disks.

1.2 LSM Fundamentals

LSM builds virtual disks, called volumes, on top of UNIX system disks. A *volume* is a DEC OSF/1 special device that contains data used by a UNIX file system, a database, or other application. LSM transparently places a volume between a physical disk and an application which then operates on the volume rather than on the physical disk. A file system, for instance, is created on the LSM volume rather than on a physical disk.

Figure 1-1 shows how disk storage is handled in systems that use LSM.

Figure 1-1: Disk Storage Management with LSM



In general, disk storage management often requires that for each file system or database created, you must be able to:

- 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.

All of these requirements can be done easier when you use LSM. Table 1-1 compares disk storage management requirements for systems running with and without LSM.

Table 1-1: Disk Storage Management With and Without LSM

Requirement	Without LSM	With LSM
Space Allocation	UNIX disks are divided into *partitions*. A partition is defined by its start address on the physical disk and its length. The administrator must partition the disks according to the needs of the users on the system. Partitions cannot be moved or extended in size once the partition is in use.	LSM obtains space for a file system or raw database by creating an LSM volume of the appropriate size. A 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 extend volumes by adding disk space that is not contiguous with the space already allocated, and to create 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*. Reconfiguring disks (for example, moving a disk to a new controller) requires a change of the addresses through which the partitions are accessed because the disk's unit number has changed. The administrator must also manually change all references to the partitions on the reconfigured disk devices.	LSM volumes are addressed using a *volume name* that is independent of the manner in which the volume is mapped onto physical disks. You establish a symbolic *disk name* or *disk media name* to refer to a disk that is managed by LSM (for example: `disk01`). This makes it possible to easily readjust LSM volume and space allocation in case disks are moved in the configuration without effecting the application.
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. In addition, because the partitioning of disks cannot be changed easily, it is difficult for the administrator to ensure that data is placed on the available disk drives for optimal access and performance.	LSM volumes can be accessed through the standard block- and character-device interfaces, using names that are independent of the physical storage addresses used by the volume. In addition, because you can change LSM volume configurations online without interrupting user access to the data, you can dynamically change data placement for optimal access and performance.

1.3 LSM Features

Table 1-2 summarizes the LSM features.

Table 1-2: LSM Features and Benefits

Feature	Benefit
Manages disk administration	Frees you from the task of partitioning disks and maintaining the space administration. However, LSM allows you to keep control over disk partitioning and space allocation, if desired.
Allows transparent disk configuration changes	Allows you to change the disk configuration without rebooting or otherwise interrupting users. Also allows routine administrative tasks, such as file system backup, while the system is in active use.
Stores large file systems	Enables multiple physical disks to be combined to form a single, larger logical volume. This capability, called *concatenation*, removes limitations imposed by the actual physical properties of individual disk sizes, by combining the storage potential of several devices. Note that disk concatenation is available on all systems, including those that do not have the LSM software licensed.
Ease of system management	Simplifies the management of disk configurations by providing convenient interfaces and utilities to add, move, replace, and remove disks.
Protects against data loss	Protects against data loss due to hardware malfunction by creating a *mirror* (duplicate) image of important file systems and databases.
Increases disk performance	Improves disk I/O performance through the use of *striping* which is the interleaving of data within the volume across several physical disks.

1.4 Hardware and Software Requirements

The following sections describe the hardware and software requirements, licensing, and configuration limitations for LSM.

1.4.1 Hardware Requirements

LSM does not depend on specific hardware in order to operate. All functions can be performed on any supported Alpha AXP computer running DEC OSF/1 Version 3.0 or later. There are no restrictions on the devices supported beyond the valid configurations defined in the DEC OSF/1 Version 3.0 Software Product Descriptions (SPD 41.51).

All Small Computer Systems Interface (SCSI) and DIGITAL Storage Architecture (DSA) disks supported by DEC OSF/1 Version 3.0 are supported by LSM. SCSI Redundant arrays of independent disks (RAID) hardware devices are supported as standard disks, with each RAID device-logical unit viewed as a physical disk.

1.4.2 Software Requirements

LSM has these software requirements:

- LSM Version 1.1 requires that the system run DEC OSF/1 software Version 3.0 or later.
- The LSM Visual Administrator is a Motif-based application that requires you have the Basic X Environment subset installed on the system.

1.4.3 Licensing Requirements

The LSM software is furnished under the licensing provisions of the Digital Equipment Corporation Standard Terms and Conditions. However, note that the base UNIX license allows you to use the LSM concatenation and spanning feature. You do not need an LSM software license to include multiple physical disks within a single LSM volume.

To use LSM advanced features (such as mirroring, striping, and the Visual Administrator (dxlsm)), you must have an LSM license. License units for LSM are allocated on an unlimited system use basis.

Refer to the *Guide to Software Licensing* in the DEC OSF/1 documentation set for more information about the DEC OSF/1 License Management Facility (LMF).

1.4.4 Configuration Limitations

The maximum configuration supported by the DEC OSF/1 Logical Storage Manager is defined as follows:

- Maximum of 250 volumes
- Maximum of 8 plexes per volume

  A plex is an identical copy of data. Volumes that are not mirrored have one plex, while mirrored volumes can have anywhere from two to eight plexes.

- Maximum of 256 subdisks per plex.

  A subdisk is a basic unit of disk space allocation for a plex.

- Maximum volume size of 128 Gbytes
- Maximum of 128 disks per system

  Disks are organized into disk groups. A disk group is a named collection of disks that share the same LSM configuration database. Volumes are created within a disk group and are restricted to using disks within that disk group (for example, logical volumes cannot span disk groups).

1.5 Components of an LSM Configuration

LSM consists of physical disk devices, logical entities or objects, and the mappings that connect the physical and logical objects. LSM logically binds together the physical disk devices into a logical LSM volume that represents the disks as a single virtual device to applications and users.

LSM organizes and optimizes disk usage and guards against media failures using these objects:

1. Subdisks
2. Plexes
3. Volumes

Each object has a dependent relationship on the next-higher object, with subdisks being the lowest level objects in the structure, and volumes the highest level. LSM maintains a configuration database that describes the objects in the LSM configuration, and implements utilities to manage the configuration database. Multiple mirrors, striping, and concatenation are additional techniques you can perform with the LSM objects to further enhance the capabilities of LSM.

Table 1-3 describes the LSM objects.

Table 1-3: LSM Objects

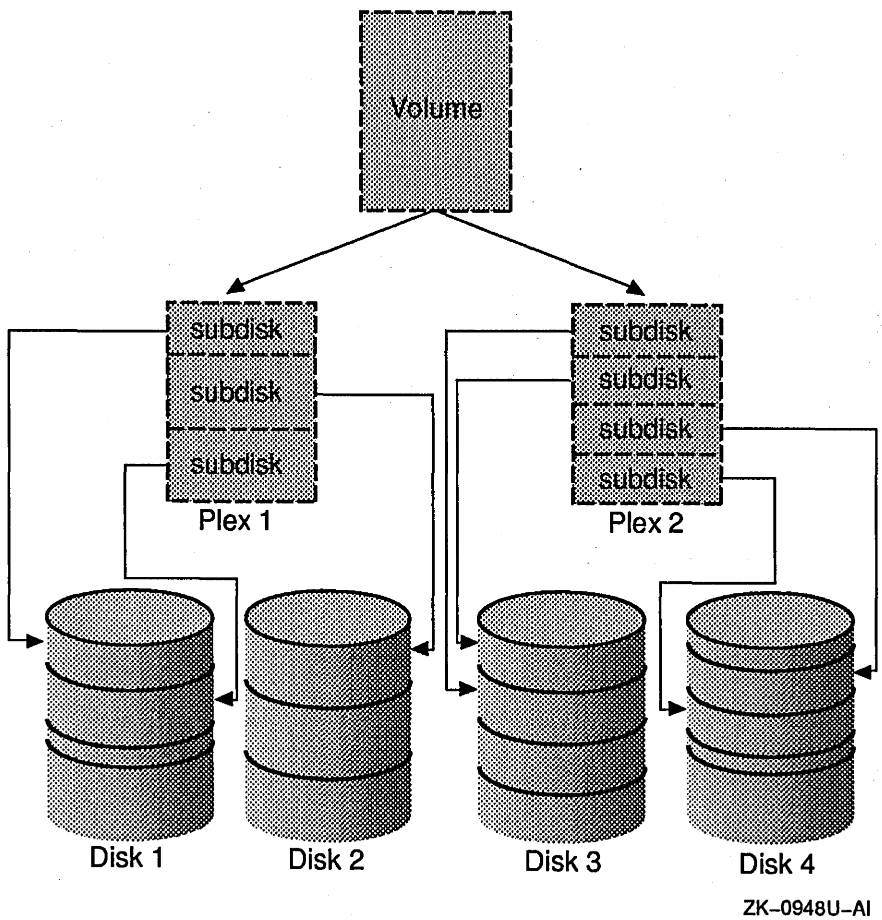
Object	Description
Volume	Represents an addressable range of disk blocks used by applications, file systems, or databases. A volume is a virtual disk device that looks to applications and file systems like a regular disk-partition device. In fact, volumes are logical devices that appear as devices in the `/dev` directory. The volumes are labeled `fsgen` or `gen` according to their usage and content type. Each volume can be composed of from one to eight plexes (two or more plexes mirror the data within the volume). Due to its virtual nature, a volume is not restricted to a particular disk or a specific area thereof. The configuration of a volume can be changed (using LSM utilities) without causing disruption to applications or file systems using that volume.
Plex	A collection of one or more subdisks that represent specific portions of physical disks. When more than one plex is present, each plex is a replica of the volume; the data contained at any given point on each is identical (although the subdisk arrangement may differ). Plexes can have a striped or concatenated organization.
Subdisk	A logical representation of a set of contiguous disk blocks on a physical disk. Subdisks are associated with plexes to form volumes. Subdisks are the basic components of LSM volumes; subdisks form a bridge between physical disks and virtual volumes.
Disk	A collection of nonvolatile, read/write data blocks that are indexed and can be quickly and randomly accessed. LSM supports standard disk devices including SCSI and DSA disks. Each disk used by LSM is given two identifiers: a disk access name and an administrative name.
Disk Group	A collection of disks that share the same LSM configuration database. The root disk group, `rootdg`, is a special private disk group that always exists.

LSM objects have the following relationships:

- A volume consists of from one to eight plexes
- A plex consists of one or more subdisks
- A subdisk represents a specific portion of a disk
- Disks are grouped into disk groups

Figure 1-2 shows the relationships between LSM objects in a disk group.

Figure 1-2: LSM Objects and Their Relationships



1.6 LSM Disks

You can use any standard disk device with LSM, such as SCSI or DSA disks. Standard disk devices are those that can be used with DEC OSF/1 utilities, such as `disklabel` and `newfs`.

Section 1.6.1 and Section 1.6.2 describe the characteristics of standard devices, and how these devices are named for use with LSM.

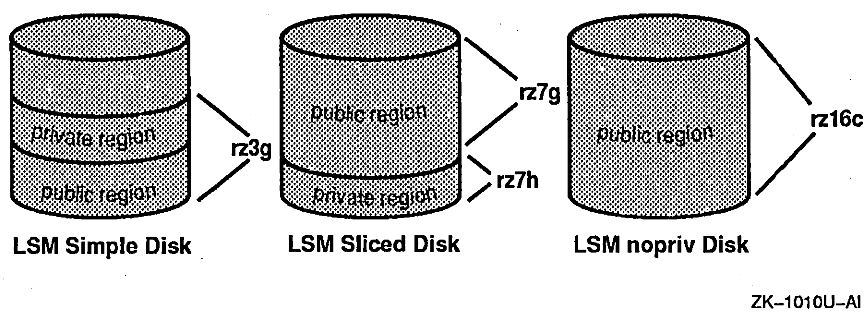
1.6.1 Types of LSM Disks

An LSM disk typically uses two regions on each physical disk. These regions have the following characteristics:

- A small region, called the private region, in which LSM keeps its disk media label and a configuration database
- A large region, called the public region, that forms the storage space for building subdisks

Figure 1-3 shows the private and public regions in LSM *simple* and *sliced* disks. The third disk, an LSM *nopriv* disk, does not contain a private region. All of these types of disks can be added into an LSM disk group.

Figure 1-3: Types of LSM Disks



In the figure:

- Simple disks have both public and private regions in the same partition (`rz3g`).
- Sliced disks use the disk label on a disk to identify the private (`rz7h`) and the public (`rz7g`) regions.
- Nopriv disks have no private region so they do not contain LSM configuration information. Because of this, nopriv disks can only be added to an existing disk group that includes a simple disk or a sliced disk.

LSM configuration databases are stored on the private region of each LSM disk except the nopriv disk. The public regions of the LSM disks collectively form the storage space for application use. For purposes of availability, each simple or sliced disk contains two copies of its configuration database.

1.6.2 Naming LSM Disks

When you perform disk operations, you should understand the disk-naming conventions for a *disk access name* and *disk media name*. This is because disk access names and disk media names are treated internally as two types of LSM disk objects. Some operations require that you specify the disk access name, while others require the disk media name. The following definitions describe these disk naming conventions:

- *Disk access name* (also referred to as *devname* or *device name*)

  The device name or address used to access a physical disk. A disk access name is of the form:

  dd[*l*] *n*[*nnn*] [*p*]

  The elements in the disk access name are described in the table below:

Element	Description
dd	A two-character device mnemonic that shows the disk type. Use `ra` for DSA disks, and `rz` for SCSI disks.
[l]	The SCSI logical unit number (LUN), in the range from a to h, to correspond to LUNs 0 through 7. This argument is optional and used for SCSI hardware Redundant Arrays of Independent Disks (RAID) devices.
n[nnn]	The disk unit number ranging from 1 to 4 digits.
[p]	The partition letter, in the range from a to h, to correspond to partitions 0 through 7. This argument is optional.

  For example, rz in the device name rz3 represents a pseudonym for a SCSI disk, and rzb10h represents a disk access name for a SCSI disk having a LUN of one, which applies to Digital SCSI RAID devices.

  For a simple disk or a nopriv disk, you must specify a partition letter. For example, `rz3d`. For a sliced disk, you must specify a physical drive that does not have a partition letter (for example, `rz3`). The proper full pathname of the `d` partition on this sliced device is `/dev/rz3d`. However, for easier reading, this document often lists only the disk-access name and `/dev` is assumed. Also, note that you do not specify `/dev` in front of the device name when using LSM comamnds.

- *Disk-media name* (also referred to as the *disk name*)

  An administrative name for the disk, such as *disk01*. If you do not assign a disk-media name, it defaults to `disk`*nn*, where *nn* is a sequence number if the disk is being added to `rootdg`. Otherwise, the default disk-media name is `groupname`*nn*, where `groupname` represents the name of the disk group to which the disk is added.

1.7 LSM Disk Groups

You can organize a collection of physical disks that share a common configuration or function into disk groups. LSM volumes are created within a disk group and are restricted to using disks within that disk group.

Disk groups can be used to simplify management and provide data availability. For example:

- On a system with a large number of disks, you might want to divide disk usage into a few disk groups based on function. This would reduce the size of the LSM configuration database for each disk group as well as reduce the amount of overhead incurred in configuration changes.
- If a system will be unavailable for a prolonged amount of time due to a hardware failure, the physical disks in a disk group can be moved for use on another system. This is possible because each disk group has a self-describing LSM configuration database.

All systems with LSM installed have the `rootdg` disk group. By default, operations are directed to this disk group. Most systems do not need to use more than one disk group.

Note

> You do not have to add disks to disk groups when a disk is initialized; disks can be initialized and kept on standby as replacements for failed disks. A disk that is initialized but not added to a disk group can be used to immediately replace a failing disk in any disk group.

Each disk group maintains an LSM configuration database that contains detailed records and attributes about the existing disks, volumes, plexes, and subdisks in the disk group.

1.7.1 LSM Configuration Databases

The LSM configuration database contains records describing all the objects (volumes, plexes, subdisks, disk media names, and disk access names) being used in a disk group.

Two identical copies of the LSM configuration database are located in the private region (illustrated in Figure 1-3) of each disk within a disk group. LSM maintains two identical copies of the configuration database in case of full or partial disk failure.

The contents of the rootdg configuration database are slightly different. The difference between a rootdg configuration database and an ordinary LSM configuration database is that the rootdg configuration database contains records for disks outside of the rootdg disk group in addition to the ordinary disk-group configuration information. Specifically, a rootdg configuration includes disk-access records that define all disks on the system.

The volboot file is used by the LSM volume daemon vold utility during startup to locate copies of the rootdg configuration database. This file may list disks that contain configuration copies in standard locations, and can also contain direct pointers to configuration copy locations. The volboot file is located in /etc/vol.

1.7.2 Moving and Replacing LSM Disks in a Disk Group

When a disk is added to a disk group it is given a disk media name, such as disk02. This name relates directly to the physical disk. LSM uses this naming convention (described in Section 1.6.2) because it makes the disk independent of the manner in which the volume is mapped onto physical disks. If a physical disk is moved to a different target address or to a different controller, the name disk02 continues to refer to it. Disks can be replaced by first associating a different physical disk with the name of the disk to be replaced, and then recovering any volume data that was stored on the original disk (from mirrors or backup copies).

1.8 LSM Interfaces

LSM provides three different methods to manage LSM disks: a graphical user interface, a menu interface, and a command-line interface. You can use any of these interfaces (or a combination of the interfaces) to change volume size, add plexes, and perform backups or other administrative tasks. Table 1-4 describes these LSM interfaces.

Table 1-4: LSM Administration Interfaces

Interface	Type	Description
Visual Administrator (dxlsm)	Graphical	Uses windows, icons, and menus to manage LSM volumes. The dxlsm interface requires a workstation and the Basic X Environment subset installed to provide its icon and menu-driven approach to volume management. This simple-to-use interface translates mouse-based icon operations into LSM commands. **Note** The Visual Administrator (dxlsm) interface requires the LSM software license.
Support Operations (voldiskadm)	Menu	Provides a menu interface to manage LSM volumes. Each entry in the main menu leads you through a particular operation by providing you with information and asking you questions. Default answers are provided for many questions so that common answers can be selected easily. This is a character-cell interface that does not require a workstation for operation.
Command Line	Command	Provides two approaches to LSM administration. With the top-down approach, you use the LSM `volassist` command to automatically build the underlying LSM objects. With the bottom-up approach, you use several commands (including `volmake`, `volplex`, `volume`, and `volsd`) to build individual objects in order to customize the construction of an LSM volume.

Once a disk is under the control of LSM, all system administrative tasks relating to that disk must be performed using LSM utilities and commands. If, for instance, you install a file system on an LSM-controlled disk using physical disk paths rather than the LSM interfaces, LSM will be unaware that the new file system exists and will reallocate its space.

The LSM interfaces can be used interchangeably. LSM objects created by one interface are fully interoperable and compatible with objects created by the other interfaces.

Refer to Chapter 4, through Chapter 7 for information about using these LSM interfaces.

Note

Most of the LSM commands and utilities can be used by privileged users only.

1.9 Accessing LSM Volumes for I/O

Once you create LSM volumes using one of the LSM interfaces, users and applications can access LSM volumes in the same way that they access any disk device:

- Block-special files for LSM volumes are located in:

  `/dev/vol/`*`diskgroupname`*

- Character-special files are located in:

  `/dev/rvol/`*`diskgroupname`*

The variable *`diskgroupname`* refers to the disk group name that contains the volume. Note that volumes in the `rootdg` disk group are located in the the `/dev/vol/` and the `/dev/rvol/` directories too.

To create a new UNIX file system (UFS) file system on an LSM volume, use the `newfs` command with the disk type argument specifying any known disk type. The disk type is used to provide the sector and track size information for the `newfs` command. For example, to create a new UFS on the LSM volume vol01, enter the following commands:

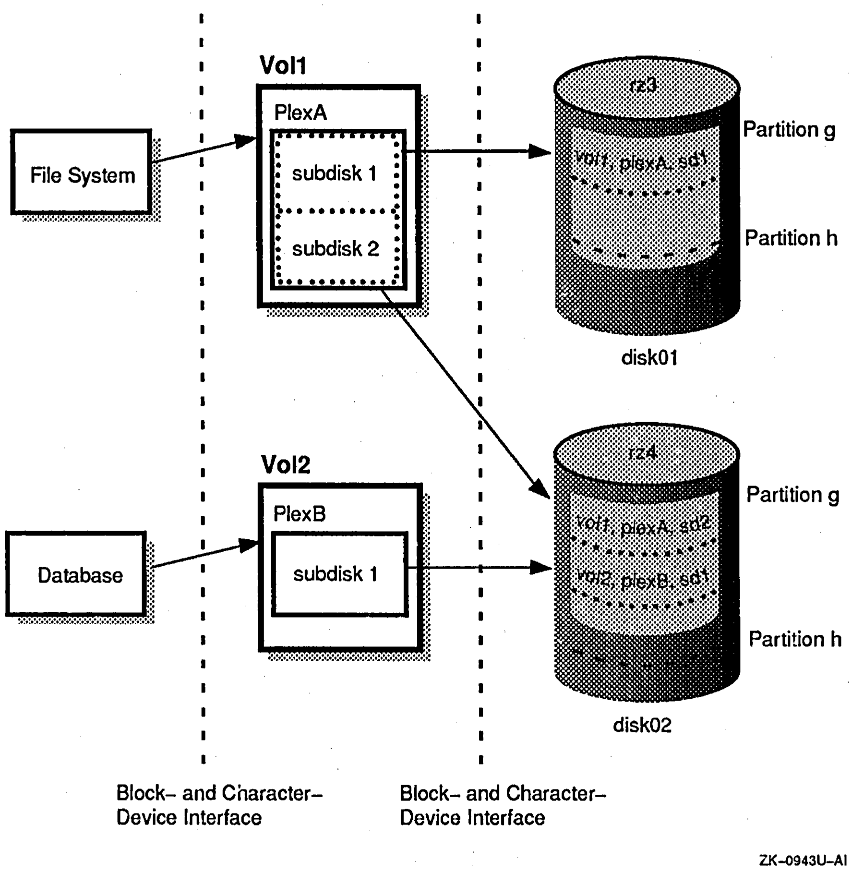
```
# newfs /dev/rvol/rootdg/vol01 rz26
# mount /dev/vol/rootdg/vol01 /mnt
```

On a system that does not have LSM installed, I/O activity from the UNIX system kernel is passed through disk device drivers that control the flow of data to and from disks. When LSM is installed, the I/O that passes from the kernel to the LSM volume device driver, then to the disk device drivers.

The LSM software maps the logical configuration of the system to the physical disk configuration. This is done transparently to the file systems, databases, and applications above it because LSM supports the standard DEC OSF/1 block-device and character-device interfaces to store and retrieve data on LSM volumes. Thus, applications do not need to be changed to access data on LSM volumes.

Figure 1-4 shows how file systems, databases, and applications store and retrieve data on LSM volumes.

Figure 1-4: I/O Activity to LSM Volumes

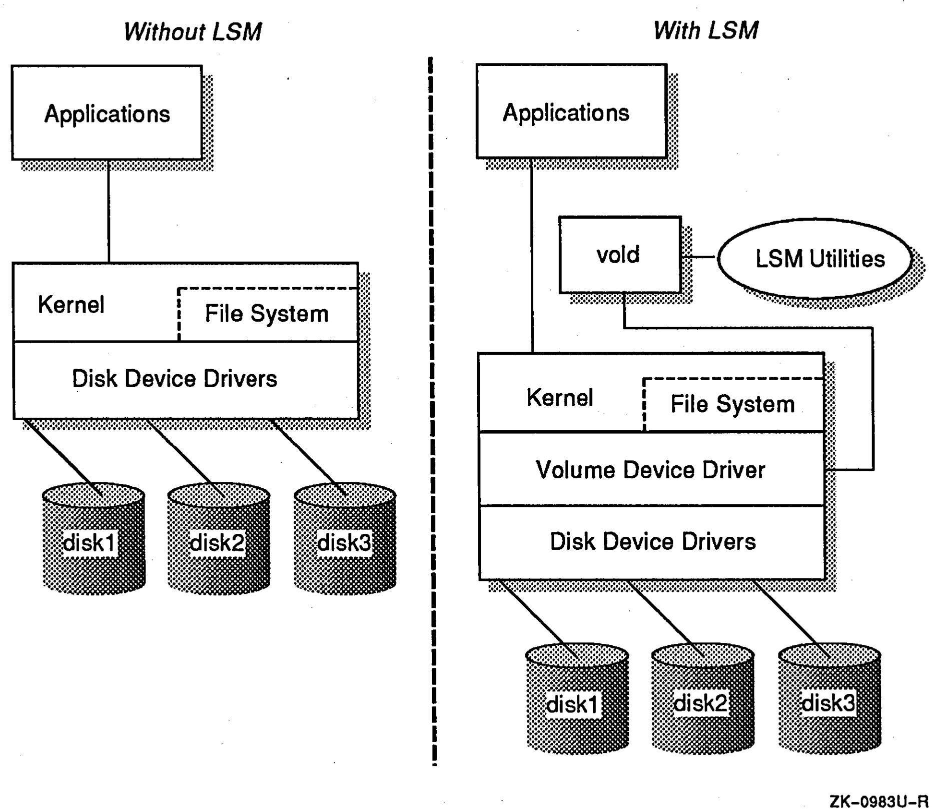


Section 1.10 describes the LSM volume device driver that handles I/O to LSM volumes, and other central software components.

1.10 LSM System Architecture

The system architecture in Figure 1-5 show the relationships between the kernel, file systems and application databases, and the device drivers for systems with and without LSM installed.

Figure 1-5: LSM Software Architecture



The central components of the LSM architecture, the volume device driver and the volume configuration daemon (`vold`), are shown in Figure 1-5 and described in the following list. The list also describes the volume extended I/O daemon (`voliod`) because this process is started immediately after the initial installation of the `vold` daemon.

- Volume Device Driver

  The LSM *volume device driver* is an installable driver that maps the logical configuration of the system to the physical disk configuration. Although the volume device driver resides between the disk device and the applications, it maps the configuration transparently to the file systems, databases, and applications above it. Thus, applications do not need to be changed in order to access data on LSM volumes.

  The volume device driver implements striping or concatenation, read and write error handling, and mirroring in LSM configurations.

The volume device driver supports the devices described in Table 1-5 in LSM configurations.

Table 1-5: Devices Supported by the Volume Device Driver

Device	Description
Volume	LSM creates a volume device-special file for every virtual disk (volume) defined by the system administrator (for example, /dev/vol/...).
Plex	Plex devices are used internally by LSM for resynchronizing mirrors and other special operations. LSM creates a character device-special file for every plex in /dev/plex.
Additional devices	Additional devices are used to communicate between LSM utilities and the kernel (volconfig, volevent, volinfo, voliod).

- Volume Configuration Daemon (vold)

  The volume configuration daemon, vold, is responsible for the interface of the LSM utilities to the kernel. All LSM configuration database changes are centralized through the vold daemon, including:

  - Creation and deletion of LSM objects
  - LSM configuration changes due to system administrator interaction
  - LSM configuration changes due to asynchronous error events

  The vold daemon packages the LSM configuration change into a transaction and passes it to the LSM volspec driver to record the actual change. Because these changes are performed by the daemon and not the kernel, the robustness of LSM is increased.

- Volume Extended I/O Daemon (voliod)

  The volume extended I/O daemon, voliod, does the following:

  - Allows for some extended I/O operations without blocking calling processes
  - Allows the volume device driver to schedule writes to volumes that have block-change logging enabled (described in Section 9.1.4)
  - Starts the LSM kernel error-log daemons to handle errors and recovery for LSM volumes

  If there are volumes with block-change logging enabled, then there will be multiple voliod processes running on the system. The voliod processes are started by the `vold` daemon and are killed by the kernel when these processes are no longer needed. Rebooting after your initial installation should start the `voliod` daemon.

For more detailed information about these daemons, refer to the `vold`(8), and `voliod`(8) reference pages.

1.11 LSM Encapsulation Tools

LSM provides a set of tools that you can use to migrate existing user data into LSM volumes, without physically moving the data. This process is referred to as encapsulation.

The LSM encapsulation process examines the UNIX device, LVM volume group, or AdvFS domain the user specifies as input, and generates files containing instructions that actually implement the encapsulation changes.

Refer to Chapter 3 for information about encapsulating user data on UNIX style partitions, LVM volume groups, or AdvFS storage domains.

LSM Configurations 2

LSM supports a variety of LSM configurations including concatenated disks, mirroring, striping, and multiple disk configurations. This chapter describes these configurations, and presents some options you should consider when planning your LSM configuration.

2.1 Planning an LSM Configuration

Prior to setting up LSM volumes, plexes, and subdisks, you should consider the needs of your site, the hardware available to you, and the rationale for creating volumes and disk groups.

The following table presents some configuration options and describes the planning considerations that apply to LSM configurations.

Table 2-1: LSM Configuration Considerations

Configuration	Description
Concatenated Volumes	You concatenate multiple LSM disks together to form a big volume. You can use a concatenated volume to store a large file or file systems that spans more than one disk. Disk concatenation frees you from being limited by the actual physical sizes of individual disks so that you can combine the storage potential of several devices. Use the default disk group, `rootdg`, to create a concatenated volume out of the public regions available. You can also add more LSM disks and create volumes out of the new disks you added.
Mirrored Volumes	You associate multiple plexes with the same volume to create a mirrored volume. If you are concerned about the availability of your data, then plan to mirror data on your system. You should map plexes that are associated with the same volume, to different physical disks. For systems with multiple disk controllers, it is best to map a volume's plexes to different controllers. The `volassist` command will fail if you specify a device that is already in the volume as the mirrored plex, whereas the bottom-up commands will not fail.

Table 2-1: (continued)

Configuration	Description
Striped Volumes	For faster read/write throughput, use a volume with a striped plex. On a physical disk drive, the drive performs only one I/O operation at a time. On an LSM volume with its data striped across multiple physical disks, multiple I/Os (one for each physical disk) can be performed simultaneously. The basic components of a striped plex is the size of the plex in multiples of the stripe width used, stripe width, and number of stripes. Stripe blocks of the stripe width size are interleaved among the subdisks resulting in an even distribution of accesses between the subdisks. The stripe width defaults to 128 sectors, but you can tune the size to specific application needs. The `volassist` command automatically rounds up the volume length to multiples of the stripe width.
Mirrored and Striped Volumes	Use mirrored and striped volumes when speed and availability are important. LSM supports mirroring of striped plexes. This configuration offers the improved I/O performance of striping while also providing data availability. Note that the different striped plexes in a mirrored volume do not have to be symmetrical. For instance, a three-way striped plex can be mirrored with a two-way striped plex, as long as the plex size is the same.

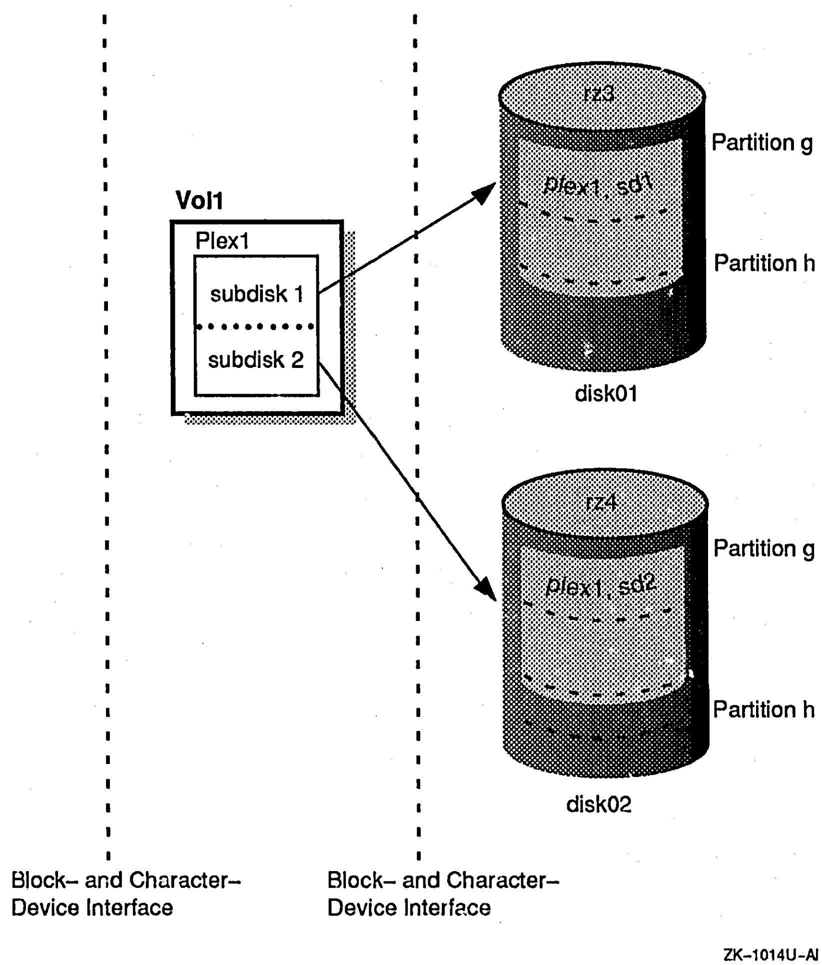
In addition, note that reads can be serviced by any plex in a mirrored volume. Thus, a mirrored volume provides increased read performance. However, LSM issues mirrored writes to all plexes in a mirrored volume. Because the writes are issued in parallel, there is a small amount of additional overhead as a result of write I/O to a mirrored volume.

2.2 Disk Concatenation

Disk concatenation involves arranging subdisks both sequentially and contiguously in the address space of a plex. With concatenation, subdisks are linked together into the logical address space. Data is then accessed from each of the subdisks in sequence.

Figure 2-1 gives an example of a concatenated disk.

Figure 2-1: Disk Concatenation Using LSM



Volumes that consist of one or more disks are *concatenated* volumes. Concatenated volumes that consist of subdisks from more than one disk are also referred to as *spanned* volumes because the volume spans multiple physical disks. The following list describes the advantages of using each of these types of volumes:

- Spanning is a special instance of concatenation that allows a particularly large file system or database to span multiple physical disks. Spanning takes advantage of the LSM virtual-disk environment to avoid the size limitations associated with physical disk size.

- Concatenation allows a volume to be created from multiple regions of one or more disks if there is not enough space for an entire volume on a single region of a disk.

2.3 Disk Striping

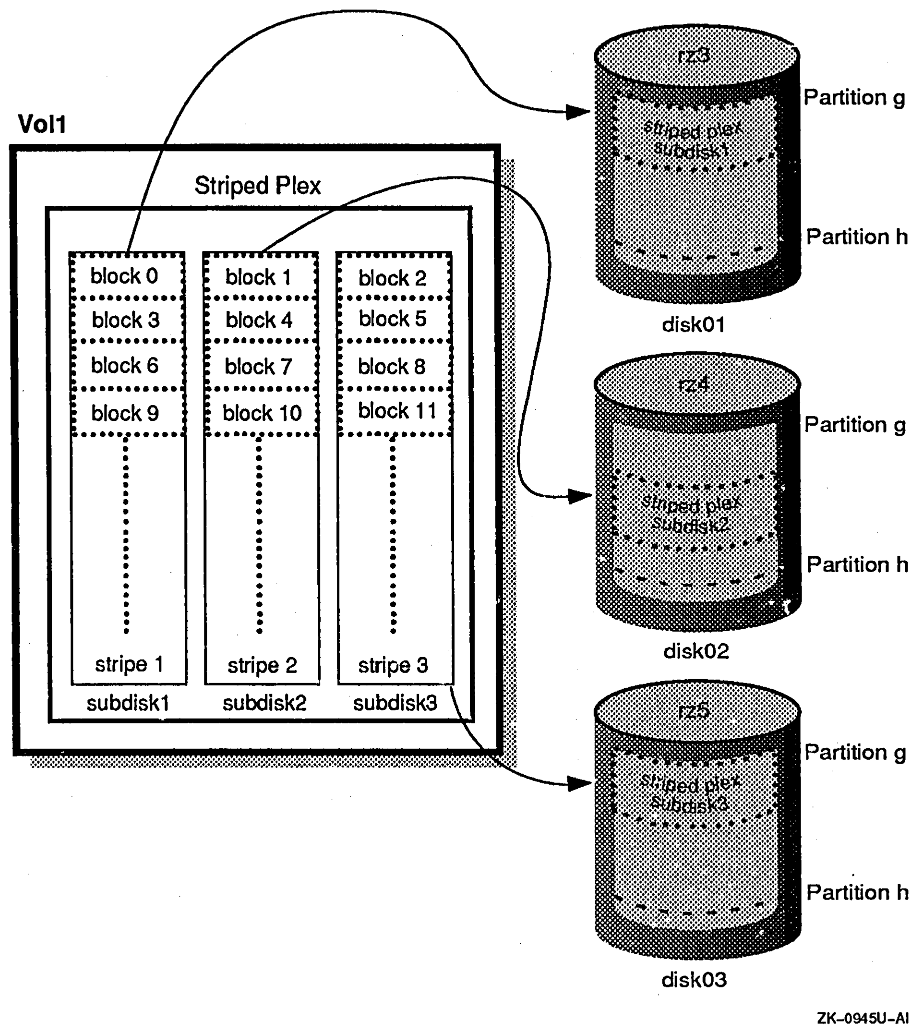
The manner in which storage space is allocated to file systems and databases has a direct impact on disk head movements and on the distribution of the I/O load between disk drives. An optimal allocation minimizes head movements and distributes the I/O load evenly between disk drives.

By supporting *striping* in addition to concatenation as a storage-allocation scheme for plexes, LSM makes it possible to evenly distribute the I/O load for a plex across a number of disk drives. Striping involves spreading data across several physical disks.

Stripes are relatively small, equally-sized fragments that are allocated alternately and evenly to the subdisks of a single plex. A striped plex consists of a number of equally-sized subdisks, each located on a separate disk drive. There should be at least two subdisks in a striped plex, each of which should exist on a different disk.

Data is stored on the subdisks in *stripe blocks* of a fixed size (referred to as the *stripe width*). Stripe blocks are interleaved between the subdisks as shown in Figure 2-2, resulting in an even distribution of accesses between the subdisks.

Figure 2-2: Striping Disks with the Logical Storage Manager



By allocating storage evenly across multiple disks, striping helps to balance I/O load in cases where high traffic areas exist on certain subdisks. Throughput increases with the number of disks across which a plex is striped. The increase in throughput depends on the applications and file systems being used, and on the number of users using them at the same time.

The effect of striping on performance is dependent on the choice of the stripe width and on application characteristics. LSM uses a default stripe width of 128 sectors that works well in most environments.

2.4 Mirroring Disks

In a system without LSM, failures of a physical disk result in loss of the data on that disk. To recover from such an event, the data needs to be restored from a backup and all changes made to the data since that backup have to be reapplied. This is a time-consuming process during which applications have no access to the data.

LSM makes it possible to protect critical data against disk failures by maintaining multiple copies (called mirrors) of the data in a volume. The LSM object that corresponds to a mirror is a *plex*. In the event of a physical disk failure, the plex on the failed disk becomes temporarily unavailable, but the system continues to operate using the unaffected plexes. Note the following rules when using LSM plexes to mirror disks:

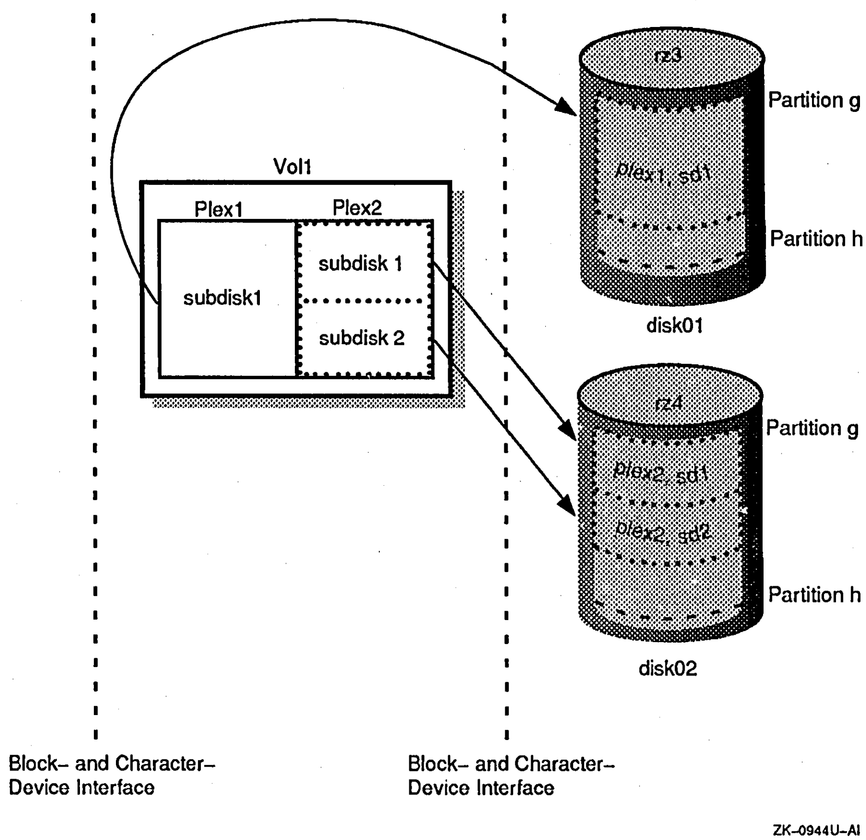
- Although a volume can have a single plex, at least two plexes are required to provide duplication of data. Each of these plexes should contain disk space from different disks for the redundancy to be effective. If possible, locate the plexes' disk space on different controllers for a higher level of availability and safety.
- When striping or spanning across a large number of disks, failure of any one of those disks will usually make the entire plex unusable. The chance of one out of several disks failing is enough to make it worthwhile to consider mirroring to improve the reliability of a striped, spanned, or concatenated volume.

All copies of a plex are kept up to date as updates are made to the contents of the volume. If a read to a plex fails, other plexes are used to correct or mask the error. Users of a volume are shielded against any failures unless all plexes fail.

If your applications perform an equal proportion of read and write operations, or if your applications perform more writes, you are not likely to gain performance (in fact, you might lose performance) if you mirror the data. However, if your files or applications perform significantly more read operations, you can improve performance with mirroring.

Figure 2-3 shows a mirrored LSM configuration.

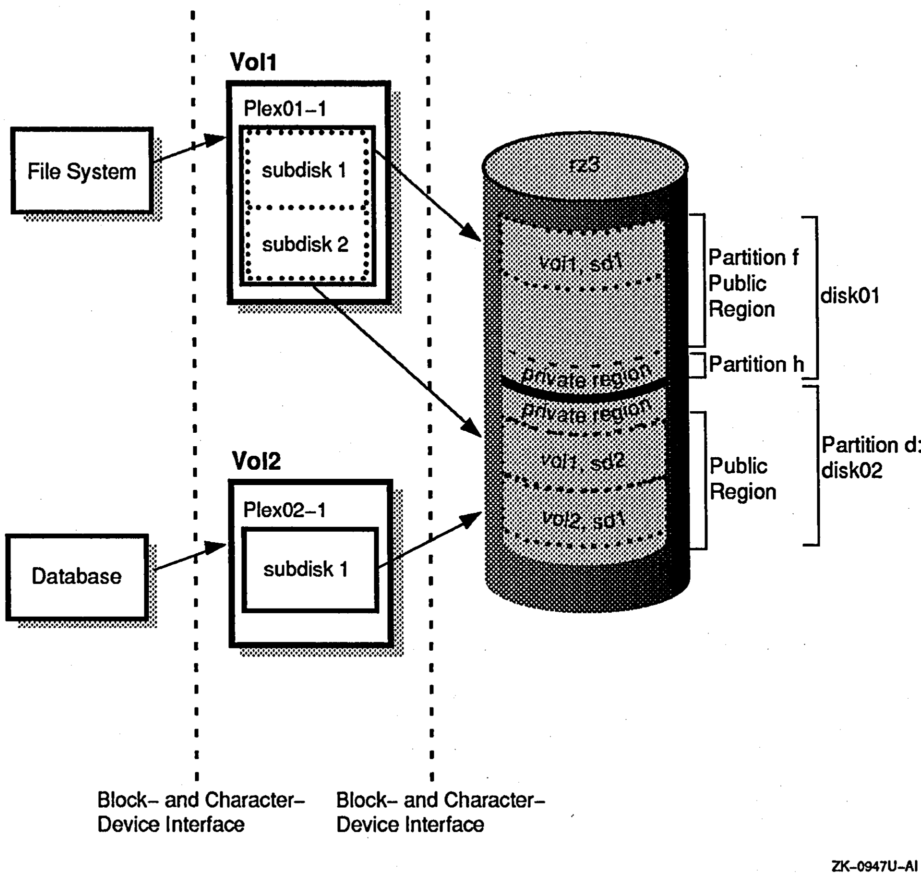
Figure 2-3: Mirroring Using LSM



2.5 Multiple Logical Devices per Physical Disk

With LSM, it is possible to create multiple LSM disk devices on a single-physical disk drive. These are created as nonoverlapping partitions that are addressed through their physical address `rz`*n* or a symbolic device name. For the LSM sliced disk `rz3` in Figure 2-4, the private region is in partition `h` and the public region is in partition `f`. For LSM simple disks, the private and public regions are both contained in a single partition `rz3d` (partition `d`). Figure 2-4 illustrates how multiple LSM disks are created on a single physical disk drive.

Figure 2-4: Multiple Devices per Physical Disk



2.6 Making LSM Configuration Changes

LSM permits dynamic reconfiguration of the volumes, making it easy to adapt to changes in I/O load and application needs, and to maximize system availability. LSM provides a high degree of flexibility in the way volumes can be mapped to disk and partition devices. This flexibility allows you to optimize performance, change volume size, add plexes, and perform backups or other administrative tasks without interrupting system applications and users. See also Section 9.3 for more information about implementing configuration changes.

Setting Up LSM 3

This chapter helps you start up the Logical Storage Manager (LSM), and encapsulate user data currently existing on UNIX style partitions, LVM volumes, and AdvFS storage domains. The topics covered in this chapter are as follows:

- Section 3.1 — Initial LSM Set Up
- Section 3.2 — Starting LSM
- Section 3.3 — Encapsulating existing user data to LSM volumes
- Section 3.4 — UNIX style-partition encapsulation
- Section 3.5 — LVM volume encapsulation
- Section 3.6 — AdvFS domain storage encapsulation

The examples in this chapter use the command line interface. Chapter 7 contains information about using the command line interface. See also Chapter 4 and Chapter 5 for information about the LSM graphical user interface (GUI), and Chapter 6 for information about the LSM Support Operations menu interface.

3.1 Initial LSM Set Up

The following sections describe running the `volinstall` procedure to create the special device files required by LSM, and running the `volsetup` procedure to perform initial LSM set up.

3.1.1 Running the volinstall Procedure

You can execute the `/sbin/volinstall` procedure to create the special device files for LSM daemons to communicate with the LSM kernel components. In addition, `/sbin/volinstall` optionally sets up the system for automatic LSM restart after a reboot.

Enter the following command to run /sbin/volinstall:

```
# volinstall
```

During the volinstall procedure, you are asked if you want to have LSM automatically started whenever the system booted. If you choose to have LSM automatically started, the volinstall procedure adds the following line to the /etc/inittab file:

```
lsm:23:wait/sbin/lsmbstartup</dev/console>/dev/console 2>&1
```

If you want to have LSM volumes mounted in /etc/fstab, you must use the automatic restart method because the volumes must be accessible before multiuser mode. For manual restart, see Section 3.2.1.

3.1.2 Running the volsetup Procedure

Warning

> Skip this section if you are already running LSM and the rootdg disk group is already initialized. For example, after an upgrade installation from DEC OSF/1 Version 2.0, you must skip this step.

To perform an initial LSM set up that includes creating the rootdg disk group, starting the LSM daemon processes, and adding devices to the disk group, execute the /sbin/volsetup interactive utility. To do this, enter the following command:

```
# /sbin/volsetup rz0
```

In the above example, the device rz0 is the disk to be added to LSM.

Warning

> Invoke the volsetup utility only once. To later add more disks, use the voldiskadd utility (as described in Section 7.2.1).

After running the volsetup utility, LSM creates the /etc/vol/volboot file. This file is used during system start up to locate copies of the rootdg disk group configuration. Do not delete the /etc/vol/volboot file; it is critical for starting LSM.

The volsetup procedure also starts LSM. After running the volsetup procedure, skip to Section 3.2.2.

3.2 Starting LSM

The following sections describe how to start LSM manually, and ensure that the volume daemon processes are enabled and running.

3.2.1 Manually Starting LSM

If you just ran the `volsetup` procedure described in Section 3.1.2, or if you chose the automatic LSM startup option during the `volinstall` procedure (described in Section 3.1), LSM should have started already. If LSM is started, you can skip to Section 3.2.2. Otherwise, for manual restart, you must:

1. Start the error daemons by entering:

   ```
   # voliod set 2
   ```

2. Start up `vold` in enabled mode by entering:

   ```
   # vold
   ```

3. Enable the LSM volumes by entering:

   ```
   # volrecover -sb
   ```

3.2.2 Ensuring the Volume Configuration Daemon (vold) is Running

The `vold` daemon must be running before any LSM operations can be performed. Typically, the `vold` daemon is configured to start automatically during the reboot procedure. Perform the following steps to determine the state of the volume daemon:

1. Determine if the volume daemon is running and enabled by entering the following command:

   voldctl mode

IF...	THEN...
The `vold` daemon is both running and enabled	The following message displays: **mode:enabled**
The `vold` daemon is running, but is not enabled	The following message displays: **mode:disabled**

IF...	THEN...
The vold daemon is not running	The following message displays: **mode:not-running**

2. If necessary, enable the volume daemon by entering the following command:

   voldctl enable

3. If necessary, start the volume daemon by entering the following command:

   vold

For additional information about the vold daemon, refer to the vold(8) reference page.

3.2.3 Ensuring the Volume Extended I/O Daemon (voliod) is Running

Volume log I/O voliod processes are started by the vold daemon (if block-change logging is enabled) and are killed by the kernel when these processes are no longer needed. Volume error daemons are started automatically by LSM startup procedures. Rebooting after your initial installation should start the voliod error daemon automatically. You can perform these steps to determine the state of the error daemon:

1. Verify that the error daemon is running and enabled by entering the following command:

   voliod

IF...	THEN...
Any voliod processes are running	The following message displays: **# volume I/O daemons running** The # symbol in the previous example indicates the number of voliod daemons running.

IF...	THEN...
There are no `voliod` daemons currently running	Start some daemons by entering the following command: **voliod set 2**

2. If necessary, enable the volume daemon by entering the following command:

   ```
   voliod set 2
   ```

For more detailed information about the `voliod` daemon, refer to the `voliod`(8) reference page.

3.3 Encapsulating Existing User Data to LSM Volumes

Prior to DEC OSF/1 Version 2.0, disk management was performed using the following tools:

- DEC OSF/1 Logical Volume Manager (LVM)
- DEC OSF/1 port of ULTRIX Disk Shadowing (UDS)
- DEC OSF/1 port of ULTRIX Striping Driver

The DEC OSF/1 Logical Storage Manager (LSM) is a single integrated component intended to replace LVM, UDS, and the striping driver.

LSM migrates or encapsulates existing user data to LSM volumes using a process called encapsulation. LSM supports data encapsulation from the following formats:

- UNIX File System (UFS) data partitions
- LVM volumes
- Advanced File System (AdvFS) storage domains

The following sections describe how to encapsulate existing user data to LSM volumes.

Note

LSM does not support encapsulation of user data on:

- Stripe volumes
- UDS volumes

3.3.1 Encapsulation Process

During the encapsulation process, LSM transforms a UNIX style disk or disk partition, an LVM volume group, or an AdvFS storage domain into an LSM logical disk. Using a physical device name that you supply in an encapsulation command, LSM identifies how the device can be used for file systems and generates LSM volumes to cover those areas on the disk.

The encapsulation process varies depending on whether you are migrating from UNIX, LVM, or AdvFS. The commands shown in the following list help you perform a one-time conversion of existing user data into LSM volumes:

- Migration to LSM from UNIX physical devices or disk partitions (for example, a UFS file system residing on a partition or database data in a partition) is invoked by the `volencap` command. This is described in Section 3.4.
- Migration of LVM volume groups to LSM is invoked by the `vollvmencap` command. This is described in Section 3.5.
- Migration of AdvFS storage domains to LSM is invoked by the `voladvdomencap` command. This is described in Section 3.6.

In addition to the section references given in the list, you should also refer to the reference pages for the `volencap`(8), `vollvmencap`(8) , and `voladvdomencap`(8) functions to obtain detailed command syntax and usage.

3.3.2 Encapsulation Requirements

The following list describes requirements for performing encapsulation functions:

- Perform encapsulation when data is off line.

  To minimize the risk of configuration changes during encapsulation, ensure the data remains off line during the encapsulation process.
- Back up all user data before beginning the encapsulation process.

  The encapsulation process is not atomic. In the event that a failure occurs during encapsulation, you can restore the saved data to return data to its original state.
- The rootdg disk group must exist and be enabled.

  Some configurations require that the encapsulation process creates an LSM nopriv disk. This type of encapsulation uses `rootdg` to store the configuration data during encapsulation.

3.4 UNIX Style Partition Encapsulation

The encapsulated process for UNIX style disks and disk partitions uses a physical disk name or a partition name that you specify with the `volencap` command, and transforms the disk or partition into an LSM disk.

The `volencap` command automatically encapsulates user data for common configuration layouts such as the following:

- An empty physical device.
- An empty partition.
- An entire physical device where one or more partitions are in use. Each partition that is in use will be encapsulated into a volume.
- A partition that is in use on a physical device.

Section 3.4.1 provides an overview of UNIX disk partitions. Section 3.4.2 describes the automatic encapsulation process. However, if a finer degree of control is desired, use the manual encapsulation procedure to tailor the encapsulation to the specific needs of your configuration. See Section 3.4.3 for information to help you encapsulate UNIX style partitions manually.

3.4.1 Overview of DEC OSF/1 (UNIX) Partitions

This section provides an overview of UNIX style devices and partitions and how partitions are mapped by a partition table. This background information is provided to help aid your understanding of how the user data on these devices is encapsulated to LSM volumes.

The partitions on a physical device are mapped by a partition table called disklabel. The disk's partitions and disklabel have the following characteristics:

- The disklabel maps a physical disk to a maximum of eight partitions that are labeled from a to h.
- The partitions can overlap each other.
- The disklabel resides at block 0 of a physical device. For most devices, the default disklabel is defined in `/etc/disktab`.
- You can modify an online disklabel using the utility program `/usr/sbin/disklabel`.

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

3.4.2 UNIX Encapsulation Overview

You supply the name of a physical device (for example, rz3) or the partition name (for example, rz3g) as input to the `volencap` command to begin the encapsulation process. For example:

```
/usr/sbin/volencap rz3.
```

The LSM encapsulation process uses information in the disklabel and the `/etc/fstab` file to find out if a partition is in use. For example, a common usage of a partition is to store user data in the UFS file system. When a UFS file system is installed on a given partition by entering the `/usr/sbin/newfs` command, a file system is created that spans the entire partition. Even if there is no data in the file system, the partition storing the file system is considered to be full. A database application fills a partition in a similar fashion. If the partition does not have information in a disklabel or `/etc/fstab` file to indicate that it is being used by an application, the partition must be encapsulated using the partition name.

See the `volencap`(8) reference page for complete details of the command and its options.

The `/usr/sbin/volencap` command generates the necessary LSM commands and files to create LSM volumes. The LSM commands are run when `/sbin/vol-reconfig` is executed. The `/etc/fstab` file is updated on successful creation of LSM volumes containing UFS file systems.

The user can edit the `/etc/inittab` file to add `/sbin/vol-reconfig` so that when the system is rebooted, the encapsulation commands generated by `/usr/sbin/volencap` take effect. This is done by adding `/sbin/vol-reconfig` after `/sbin/lsmbstartup`, but before `/sbin/bcheckrc`, as shown in the following example:

```
lsm:23:wait:/sbin/lsmbstartup < /dev/console > /dev/console 2>&1
volencap:23:wait:/sbin/vol-reconfig< /dev/console >/dev/console 2>&1
fs:23:wait:/sbin/bcheckrc < /dev/console > /dev/console 2>&1
```

You should use the above method if any disk or partition that was encapsulated was in use.

The second option is for the user to execute `/sbin/vol-reconfig` after all disks and partitions have been encapsulated using the `/usr/sbin/volencap` command. If any partition or disk that has been encapsulated is still in use, reboot the system.

The results of the encapsulation process are as follows:

- If the encapsulation process is performed on a disk partition, then LSM encapsulates the partition as a *nopriv* disk because there is no free space available to store the configuration database. A nopriv disk can be added to a disk group only if other disks in the disk group have a defined metadata region.

- If the encapsulation process is performed on a physical disk, then LSM adds the physical disk to the LSM environment as a sliced disk when the following conditions are true:
  - If all user data and, hence, the public region of the LSM sliced disk can be defined as a contiguous region.
  - If at least two partition entries are available in the disklabel to identify the public and the private regions of a sliced LSM disk.
  - If there is room for a small private region at the beginning or end of the disk.

  If the disk cannot be encapsulated as a sliced disk, LSM adds the physical disk to the LSM environment as a nopriv disk.

Once a disk or partition has been converted to either a sliced or nopriv LSM disk:

1. LSM converts each partition that is in use (for example, a UFS file system) to a subdisk.
2. LSM creates a plex using the subdisk and in turn generates an LSM volume using the plex.
3. After the data partition is encapsulated into an LSM volume, the procedure changes all the associated entries in the `/etc/fstab` file to use the LSM volume name instead of the block device name of the physical disk partition.

Note

Block 0 on a DEC OSF/1 disk device is read-only by default. UFS does not use block 0 when putting data on device partitions. To preserve the LBN mapping, the LSM nopriv disk must start at LBN block 0. As long as this disk is used for UFS volumes, this does not present a problem. However, if the disk is reused for other applications which write to block 0, then a write failure will occur. To help avoid such failures, the LSM nopriv disk is labeled with the unique administration name *device-name*`_blk0`. You should remove this nopriv disk from the LSM disk group and redefine it without block 0 if the volume is no longer needed.

3.4.3 UNIX Manual Encapsulation

This section describes how to manually encapsulate UNIX style user disks to generate LSM volumes. You should perform a manual encapsulation only when the automatic volencap encapsulation process does not apply to your configuration. The following instructions describe the manual process in a step-by-step format:

1. Before beginning the encapsulation process, ensure that all the disks or partitions that you intend to encapsulate are not in use. If a partition is currently mounted, it should be unmounted.
2. Save the original `/etc/fstab` file. Change the file to use the LSM volume names instead of the partition names.
3. Make sure the `rootdg` disk group exists and is active.
4. The following list describes the steps to encapsulate a partition:
   a. Encapsulate the `/dev/rz3h` partition that is being used as `/usr/staff`. The following examples shows how one of the lines appears in the `/etc/fstab` file:

   ```
   /dev/rz3h        /usr/staff    ufs rw 1 2
   ```

   b. Add the rz3h partition as a nopriv LSM disk. For example:

   ```
   voldisk -f init rz3h type=nopriv
   ```

   c. Add the rz3h partition to a disk group using the following instructions:

      - To add rz3h to rootdg, enter the following command:

        ```
        voldg -g rootdg adddisk rz3h
        ```

      - To add the rz3h partition to the existing disk group dg1, enter the following command:

        ```
        voldg -g dg1 adddisk rz3h
        ```

   d. Using the information in disklabel, find the size of partition `h`. Use the following command to display the disklabel information for rz3:

   ```
   # disklabel -r rz3:
   #   size    offset fstype [fsize bsize cpg]
   a:  131072       0 4.2BSD 1024 8192 16 # (Cyl.    0 - 164*
   b:  262144  131072 unused 1024 8192    # (Cyl.  164*- 492*)
   c: 2050860       0 unused 1024 8192    # (Cyl.    0 - 2569)
   d:  552548  393216 unused 1024 8192    # (Cyl.  492*- 1185*)
   e:  552548  945764 unused 1024 8192    # (Cyl. 1185*- 1877*)
   f:  552548 1498312 unused 1024 8192    # (Cyl. 1877*- 2569*)
   g:  819200  393216 4.2BSD 1024 8192 16 # (Cyl.  492*- 1519*)
   h:  838444 1212416 4.2BSD 1024 8192 16 # (Cyl. 1519*- 2569*)
   ```

   The example output shows that the size of partition `h` is 838444 sectors.

   e. Find a unique name for the LSM volume corresponding to partition rz3h. Use a volume name that uses the partition name (for example, vol-rz3h).

f. Create an LSM volume. Provide the correct disk group name. The following example uses rootdg as the disk group name:

```
# /sbin/volassist -g rootdg  make vol-rz3h 838444s rz3h
```

g. Change the `/etc/fstab` file to look as follows:

```
# /dev/vol/rootdg/vol-rz3h        /usr/staff    ufs rw 1 2
```

5. Encapsulate a complete disk to LSM and convert partitions that are in use into LSM volumes.

   In the following example for the disk rz4, two partitions store user data by means of a UFS file system, and one partition allows applications to directly access and store user data using the device interface.

   The `/etc/fstab` file includes the following lines that correspond to the rz4 disk:

```
/dev/rz4d        /data1    ufs rw 1 2
/dev/rz4e        /data2    ufs rw 1 2
```

   Partition rz4f is used by applications to store user data directly using the device interface. The following example shows the command and display for the disklabel information for the rz4 disk:

```
# disklabel -r rz4
#      size  offset  fstype [fsize bsize cpg]
a:  131072       0  unused  1024 8192 16 # (Cyl.    0 - 164*
b:  262144  131072  unused  1024 8192    # (Cyl.  164*- 492*)
c: 2050860       0  unused  1024 8192    # (Cyl.    0 - 2569)
d:  552548  393216  4.2BSD  1024 8192    # (Cyl.  492*- 1185*)
e:  552548  945764  4.2BSD  1024 8192    # (Cyl. 1185*- 1877*)
f:  552548 1498312  unused  1024 8192    # (Cyl. 1877*- 2569*)
g:  819200  393216  unused  1024 8192 16 # (Cyl.  492*- 1519*)
h:  838444 1212416  unused  1024 8192 16 # (Cyl. 1519*- 2569*)
#
```

6. Edit the disklabel before beginning the encapsulation process.

   In the example for the rz4 disk, the disklabel output shows that partitions `d` and `e` are in use for UFS file system. The f partition is marked as unused even though it is in use. Applications that access data directly on partitions using the device interface must edit the disklabel using the command `disklabel -e` to change the disklabel before encapsulation. Because no appropriate disklabel tags are provided, use of disklabel tag 4.1BSD (shown in the following example) is suggested:

```
#      size  offset fstype [fsize bsize cpg]
a:  131072       0 unused 1024 8192 16 # (Cyl.    0 - 164*
b:  262144  131072 unused 1024 8192    # (Cyl.  164*- 492*)
c: 2050860       0 unused 1024 8192    # (Cyl.    0 - 2569)
d:  552548  393216 4.2BSD 1024 8192    # (Cyl.  492*- 1185*)
e:  552548  945764 4.2BSD 1024 8192    # (Cyl. 1185*- 1877*)
f:  552548 1498312 4.1BSD 1024 8192    # (Cyl. 1877*- 2569*)
```

```
g:  819200  393216 unused 1024 8192 16 # (Cyl.  492*- 1519*)
h:  838444 1212416 unused 1024 8192 16 # (Cyl. 1519*- 2569*)
```

7. Find two partitions that are free and can be used to add rz4 as an LSM sliced disk. In the example, the disklabel output shows that partitions a, b, c, g, and h are possible candidates. Because a sliced LSM disk can be created only when 512 sectors are free either at the beginning or at the end of the disk, partitions g and h were chosen. The display indicates that these partitions have 512 sectors free at the beginning of the disk and at the end of the disk.

   Use partition g to store the offset and size for the public region of the LSM disk and partition h will be used to store the offset and size for the private region of the LSM disk.

   If there are no partitions with adequate free sectors, you must encapsulate rz4c as a nopriv disk.

   Note

   Block 0 on a disk is write-locked. Therefore, block 0 should not be used either for the public or private region of the disk.

8. Edit the disk label to change partitions g and h, and verify the changes according to the following instructions:

   a. Change partition h to be 512 sectors and have a disklabel tag of LSMpriv. Partition h will start at an offset of 512 sectors from the end of the disk.

   b. Start partition g at block 1 and extend it to the beginning of partition h.

      Change the disklabel tag for partition g to LSMpubl.

   The following display shows the disklabel after these changes have been made:

```
#      size   offset fstype [fsize bsize cpg]
a:  131072        0 unused  1024 8192 16  # (Cyl.    0 - 164*
b:  262144   131072 unused  1024 8192     # (Cyl.  164*- 492*)
c: 2050860        0 unused  1024 8192     # (Cyl.    0 - 2569)
d:  552548   393216 4.2BSD  1024 8192     # (Cyl.  492*- 1185*)
e:  552548   945764 4.2BSD  1024 8192     # (Cyl. 1185*- 1877*)
f:  552548  1498312 4.1BSD  1024 8192     # (Cyl. 1877*- 2569*)
g: 2050347        1 LSMpub  1024 8192 16  # (Cyl.  492*- 1519*)
h:  512     2050358 LSMpri  1024 8192 16  # (Cyl. 1519*- 2569*)
```

9. Initialize rz4 and add it to the rootdg disk group as follows:

a. For a LSM sliced disk, enter the following:

```
# voldisk -f init rz4 type=sliced
```

b. For a LSM nopriv disk, enter the following:

```
# voldisk -f init rz4c type=nopriv
```

10. Add LSM disk to `rootdg`, as follows:

a. For an LSM sliced disk, enter the following:

```
# voldg -g adddisk rz4
```

b. For a LSM nopriv disk, enter the following:

```
# voldg -g adddisk rz4c
```

11. Create LSM volumes for all partitions in use.

a. For a LSM sliced disk, perform the following:

1. To convert partition `d`, specify the starting offset of partition d in the public region. Because the public region starts at block 1, subtract 1 from the offset. (Note that the calculation differs when the private region starts at the beginning of the disk.) The following example shows the conversion calculation for partition `d`:

```
[ partition d offset - 1 ] or [ 393216 - 1 ]
```

Enter the following command for partition `d`:

```
# volassist make vol-rz4-01 552548 rz4,393215
```

2. To convert partition `e`, specify the starting offset [945764 - 1] in the public region. For example:

```
# volassist make vol-rz4-02 552548 rz4,945763
```

3. To convert partition `f`, specify the starting offset [1498312 - 1] in the public region. For example:

```
# volassist make vol-rz4-03 552548 rz4,1498311
```

b. For a LSM nopriv disk, the calculation for the partition offset does not need to be changed because nopriv disks do not contain any metadata. The LSM disk partitions are as follows:

- For partition `d`, enter the following:

```
# volassist make vol-rz4-01 552548 rz4,393216
```

- For partition e, enter the following:

  ```
  # volassist make vol-rz4-02 552548 rz4,945764
  ```

- For partition g, enter the following:

  ```
  # volassist make vol-rz4-03 552548 rz4,1498312
  ```

12. Change the /etc/fstab file as follows:

  - For an LSM sliced disk, enter the following:

    ```
    /dev/vol/rootdg/vol-rz4-01      /data1    ufs rw 1 2
    /dev/vol/rootdg/vol-rz4-02      /data2    ufs rw 1 2
    ```

  - For an LSM nopriv disk, enter the following:

    ```
    /dev/vol/rootdg/vol-rz4c-01     /data1    ufs rw 1 2
    /dev/vol/rootdg/vol-rz4c-02     /data2    ufs rw 1 2
    ```

  Applications that were using /dev/rrz4f should now use /dev/rvol/rootdg/vol-rz4-03.

3.5 LVM Volume Encapsulation

The LVM encapsulated process uses the name of a volume group that you specify with the vollvmencap command, and transforms the LVM volumes into LSM volumes.

The automatic encapsulation process is described in Section 3.5.2. Note that there is no manual encapsulation process for LVM volume groups.

3.5.1 Overview of LVM Support in DEC OSF/1

This section provides an overview of LVM volume groups. This background information is provided to help aid your understanding of how user data on LVM devices is encapsulated to LSM volumes.

Encapsulation of LVM volumes is based on volume groups. A volume group is a collection of physical volumes, each of which contains:

- LVM metadata area
- LVM data area
- LVM bad block replacement directory

In addition, the LVM data area is divided into physical extents, which are the basic building blocks of LVM volumes. The physical extents for all physical volumes in a volume group are all the same size.

Finally, a volume consists of a series of logical extents, each of which maps to one or more physical extents. Because DEC OSF/1 does not support mirroring, each logical extent can map to only one physical extent except in the occasional event when LVM requires temporary mirrors to be added to a volume for the duration of the command execution. For encapsulation purposes, these transient conditions are not considered.

Note

> For DEC OSF/1 systems, the physical extent bad block directory is not used.

The `/etc/lvmtab` file defines all the volume groups and their associated physical volumes on a system. When a system reboots, LVM restarts based on the information defined in this file.

There is an LVM record at the beginning of each physical volume in a volume group. The LVM record contains a number, and the location and length of the metadata region. In the metadata region, there are entries for each logical volume defined in the volume group, and mappings of each physical extent to logical extents of the physical volumes.

A typical LVM configuration has a few physical device partitions in a volume group. An arbitrary number of volumes is defined by mapping logical extents to physical extents in a volume group. These volumes are used for UFS file systems or user data can also be accessed directly through the device interface.

3.5.2 LVM Encapsulation Overview

You supply the name of an LVM volume group as input to the `vollvmencap` command to begin the encapsulation process. For example:

```
/usr/sbin/vollvmencap /dev/vg1
```

See the `vollvmencap`(8) reference page for complete details about the command and its options.

The `/usr/sbin/vollvmencap` command generates LSM commands and files to create LSM volumes. The LSM commands are run when `/sbin/vol-lvm-reconfig` is executed. This executes all LSM commands that were generated during the `/usr/sbin/vollvmencap` command executed the encapsulation process. When the encapsulation is successful, a message is printed that indicates a name of a script that you must execute to remove LVM volumes. Run this script only after ensuring that the encapsulation was successful. Note that all LVM volumes in the volume group that was encapsulated should not be in use when `/sbin/vol-lvm-reconfig` is executed. The `/etc/fstab` file is updated when the LSM volumes are successfully created.

Note the following requirements for the LVM encapsulation process:

- The entire volume group must be encapsulated at the same time.
- The volume group configuration must be static for the duration of the encapsulation process.
- The LVM volume group can be accessed using the `vgchange` command.
- A volume group can be encapsulated into the default LSM disk group `rootdg`. (The `rootdg` disk group must exist before the encapsulation process begins.)
- A volume group can be encapsulated into another disk group if the disk group already exists.

The encapsulation process involves creation of an LSM subdisk for contiguous physical extents in physical volume mappings to a logical volume. Because LVM volumes in DEC OSF/1 are not mirrored, the LSM volume has only one plex. The plex consists of a set of subdisks obtained by mapping physical extents associated with each logical extent. The plex is used to create an LSM volume. The LSM volume name replaces the LVM volume name in the `/etc/fstab` file.

Note

> Block 0 on a DEC OSF/1 disk device is read-only by default. UFS does not use block 0 when putting data on device partitions. To preserve the LBN mapping, the LSM nopriv disk must start at LBN block 0. As long as this disk is used for UFS volumes, this does not present a problem. However, if the disk is reused for other applications which write to block 0, then a write failure will occur. To help avoid such failures, the LSM nopriv disk is labeled with the unique administration name *device-name*_blk0. You should remove this nopriv disk from the LSM disk group and redefine it without block 0 if the volume is no longer needed.

3.6 AdvFS Domain Storage Encapsulation

Encapsulation of AdvFS user data is at the storage domain level. The entire AdvFS domain must be encapsulated together. 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.

For most configurations, you can encapsulate user data automatically using the `voladvdomencap` command. However, if a finer degree of control is desired, use the manual encapsulation procedure to tailor the encapsulation to the specific needs of your configuration. Section 3.6.3 provides instructions to help you encapsulate AdvFS domains manually.

The following sections provide an overview of AdvFS support and describe the procedure for encapsulating AdvFS storage domains.

3.6.1 Overview of AdvFS Support on DEC OSF/1

This section provides an overview of AdvFS storage domains. This background information is provided to help aid your understanding of how user data from AdvFS domains is encapsulated to LSM volumes.

An AdvFS domain is a single-storage container consisting of one or more physical disk partitions. File systems, called filesets, are created and defined in the domain and can expand and contract within the domain if space is available. Storage devices can be added or removed from a domain even when filesets are mounted. Active filesets and domains are determined as follows:

- A fileset is considered active when it is mounted at a file system mount point.
- A domain is active if one or more of its filesets are active.

Storage devices can be physical devices or logical volumes. Each domain has a directory tree in the `/etc/fdmns` file that describes the physical disk partitions that constitute the storage container. The root of the tree is the domain and each leaf node in the domain directory tree is a physical disk partition name. This is a soft link of the full-access path of the physical disk partition.

A typical system has a number of domains using a few physical devices as storage in each domain, and with many filesets created on these domains.

3.6.2 AdvFS Encapsulation Overview

The goal of the encapsulation is to capture the data in physical disk partitions of a domain into LSM volumes, and present the same data access to AdvFS by changing the soft links in the domain directory 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. Note that because an LSM nopriv disk does not require a disklabel tag, the physical device partitions retain the same disklabel usage tag after the encapsulation.

Note

Block 0 on a DEC OSF/1 disk device is read-only by default. AdvFS does not use block 0 when putting data on device partitions. To preserve the LBN mapping, the LSM nopriv disk must start at LBN block 0. As long as this disk is used for AdvFS volumes, this does not present a problem. However, if the disk is reused for other applications which write to block 0, then a write failure will occur. To help avoid such failures, the LSM nopriv disk is labeled with the unique administration name `advfs` *`device-name`*. You should remove this nopriv disk from the LSM disk group and redefine it without block 0 if the volume is no longer needed.

No mount point changes are necessary during encapsulation, because the filesets that are mounted 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 users of the AdvFS domain.

You supply the name of a domain as input to the `voladvdomencap` command to begin the encapsulation process. For example:

```
/usr/sbin/voladvdomencap dom1
```

See the `voladvdomencap`(8) Reference Page for complete details about the command and its options. The `/usr/sbin/voladvdomencap` command generates LSM commands and files to create LSM volumes. The LSM commands are run when `/sbin/vol-reconfig` is executed. The `/etc/fstab` file is updated on successful creation of LSM volumes.

The user can edit the `/etc/inittab` file to add `/sbin/vol-reconfig` so that when the system is rebooted the encapsulation commands generated by the `/usr/sbin/voladvdomencap` command take effect. This is done by adding `/sbin/vol-reconfig` after `/sbin/lsmbstartup` but before `/sbin/bcheckrc` as shown in the following example:

```
lsm:23:wait:/sbin/lsmbstartup < /dev/console > /dev/console 2>&1
volencap:23:wait:/sbin/vol-reconfig< /dev/console >/dev/console 2>&1
fs:23:wait:/sbin/bcheckrc < /dev/console > /dev/console 2>&1
```

Another option is to execute `/sbin/vol-reconfig` after all domains have been encapsulated using the `/usr/sbin/voladvdomencap` command. The domain should not be in use when `/sbin/vol-reconfig` is run. All filesets in the domain should be unmounted.

3.6.3 AdvFS Manual Encapsulation

This section describes how to manually encapsulate AdvFS storage domains to generate LSM volumes. You should perform a manual encapsulation only when the automatic `voladvdomencap` encapsulation process does not work for your configuration. The following instructions describe the manual process in a step-by-step format:

1. Check the AdvFS domain by entering the AdvFS domain inquiry command showfdmn on the domain that is to be encapsulated. You should encapsulate the domain only if the AdvFS indicates the domain is inactive. The following list shows the possible outcome of the showfdmn command:

   a. If the domain does not exist, AdvFS returns the following error message:

   ```
   # showfdmn dom2
   showfdmn: unable to get info for domain 'dom2'
   showfdmn: error = No such file or directory
   ```

   b. If the domain is active, AdvFS displays the following information:

   ```
   # showfdmn dom2
   Id                  Date Created              LogPgs  Domain Name
   2d2b5782.0009cca0  Wed Jan  5 19:12:50 1994    512   dom2
    Vol   512-Blks       Free   % Used  Cmode  Rblks  Wblks  Vol Name
    1L    1024000     1015408      1%     on    256    256  /dev/rz3c
   #
   ```

   c. If the domain is inactive, AdvFS returns the following information:

   ```
   # showfdmn dom2
   Id                  Date Created               LogPgs  Domain Name
   2d2b5782.0009cca0  Wed Jan  5 19:12:50 1994      512  dom2
   showfdmn: unable to display volume info; domain not active
   ```

2. Check the LSM disk group by entering the LSM disk group inquiry command `voldg list` on the target disk group. For example, to encapsulate the AdvFS storage domain into the disk group dg1, check that rootdg is enabled as shown in the following example:

   ```
   # voldg list
   NAME          STATE     ID
   rootdg        enabled   761416202.1025.chowfun.zk3.dec.com
   dg1           enabled   761416202.1034.chowfun.zk3.dec.com
   ```

3. Save the following information in the event that a recovery is needed:

   a. Save the LSM disk group name in the `/etc/vol/reconfig.d/domain.d/domain_name.d` directory. For example, before encapsulating the AdvFS domain dom2, enter the following commands:

```
# mkdir -p /etc/vol/reconfig.d/domain.d/dom2.d
# echo "dg1" > /etc/vol/reconfig.d/domain.d/dom2.d/dg
# cat /etc/vol/reconfig.d/domain.d/dom2.d/dg
# dg1
```

b. Save the domain directory tree to the `/etc/vol/reconfig.d/domain.d/domain_name.d` directory. For example:

```
# cp -R /etc/fdmns/dom2

#  /etc/vol/reconfig.d/domain.d/dom2.d
```

4. Encapsulate the physical devices of the AdvFS domain into the target disk group. For example:

```
# ls -R /etc/fdmns/dom2
rz3c    rz16g
# voldisk -f init rz3c type=nopriv
# voldisk -f init rz16g type=nopriv
# voldg -g dg1 adddisk advfs_rz3c=rz3c advfs_rz16g=rz16c
```

5. Define volumes to represent the user data. For example:

```
# volprint -g dg1 -F "%len" -d advfs_rz3c
4109967
# volprint -g dg1 -F "%len" -d advfs_rz16g
301986
# volassist -g dg1 make vol_rz3c 4109967 advfs_rz3c
# volassist -g dg1 make vol_rz16g 301986 advfs_rz16g
```

6. Change the AdvFS soft links.   For example:

```
# ln -sf /dev/vol/dg1/vol_rz3c /etc/fdmns/dom2/rz3c
# ln -sf /dev/vol/dg1/vol_rz16g /etc/fdmns/dom2/rz16g
```

7. Once the encapsulation is complete, mount filesets using their regular names, as shown:

```
# mount -t advfs dom2#fset2 /mnt
#
```

8. If the encapsulation fails, try to recover the domain by restoring the soft links. For example:

```
# cp -R /etc/vol/reconfig.d/domain.d/dom2.d/dom2 /etc/fdmns
```

3.6.4 Using voldisk for Special Encapsulations

In some cases, you may want to encapsulate a disk that does not have any data that can be used for LSM private region partition. The normal disk encapsulation procedure using the `volencap` utility (see Section 3.3 and the volencap(8) reference page) requires that some space be available at the beginning or end of the disk for storing LSM identification and configuration information.

The voldisk utility can be used to encapsulate disks that do not have available space. This is done using special types of disk devices, called nopriv devices, that do not have private regions. To use this, create a partition on the disk device (see disklabel(8)) that maps all parts of the disk that you want to be able to access, then add the partition device for that partition with the command:

voldisk define *partition-device* **type=nopriv**

Here, *partition-device* is the basename of the device in the /dev directory. For example, to use partition h of disk device rz3, use the command:

voldisk define rz3h type=nopriv

To create volumes for other partitions on the disk drive, add the device to a disk group, figure out where those partitions reside within the encapsulation partition, then use volassist to create a volume with that offset and length.

A major drawback with using these special encapsulation partition devices is that LSM cannot track changes in the address or controller of the disk. Normally, LSM uses identifying information stored on the physical disk to track changes in the location of a physical disk. Because nopriv devices do not have identifying information stored on the physical disk, this cannot occur.

The best use of special encapsulation partition devices is to encapsulate a disk so that LSM can be used to move space off of the disk. When space is made available at the beginning or end of the disk, the special partition device can be removed and the disk can then be encapsulated as a standard disk device.

A disk group cannot be formed entirely from nopriv devices. This is because nopriv devices do not provide space for storing disk group configuration information. Configuration information must be stored on at least one disk in the disk group.

Graphical User Interface 4

The Visual Administrator is a graphical user interface (GUI) for LSM. You enable the LSM Visual Administrator using the dxlsm command. The usage of windows, menus, and icons makes managing LSM disks relatively intuitive. Although this interface is designed primarily for disk and volume operations, the Visual Administrator also provides a limited set of file-system operations.

Because LSM and dxlsm function closely together, you should understand the principles of LSM operations before attempting to use the Visual Administrator. See Chapter 1 for an overview of LSM.

This chapter describes starting and using the Visual Administrator to manage LSM disk configurations.

4.1 Starting the Visual Administrator

If this is the first time you are running the Visual Administrator, you might need to update the PATH environment variable, as described in the following table:

IF...	THEN...
You are using a Bourne Shell-based shell (sh or ksh)	Update PATH to reflect /usr/bin/X11 with the following command: **PATH=${PATH}:/usr/bin/X11**
You are using a C Shell-based shell (csh)	Update PATH to reflect /usr/bin/X11 with the command: **setenv PATH ${PATH}:/usr/bin/X11**

Before you can start the LSM Visual Administrator, you must be logged into an account that has superuser privileges.

To start the Visual Administrator from the command line, enter the dxlsm command as follows:

dxlsm

The system returns the following message:

```
dxlsm is coming up, please wait.
```

When dxlsm comes up, it displays the main LSM Visual Administrator window called the root window, and the View of rootdg window. These windows are described later in Section 4.4.

4.2 Mouse Buttons

A two- or three-button mouse is required in order to use dxlsm. Table 4-1 describes the mouse buttons, referred to throughout this chapter as the MB1, MB2, and MB3 buttons.

Table 4-1: 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	Views properties of an object. If the icon is not undergoing analysis, it displays the properties form for that object. If the icon is undergoing analysis, it displays the analysis statistics form for that object.
Shift -MB1	Shift -Left	Shift -Left	Toggles between minimizing or maximizing an icon. A maximized icon displays all of its components in detail. A minimized icon is compressed and its components are concealed.
Shift -MB2	Shift -Middle	Ctrl -Right	Toggles between starting or 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.

Note

The examples in this document assume that you are using a three-button mouse, set up according to Table 4-1. Note that it is possible to redefine mouse buttons (using the `xmodmap` command, for instance). Refer to your X Window System documentation for details.

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.

4.3 Icons

The Visual Administrator interface uses icons to represent the following LSM objects:

- volumes
- plexes
- subdisks
- 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.

The following sections illustrate the icons, and describe how to manipulate LSM objects using them via the Visual Administrator's GUI.

4.3.1 Icon Characteristics

Figure 4-1 illustrates the icons that the dxlsm GUI uses to represent LSM objects.

Figure 4-1: Icons That Represent LSM Objects

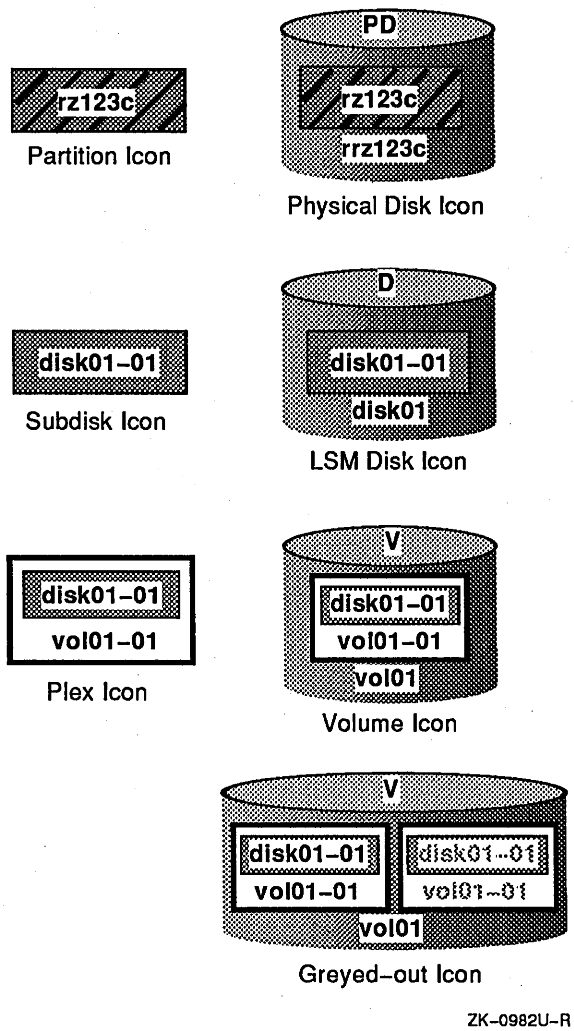


Table 4-2 describes the icons and their characteristics.

Table 4-2: Visual Administrator Icon Characteristics

Icon	Description
Physical Disk	Physical disks appear as cylindrical icons labeled PD. These icons represent physical disks known to dxlsm. Physical disk icons appear in the View of Disks window.
Partition	Partitions appear as rectangular icons within physical disk icons. The partition icon is labeled with the device name. If a disk has been added to a disk group, the corresponding partition icon is shaded. Partition icons appear in the View of Disks window.
LSM Disk	LSM disks appear as cylindrical icons labeled D. They usually contain subdisks, which are represented as rectangles. LSM disk icons represent disks that are both under LSM control and assigned to a disk group. LSM disk icons are labeled with the disk name, by default. LSM disk icons typically appear in a disk group view.
Subdisk	Subdisks appear within LSM disks (and often within plexes) as rectangular shaped icons. Subdisk icons typically appear in disk group views or in the View of Volumes window. Log subdisks (used to log recent disk activity) have icons with double borders to distinguish them from regular subdisk icons.
Plex	Plexes appear either alone or within volumes as relatively large rectangles containing subdisks. Plex icons have a heavy border to distinguish them from partition or subdisk icons. Plex icons typically appear in disk group views or in the View of Volumes window.
Volume	Volumes appear as cylindrical icons labeled V. These icons often contain plex and subdisk icons. Volume icons are distinguished from disk icons by a heavy border. Volume icons typically appear in disk group views or in the View of Volumes window.
Greyed out	With some operations, icons are updated almost instantly to reflect the results of the operation just performed. During other operations, it may take awhile for a particular icon to update itself. While being updated, icons are prevented from accepting input or undergoing configuration changes. Since an icon that is busy being updated should not be selected or manipulated, dxlsm greys out the text in that icon so that the user is aware that it is temporarily inaccessible. No input is accepted by an icon while it is greyed out. As soon as the icon is fully updated, it returns to its normal visual state and accepts input again. Icons that are temporarily greyed out in this manner are also referred to as *blocked icons*.

4.3.2 Manipulating Icons

There are two ways to manipulate icons:

Method	Description
Select-operate	The user first selects an icon by positioning the pointer on it and then clicking MB1 (when selecting a single icon) or MB2 (when selecting multiple icons) button. The mouse or keyboard can then be used to choose an operation (typically from a menu) to be applied to the selected icons.
Drag and drop	The user drags an icon and then drops it elsewhere. An icon is dragged by holding down MB1 and then moving the mouse, which moves an outline of that icon. The icon can then be positioned in a different location or on top of another icon and dropped there by releasing MB1. The resulting operation depends on the icon type and drop location.

The discussions in this chapter use the select-operate approach to dxlsm icon manipulation.

See also Chapter 5 for more information about these styles of operation.

4.4 Windows, Views, Menus, and Forms

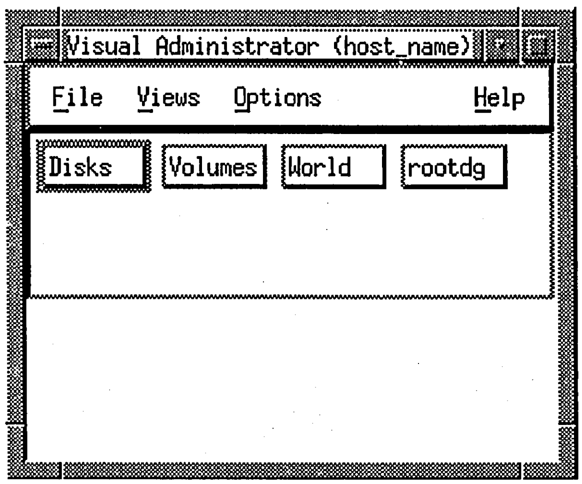
Once you start the Visual Administrator, any of the view windows can be accessed via the Visual Administrator root window. Views are special windows that display icons representing all LSM objects or a subset of objects currently known to LSM.

The following sections describe how to use the Visual Administrator views, menus, and forms.

4.4.1 Viewing the Root Window

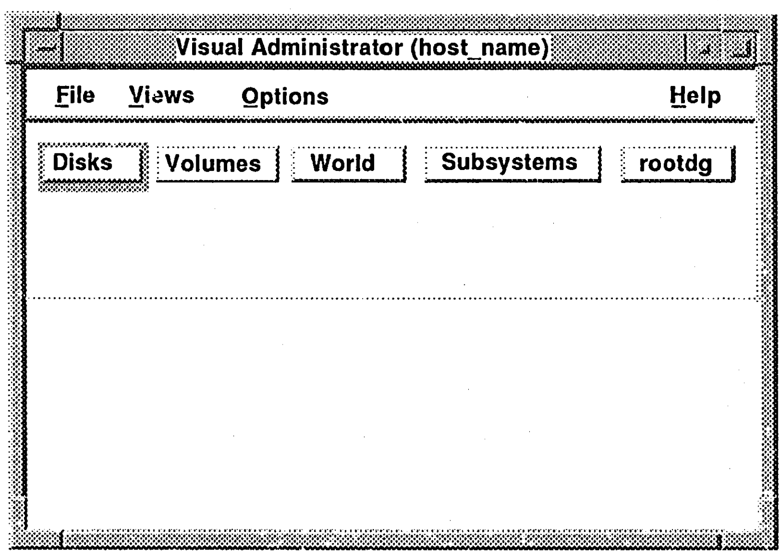
When the Visual Administrator comes up, it displays the main Visual Administrator window (also known as the root window). The root window contains a menu bar and a set of buttons. The set of buttons varies slightly depending on whether you have RAID (Redundant Arrays of Independent Disks) subsystems installed on your system. If you do not have RAID subsystems installed, the root window displays as shown in Figure 4-2.

Figure 4-2: Visual Administrator Root Window



If you have RAID subsystems installed, the root window displays as shown in Figure 4-3.

Figure 4-3: Visual Administrator Root Window for RAID Access



The menu bar contains the following pull-down menu items:

Item	Description
File	Closes the current window or exits the Visual Administrator interface completely
Views	Creates and manipulates user views
Options	Sets user preferences when using the Visual Administrator GUI
Help	Accesses the help facility

The root window provides access to the view windows. Immediately after the root window displays, the View of rootdg window also appears automatically. This view displays icons representing everything that currently exists in the `rootdg` disk group.

4.4.2 Views

From the Visual Administrator root window, you can use the pull-down views menu to get to the views windows. With views, you can examine and manipulate different parts of the physical and logical storage systems. You can add or remove icons from views only by using the LSM Visual Administrator.

Each view window title includes the name of the machine on which the session is running.

The Visual Administrator root window provides a view button area (the Disks, Volumes, World and rootdg views subwindows are shown in Figure 4-2) containing a button for every view on the system. Views are accessed by clicking MB1 on one of the view buttons in the views subwindow.

The Visual Administrator allows for two types of views in the Visual Administrator: default views and user-created views. Both types function identically, but certain restrictions are placed on default views. Default views cannot be removed or renamed by the user, as user-created views can.

4.4.2.1 Default Views

Click on the mouse buttons to access the following default view windows:

Menu Buttons	Window	Access
Disks	View of Disks	Displays all physical disks on the system
Volumes	View of Volumes	Displays all volumes, as well as plexes and associated subdisks, on the system

Menu Buttons	Window	Access
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

4.4.2.2 User-created Views

A user-created view is a view window that focuses on a particular part of a physical and a logical mass storage system, as defined by the system administrator. The system administrator can create views consisting of a selected collection of icons. For example, a user might create a special view to correspond to a physical or logical grouping (such as a view for the accounting department). User-created views enable the user to isolate part of the mass storage subsystem to observe or monitor that part of the configuration.

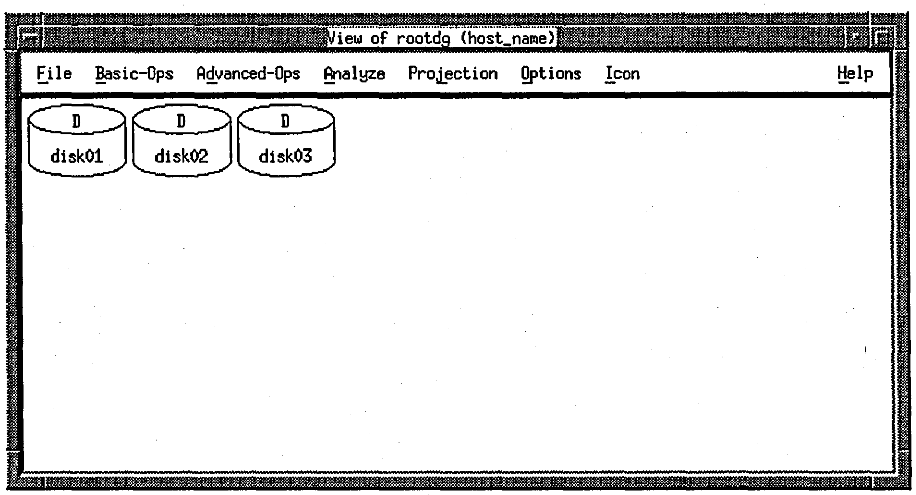
User-created views differ from default views in that they contain *copies* of icons from default views. Operations performed on these icon copies are reflected in the default views that display the affected icons. However, icons that appear in user-created views are not always updated whenever those icons are altered in the corresponding default view.

User-created views can be created using the Views pull-down menu from the Visual Administrator root window. Once created, icons can be added to a new view window by copying them over from existing views via the Icon menu.

4.4.2.2.1 Viewing the rootdg Window – By default, the View of rootdg window (which contains objects belonging to the root disk group) appears immediately after the Visual Administrator window displays (see Figure 4-4). The View of rootdg window contains icons representing objects currently recognized by LSM and belonging to the root disk group. You should perform operations in the View of rootdg window (or in another disk group view) whenever possible.

Figure 4-4 shows a View of rootdg window that does not yet contain any volumes.

Figure 4-4: View of rootdg



The View of rootdg window has a menu bar containing the following menu items:

Item	Description
File	Closes the current window or exits the Visual Administrator completely.
Basic-Ops	Accesses basic volume, file system, and disk operations.
Advanced-Ops	Accesses advanced operations involving volumes, disks, and other LSM objects.
Analyze	Analyzes and displays the activity level of objects.
Projection	Illustrates the relationships between certain objects.
Options	Sets user preferences for using the GUI. Also displays the Command Info window.
Icon	Manipulates icons.
Help	Accesses the help facility.

4.4.2.2.2 Viewing the Command Info Window – The Command Info window displays commands that are currently being executed by the Visual Administrator, as well as previous commands. Both LSM and other system commands are displayed here as they are invoked. The status and output of these commands is also displayed here.

You access the Command Info window by selecting Popup the Command Window from the Options menu. This window may also display automatically when an error occurs with a command.

This window is described in more detail in Chapter 5.

4.4.2.3 Viewing the Visual Administrator Help Window

Information pertaining to windows, functions, menus, and icons is available to the user at any time during a Visual Administrator session. The Help facility provides the user with concise information about current menu selections, form fields, Visual Administrator concepts, or other relevant topics. The Help facility is accessible through almost every window and form.

Help for a menu is accessed by clicking MB1 on the Help option located in the menu bar of a window or as the last option listed in a menu. Help for a form is accessed by clicking MB1 on the Help button located at the bottom of the form.

This window is described in more detail in Chapter 5.

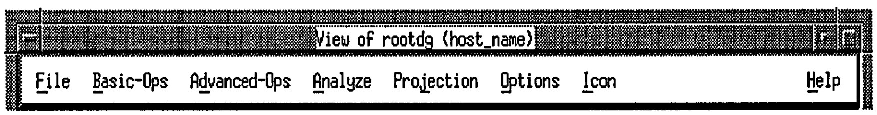
4.4.3 Pull-Down Menus

The Visual Administrator provides hierarchical menus called *pull-down menus*, that provide access to various Visual Administrator features, operations, forms, and user preference settings. Some of these menus contain submenus of their own. These menus allow you to select operations or access forms. Help is available on every menu to assist the user in understanding the current operation and navigating through the Visual Administrator. Menu access varies according to the type and location of the menu.

Each window contains a title bar that identifies the window. Below the title bar is a menu bar, which provides access to the menus available with that window. The menu bar contains the names of each available pull-down menu. One of the characters of each menu name is underlined; this is the mnemonic associated with that particular menu.

Figure 4-5 illustrates both the title bar and menu bar of a typical view window.

Figure 4-5: Title and Menu Bars



Note

From any view window, you can use the Basic-Ops or Advanced-Ops menus to create new LSM objects (represented by icons) or to manipulate existing icons.

Do not use the advanced operations to create and manipulate volumes and their components until you have a thorough understanding of LSM concepts.

Once a pull-down menu has been accessed, any available option in that menu can be selected. Any unavailable options are displayed within the menu in greyed-out text to distinguish them from available ones. Some menu selections lead to other submenus; some lead to forms that can be completed and then applied; some simply execute a specific operation immediately.

4.4.4 Forms

The Visual Administrator uses forms to present textual information for the user to examine and change. These forms provide useful information about existing objects and configurations.

There are two types of forms:

- General forms tend to appear during operations or setup requests and usually accept or require user input. You access general forms via certain menu selections.
- Properties forms display detailed information about a specific object's current characteristics, some of which can be modified directly through the properties form. You access properties forms by placing the pointer on the chosen icon and then clicking MB3 (unless the icon is undergoing analysis, in which case you must use the Shift-MB3 key instead).

Forms generally contain one or more fields that may accept or require user input. Forms contain buttons that can be used to carry out certain actions relating to the form. Error messages appear directly in forms when there is a problem with the form or its input. The following subsections describe the fields, buttons, and error messages.

4.4.4.1 Fields

Many forms require input to proceed with an operation. If a required form field is either blank or incorrect, an error message will result. Other form fields already contain information (such as default values), which may either be altered or accepted by the user. Yet other form fields are read-only and therefore cannot be changed; these fields beep when a user attempts to change them.

Form fields can be altered or completed in various ways, depending on the type of field. Some require text to be typed in, while others make use of assorted buttons that can be toggled or selected.

4.4.4.2 Buttons

All forms have buttons on the bottom of the screen that perform standard functions:

Apply	Accepts the information on this form, checks for errors, and continues with an operation. Pressing the Return key is equal to selecting the Apply key with a form. For a properties form, the Visual Administrator issues the appropriate commands to make the changes.
Reset	Fills in the fields of the form with its default values. If the form is a properties form, the Visual Administrator uses the values that were present when the form first displayed.
Cancel	Ignores all changes made on the form and closes it. If this was brought up as part of an operation, that operation is canceled.
Help	Displays the Help window with information about this form.

Some forms are read-only; only the the Help key and the Cancel buttons are provided on these forms.

Note

On some displays, certain forms may be too long to fit entirely on the screen and the buttons and fields at the bottom of the form may not be visible. If this is the case, the window manager's move window function (Alt - F7 , by default) can be used to move the form to view the desired buttons or fields.

4.4.4.3 Error Messages

Error messages are presented to the user when the Apply button is selected with one or more fields incorrect on the form. If this happens, then a message is printed at the bottom of the form and the user is allowed to correct the values for those fields. If the error cannot be corrected or the operation is no longer desired, you should select the Cancel button.

4.5 Disk Operations

This section describes the basic disk operations you can perform using the Visual Administrator.

Note

When you perform disk administration, it is important that you recognize the difference between a *device name* and a *disk name*. Refer to Section 1.6.2 for more information about how LSM disks are named.

4.5.1 Initializing a Disk

Whenever you add a new disk, you must identify the new disk to the system. This is also called initializing the disk. The system software cannot make use of the physical disk device without a software pointer to that disk. In order for LSM to be able to control a disk, that disk must also be identified to LSM.

To initialize a disk and identify it to LSM, use the LSM Support Operations menus (voldiskadm). Refer to Section 6.2.1 for information about using `voldiskadm` to initialize a disk. You can also refer to the `voldiskadm`(8) reference page.

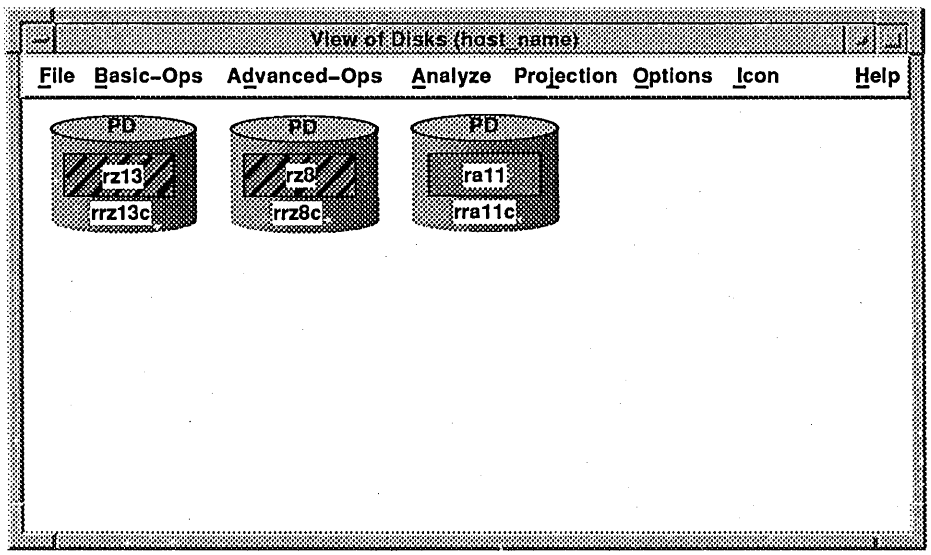
4.5.2 Displaying Disk Information

Before you use a disk, you should confirm that it has been initialized. With the Visual Administrator, physical disk icons containing partition icons appear in the View of Disks window. Those disks that are under LSM control contain partition icons that are colored or patterned. The LSM disks corresponding to colored or patterned partition icons are displayed in the window for the disk group to which they belong. Disk groups are represented visually as disk group views rather than icons. The View of rootdg window contains icons representing all LSM disks that exist within the `rootdg` disk group.

To see the disk information displayed in the View of Disks window, click MB1 on the Disks button in the Visual Administrator window. A View of

Disks window appears (see Figure 4-6).

Figure 4-6: View of Disks



The Visual Administrator allows you to see detailed information about a particular LSM disk in a properties form.

To display disk configuration information for a particular LSM disk, in the View of rootdg (or appropriate disk group view), click MB3 on the disk icon whose properties you are viewing. The disk's properties form appears, displaying detailed information about the disk.

It is possible to alter certain characteristics of the LSM disk (such as its name) via its properties form by editing the appropriate properties form field and then clicking MB1 on Apply.

4.5.3 Adding a Disk to a Disk Group

Disks are added to and grouped in disk groups for ease of administration.

To add another disk to an existing disk group, do the following:

1. In the View of Disks window, click MB1 on the partition icon representing the disk you want to add to a disk group.
2. Choose

   Advanced-Ops → Disk Groups → Add Disk

3. When the Add Disk Form appears, either leave rootdg as the default name in the Disk group field or enter the name of another existing disk group.
4. Enter the disk media name or leave the default name.
5. Click the MB1 button on Apply.

   If you look at the view window for the group you specified, you see that the new disk has been added to that disk group.

Note

You cannot put a disk in more than one disk group. If you attempt to do so, LSM returns an error message.

4.5.4 Renaming a Disk

Disks might be named according to their purpose (for example `docs0`), their owner (for example `smith02`), or their work group (for example `pubs01`). However, it is not necessary to give the disks special names (names such as `disk01` suffice). If the owner or purpose of the disk changes, you might want to change the disk name to reflect the change of ownership or use.

To rename a disk, do the following:

1. In the View of rootdg window (or the appropriate disk group view), click MB3 on the icon representing the disk to be renamed. This accesses the LSM disk properties form corresponding to that disk.
2. Enter the new disk name in the Logical Storage Manager disk name field in the properties form.
3. Click MB1 on Apply to change the disk name.

4.5.5 Initializing a New Disk Group

A disk group consists of one or more disks that share a common configuration. Disks are grouped together in disk groups for ease of administration.

To create a disk group, perform the following steps:

1. In the View of Disks window, select a disk by selecting the desired partition icon. This disk cannot already belong to a disk group.
2. Choose

   Advanced-Ops → Disk Groups → Initialize

3. In the Disk Group Initialize Form, fill in the Disk group field with the name you have chosen for the disk group.
4. Click MB1 on Apply to execute the initialize command.

4.5.6 Deporting a Disk Group

When a disk group is deported, access to that disk group is disabled. Disk groups may be deported if you intend to move or reuse the disks that are currently in the disk group.

To deport a disk group, do the following:

1. In the view of the disk group to be deported, choose:

   Advanced-Ops → Disk Group → Deport Disk Group

   In the Deport Disk Group Form, enter the name of the disk group to be deported.
2. Click MB1 on Apply to deport the disk group.

The deported disk group's view disappears.

4.5.7 Importing a Disk Group

Importing a disk group enables access to a deported disk group. The disk group to be imported must have been deported at one time and at least one disk that belonged to this disk group before it was deported must remain unused.

To import a disk group, do the following:

1. In any view window, select:

   Advanced-Ops → Disk Group → Import Disk Group

   In the Import Disk Group Form, enter the name of the disk group to be imported.
2. Click the MB1 button on Apply to import the disk group.

The view button for the disk group that has been imported appears in the Visual Administrator root window.

4.5.8 Displaying Disk Group Information

With the Visual Administrator, disk groups are represented by view windows rather than icons. A view for each disk group that exists is accessible through a button in the Logical Storage Visual Administrator root window.

To view the objects belonging to a particular disk group, click MB1 on the appropriate disk group button in the Logical Storage Visual Administrator root window. The view window for the desired disk group appears, displaying an icon for each object that belongs in that disk group.

4.5.9 Displaying Free Space

Before you add volumes and file systems to your system, you may want to make sure you have enough free disk space to adequately meet your needs. LSM lets you request a display of free space.

To display the free space on a disk, in the View of rootdg window (or the appropriate disk group view), click MB3 on the desired LSM disk. The properties form for the selected disk appears. The last field in this window shows the maximum free space available on that particular disk. Confirm that the amount of free space is sufficient.

4.5.10 Removing a Disk

The disk hardware is removable and can be moved between systems to where it is needed the most. However, before removing the disk from the current system, you must remove the software connections the disk has with the system. First you must remove the disk from its disk group, then you can remove the disk.

To remove a disk from a disk group, do the following:

1. In the disk group view to which the disk belongs, select the LSM disk you want to remove by clicking MB1 on its icon.
2. Choose

   Advanced-Ops → Disk Group → Remove Disks

The disk icon disappears from its disk group view.

To remove a disk that no longer belongs to a disk group, do the following:

1. In the View of Disks window, select the disk you want to remove by clicking MB2 on all partitions in the disk.
2. Choose

   Advanced-Ops → Disk → Remove Disks

Note

LSM does not allow you to remove the last disk in a disk group. To remove the last disk from a disk group, deport that disk group first, then reuse the disk.

4.6 Volume Operations

A volume is a virtual disk on which file systems or databases can be placed. This section describes the volume operations available with the Visual Administrator.

4.6.1 Creating a Simple Volume

To create a simple concatenated volume, perform the following steps:

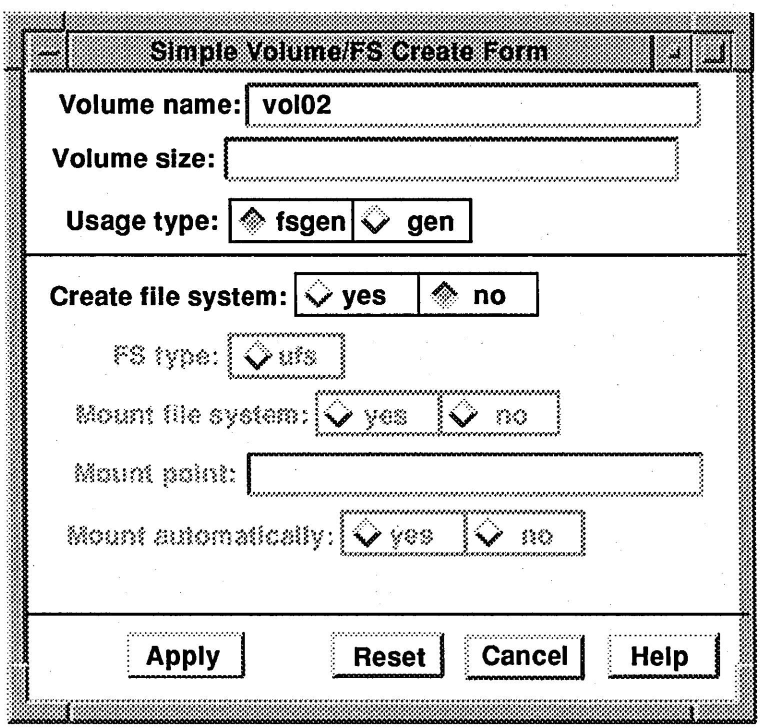
1. In the View of rootdg (or the appropriate disk group view), select

   Basic-Ops → Volume Operations → Create → Simple

2. In the Simple Volume/FS Create Form (Figure 4-7), either enter a name for the new volume or allow LSM to choose a default name (which appears in the Volume name field).
3. Enter a value in the Volume size field. If no unit is specified, sectors are assumed.
4. Select the Usage type. The default is `fsgen`. Use `gen` if you do not plan to create a file system on the volume.
5. Click the MB1 button on Apply to initialize the simple volume.

Figure 4-7 shows an example of this form.

Figure 4-7: Simple Volume/FS Create Form



Note

Because a file system is not created by default with this operation, the file system fields in the Simple Volume/FS Create Form are greyed out.

4.6.2 Creating a Striped Volume

A striped volume is one whose plex consists of a number of equal-sized subdisks, each located on a separate disk. Striped volumes provide faster average response time.

To create a striped volume, perform the following steps:

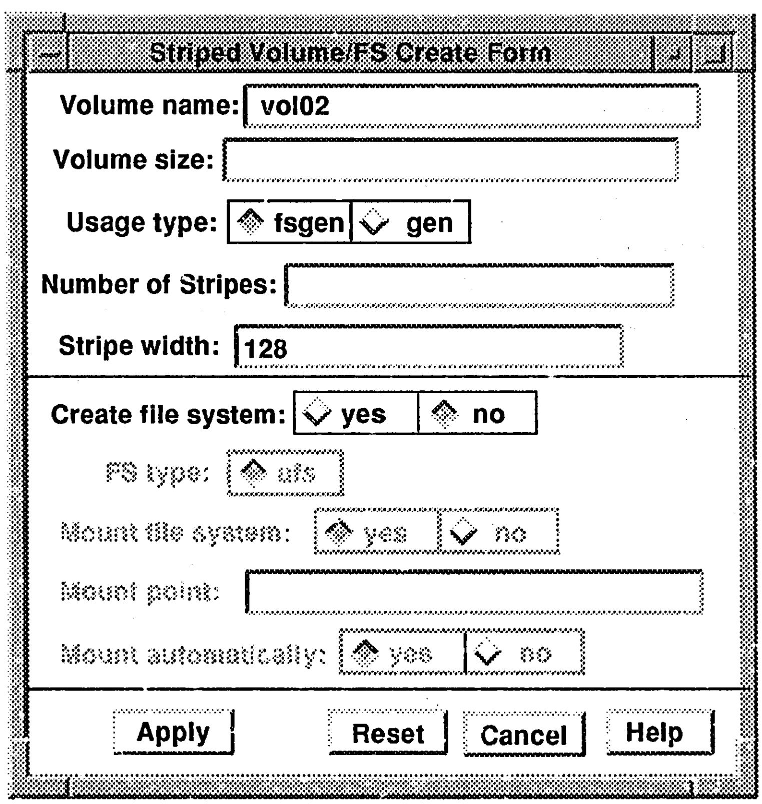
1. In the View of rootdg (or the appropriate disk group view), choose:

   Basic-Ops → Volume Operations → Create → Striped

2. In the Striped Volume/FS Create Form (Figure 4-8), either enter a name for the new volume or allow LSM to choose a default name (which appears in the Volume name field).
3. Enter a value in the Volume size field. If no unit is specified, sectors are assumed.
4. Select the Usage type. The default is `fsgen`. Use `gen` if you do not plan to create a file system on the volume.
5. Choose the number of stripes (which is equal to the number of disks on which the volume is to be created).
6. Choose the stripe width or let LSM use the default value.
7. Click the MB1 button on Apply to initialize the striped volume.

Figure 4-8 shows an example of this form.

Figure 4-8: Striped Volume/FS Create Form



Note

Because a file system is not created by default with this operation, the file system fields in the Striped Volume/FS Create Form are greyed out.

4.6.3 Creating a Volume on a Specific Disk

LSM automatically selects the disk or disks each volume will reside on, unless you specify otherwise. If you want a volume to reside on a specific disk, you must designate the disk for LSM.

To create a simple volume on a specific disk, do the following:

1. In the View of rootdg (or the appropriate disk group view), click MB1 on the desired LSM disk icon.
2. Choose

   Basic-Ops → Volume Operations → Create → Simple

3. In the Simple Volume/FS Create Form either enter a name for the new volume or allow LSM to choose a default name (which appears in the Volume name field).
4. Enter a value in the Volume size field. If no unit is specified, sectors are assumed.
5. Choose the Usage type. `fsgen` is the default. Use `gen` if you do not plan to create a file system on the volume.
6. Click the MB1 button on Apply to initialize the simple volume.

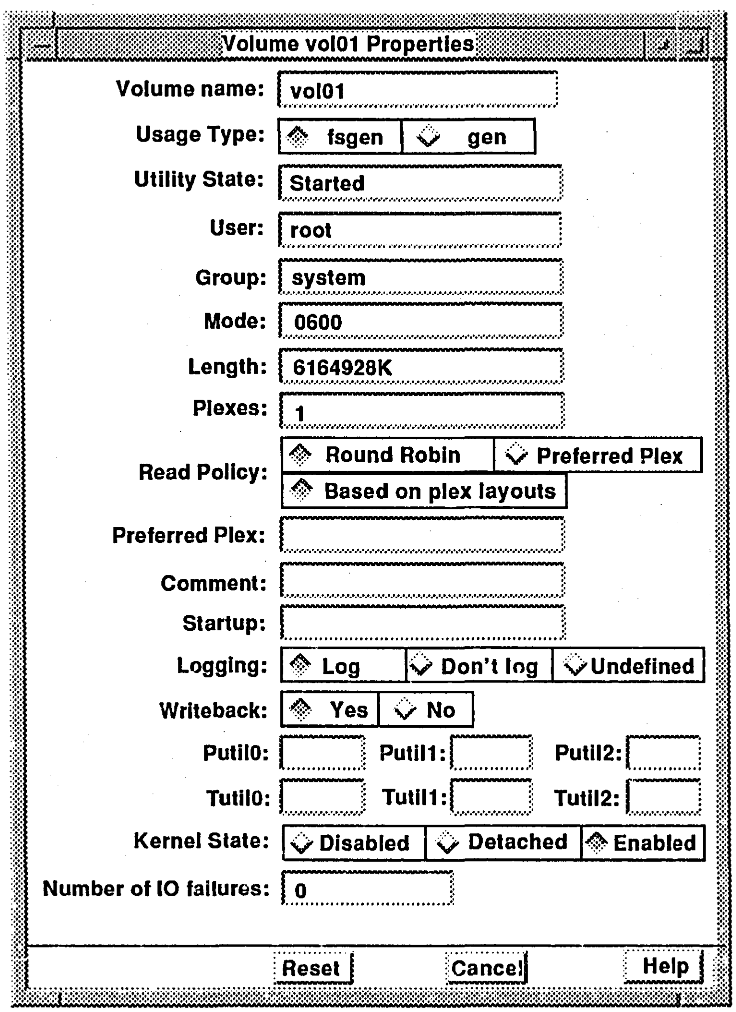
4.6.4 Displaying Volume Information

At times, you may want to see how a volume configured. The Visual Administrator allows you to view detailed information about a particular volume in a volume properties form.

To display volume configuration information for a particular volume, in the View of rootdg (or appropriate disk group view), click MB3 on the volume icon whose properties are to be viewed. The volume's properties form appears, displaying detailed information about the volume (see Figure 4-9).

It is possible to alter certain characteristics of the volume (such as its name) via its properties form by editing the appropriate properties form field and then clicking MB1 on Apply.

Figure 4-9: Volume Properties Form



4.6.5 Mirroring a Volume

A mirror is a copy of a volume. The mirror copy is not stored on the same disk as the original copy of the volume. Mirroring a volume assures you that the data in that volume will not be lost if one of your disk fails.

To mirror a volume, perform the following procedures:

1. In the View of rootdg (or appropriate disk group view), select the volume you want to mirror.

2. Choose

   Basic-Ops → Volume Operations → Add Mirror

3. At this point, indicate whether a simple or striped plex is to be created by selecting either Simple or Striped, respectively.

An additional plex appears within the volume icon's borders.

4.6.6 Extending a Volume

If the volume is not large enough for the amount of data that needs to be stored in it, you need to extend the volume's length.

Note

A striped volume cannot be extended.

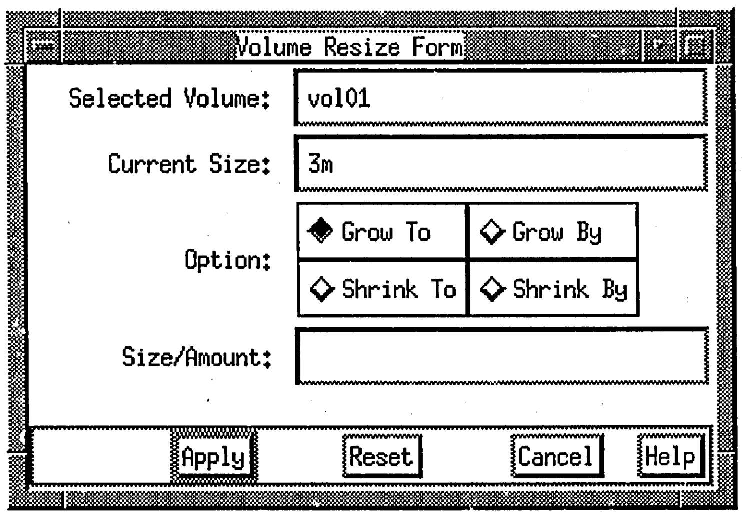
To extend a volume, do the following:

1. In the View of rootdg (or appropriate disk group view), select the volume you want to extend.
2. Choose

   Basic-Ops → Volume Operations → Resize

3. In the Volume Resize Form that appears (Figure 4-10), select either Grow To, or Grow By in the Option field.
4. If you selected Grow To in the Option field, enter the new size of the volume in the Size/Amount field. If you selected Grow By , enter the amount by which you want the volume to grow.
5. Click the MB1 button on Apply to execute the extend command.

Figure 4-10: Volume Resize Form



4.6.7 Shrinking a Volume

If you find that your volume is much larger than you really need it to be, you can shrink the volume's size. However, be aware that shrinking a volume containing data (perhaps in the form of a file system or database) can result in the loss of any data residing on the part of the volume that is removed.

Note

A striped volume cannot be shrunk.

To shrink a volume, complete the following operations:

1. In the View of rootdg (or the appropriate disk group view), select the volume you want to shrink.
2. Choose

   Basic-Ops → Volume Operations → Resize

3. In the Volume Resize Form that appears, select either Shrink To or Shrink By in the Option field.
4. If you selected Shrink To in the Option field, enter the new size of the volume in the Size/Amount field. If you selected Shrink By, enter the amount by which you want the volume to shrink.

5. Click the MB1 button on Apply to execute the shrink command.

4.6.8 Removing a Volume

Once a volume is no longer necessary (it is inactive and archived, for example), you can remove the volume and free up the disk space for other use. It is possible to remove a volume recursively, which automatically takes care of removing its associated plexes and freeing up its associated subdisks.

To remove a volume recursively, do the following:

1. In the View of rootdg (or appropriate disk group view), select the volume you want to remove.
2. Choose

   Basic-Ops → Volume Operations → Remove Volumes Recursively

3. If the volume is enabled, a dialog box containing a warning message appears to inform you that removing an enabled volume may destroy valuable data. Selecting the Cancel button abandons the removal, while selecting OK activates the removal despite the warning.

   Click the MB1 button on the OK button to remove the volume and its components.

4.6.9 Backing Up a Volume

It is very important to make back up copies of your volumes. This provides a copy of the data as it stands at the time of the backup. Backup copies are used to restore volumes lost due to disk failure, or data destroyed due to human error.

To back up a volume, perform the following steps:

1. In the View of rootdg (or appropriate disk group view), select the volume you want to back up.
2. Choose

   Basic-Ops → Volume Operations → Snapshot → Snapstart

3. The new plex that appears in the selected volume is greyed out until it is completely updated. At this point, you may want to notify users of the upcoming snapshot and ask them to save files and temporarily reduce activity.

Note

For UFS volumes, Digital recommends that you unmount the file system briefly to ensure the snapshot data on disk is consistent and complete.

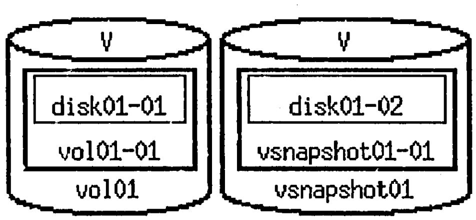
4. Choose

   Basic-Ops → Volume Operations → Snapshot → Snapshot

5. In the Snapshot Form that appears, either accept the default snapshot name that LSM supplies, or change the Snapshot Name field to the name of your choice.
6. Click the MB1 button on Apply to complete the backup snapshot. A new, snapshot volume appears (Figure 4-11). Normal usage of the original volume can now resume.
7. Backup the snapshot volume to tape.
8. Remove the snapshot volume (which now takes up unnecessary space) by selecting the snapshot volume and then selecting the following:

   Basic-Ops → Volume Operations → Remove Volumes Recursively

Figure 4-11: Volume and Snapshot



4.7 File System Operations

This section describes the file system management operations available with the Visual Administrator.

4.7.1 Creating a File System

With the Visual Administrator, a file system can be created on an underlying volume through a single operation. The layout of the volume on which the file system is created can be simple or striped.

4.7.1.1 Simple Volume

To create a file system on a simple concatenated volume, perform the following steps:

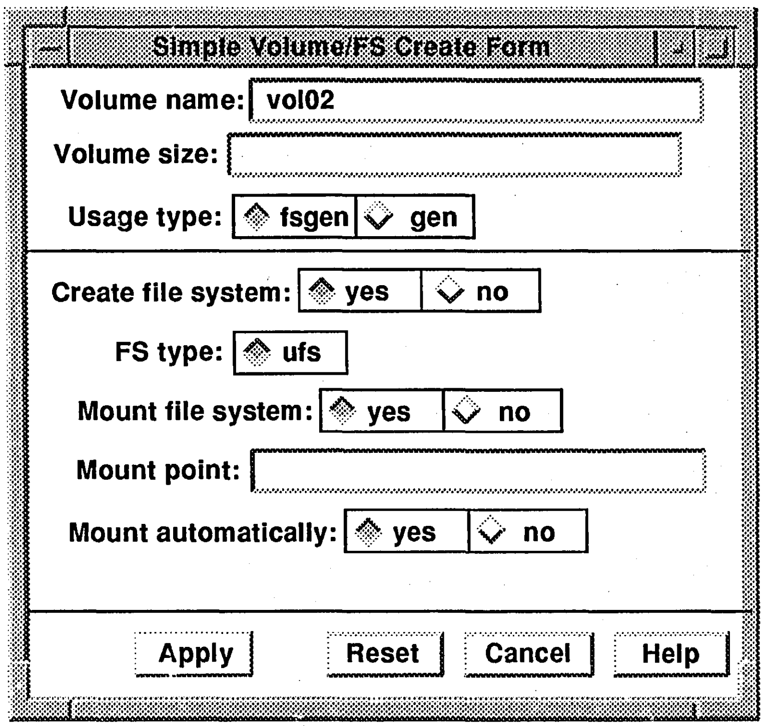
1. In the View of rootdg (or appropriate disk group view), select

   Basic-Ops → UFS Operations → Create → Simple

   This creates a new volume on which a new file system is created.
2. In the Simple Volume/FS Create Form (Figure 4-7), choose either a volume name for the new volume or allow LSM to choose a default name (which appears in the Volume name field).
3. Enter the desired volume size in the Volume size field. If no unit is specified, sectors are assumed.
4. Enter the Usage type . The default is `fsgen`.
5. Confirm that `Yes` is already selected in the Create file system field.
6. Choose a file system type. The default is `ufs`.
7. Choose `Yes` in the Mount file system field in order to mount the file system.
8. Enter the mount point for the new file system in the Mount point field.
9. Choose `Yes` in the Mount automatically field. This causes the file system to be mounted every time the system comes up.
10. Click the MB1 button on Apply to create and mount the file system.

Figure 4-12 shows an example of this form.

Figure 4-12: Simple Volume/FS Create Form



The mount point appears below the volume icon containing the mounted file system.

4.7.1.2 Striped Volume

To create a file system on a striped volume, perform the following steps:

1. In the View of rootdg (or appropriate disk group view), select:

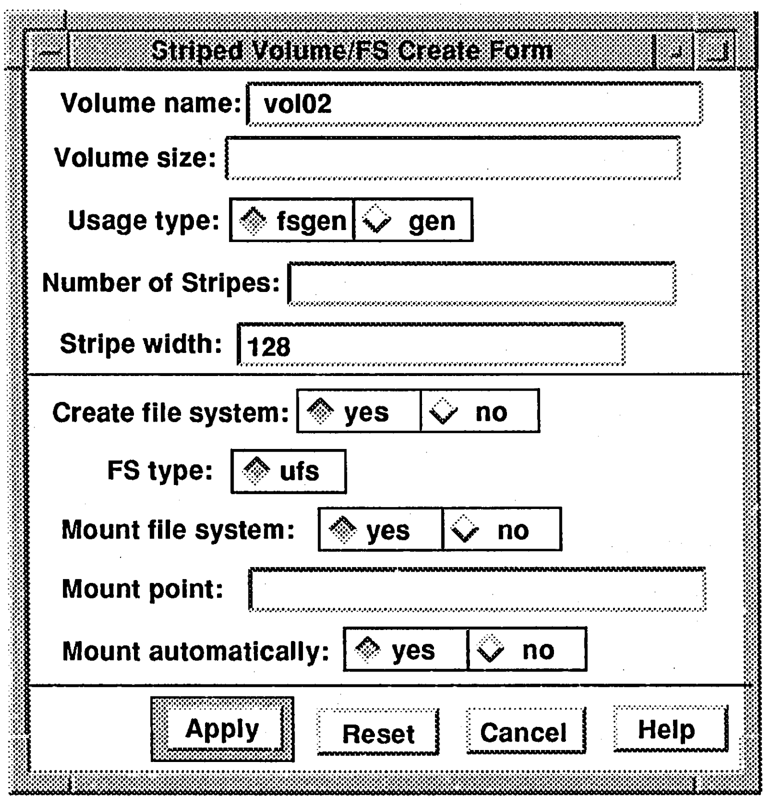
   Basic-Ops → UFS Operations → Create → Striped

2. In the Striped Volume/FS Create Form (Figure 4-13) either choose a volume name for the new file system or allow LSM to choose a default name (which appears in the Volume name field).
3. Enter the desired volume size in the Volume size field. If no unit is specified, sectors are assumed.
4. Enter the Usage type, the default is `fsgen`.

5. Enter the number of stripes to be created in the Number of Stripes field. This is equal to the number of disks on which the volume is to be created.
6. Enter the width of the stripes the volume will have in the Stripe width field. The default is 128 sectors.
7. Confirm that `Yes` is already selected in the Create file system field.
8. Choose a file system type. The default is `ufs`.
9. Choose `Yes` in the Mount file system field in order to mount the file system.
10. Enter the mount point for the new file system in the Mount point field.
11. Choose `Yes` in the Mount automatically field. This causes the file system to be mounted every time the system comes up.
12. Click the MB1 button on Apply to create and mount the file system.

Figure 4-13 shows an example of this form.

Figure 4-13: Striped Volume/FS Create Form



The mount point appears below the volume icon containing the mounted file system.

4.7.2 Mirroring a File System

A mirror is a copy of a volume. The mirror copy is not stored on the same disk as the original copy of the volume. Mirroring a volume containing a file system assures you that the data in that file system will not be lost if one of your disks fails.

To create a mirrored file system, perform the following steps:

1. Create a file system on a simple or striped volume, as described previously.
2. In the View of rootdg (or the appropriate disk group view), select the volume containing the file system to be mirrored.

3. Choose

   Basic-Ops → Volume Operations → Add Mirror

4. Choose a simple or striped plex layout by selecting `Simple` or `Striped`, respectively.

4.7.3 Making a File System

With the Visual Administrator, making a file system differs from creating a file system in that a file system is made on a volume that already exists. File systems can be created and placed on existing volumes, one file system per volume.

To make a file system, perform the following steps:

1. In the View of rootdg (or appropriate disk group view), select the volume icon on which to make the file system.
2. Choose

   Basic-Ops → UFS Operations → Make File System

3. In the Make File System Form, the name of the device on which the file system is to be made is displayed. This corresponds to the selected volume name and cannot be changed. The File system size field indicates the length of the file system to be made. This should correspond to the volume length, although it can be altered in special circumstances.
4. Choose a file system type. The default is `ufs`.
5. Choose `Yes` in the Mount file system field.
6. Enter the mount point for the new file system in the Mount point field.
7. Type `Yes` in the Mount automatically field. This causes the file system to be mounted every time the system comes up.
8. Click the MB1 button on Apply to make the file system.

4.7.4 Mounting a File System

A file system may exist on a volume, without being mounted.

To mount a file system, do the following:

1. In the View of rootdg (or appropriate disk group view), select the volume icon containing the valid, unmounted file system. (If a mount point is displayed below the volume icon, then that volume already contains a mounted file system.)

2. Choose

   Basic-Ops → UFS Operations → Mount

3. The Mount File System Form displays, containing the Device name field. This field displays the device on which to mount the file system. This corresponds to the volume you chose and you cannot change it.
4. Choose a file system type. The default is `ufs`.
5. Enter the mount point for the file system in the Mount point field.
6. Choose `Yes` in the Mount automatically field. This causes the file system to be mounted every time the system comes up.
7. Click the MB1 button on Apply to mount the file system.

When the file system is mounted, the mount point appears below the volume icon.

4.7.5 Unmounting a File System

A file system may be unmounted when it is no longer needed.

To unmount a file system, do the following:

1. In the View of rootdg (or appropriate disk group view), select the volume whose file system you want to unmount.
2. Choose

   Basic-Ops → UFS Operations → Unmount

4.7.6 Displaying a Mounted File System

With the Visual Administrator, it is possible to view the properties of mounted file systems.

To display a file system's properties, do the following:

1. In the View of rootdg (or appropriate disk group view), select the volume whose file system properties are to be displayed.
2. Choose

   Basic-Cps → UFS Operations → Display Properties

3. The File System Properties Form appears and displays detailed information about the file system mounted on the selected volume. You can select different mounted file systems from a menu box displayed in the upper left corner of this form.

4.8 Quitting dxlsm

To close a Visual Administrator window, select the following from the menu bar of that window:

File→Close

To end a dxlsm session completely, select the following from the menu bar area of any window.

File→Exit

A dialog box displays confirming that the dxlsm session is to be closed completely.

Advanced Visual Administrator Interface Operations 5

The Visual Administrator interface (dxlsm) provides the user with graphical elements such as windows, menus, and icons to ease the task of manipulating the LSM configuration.

This chapter describes in more detail some of the options available when you use graphical elements. In some cases, the discussions also describe how you can customize some of the various aspects of the Visual Administrator interface.

The Visual Administrator interface provides a consistent view of the LSM configuration. If a configuration or its objects are changed while a Visual Administrator session is running, the icons representing those objects automatically alter themselves to reflect such changes. The icons adjust themselves in this manner, regardless of whether the changes were made by the Visual Administrator itself or by another LSM interface.

5.1 Properties Form

You can display the properties of the object corresponding to a given icon (and potentially readjust them) in a properties form. Properties forms provide detailed information about the characteristics of a particular object.

You access a properties form for a particular icon by moving the pointer onto the icon and clicking the MB3 button. If the icon happens to be under analysis, use the Shift-MB3 key instead.

When the properties form appears, the Visual Administrator displays details of the icon's configuration. You can alter some of the displayed properties through this form directly by altering the appropriate fields and then selecting the form's Apply button. Selecting the Reset button converts any altered form fields back to their original values at the time when the form was popped up.

Properties forms exist for the following:

- Physical disks
- Partitions
- LSM disks
- Volumes

- Plexes
- Subdisks
- File systems

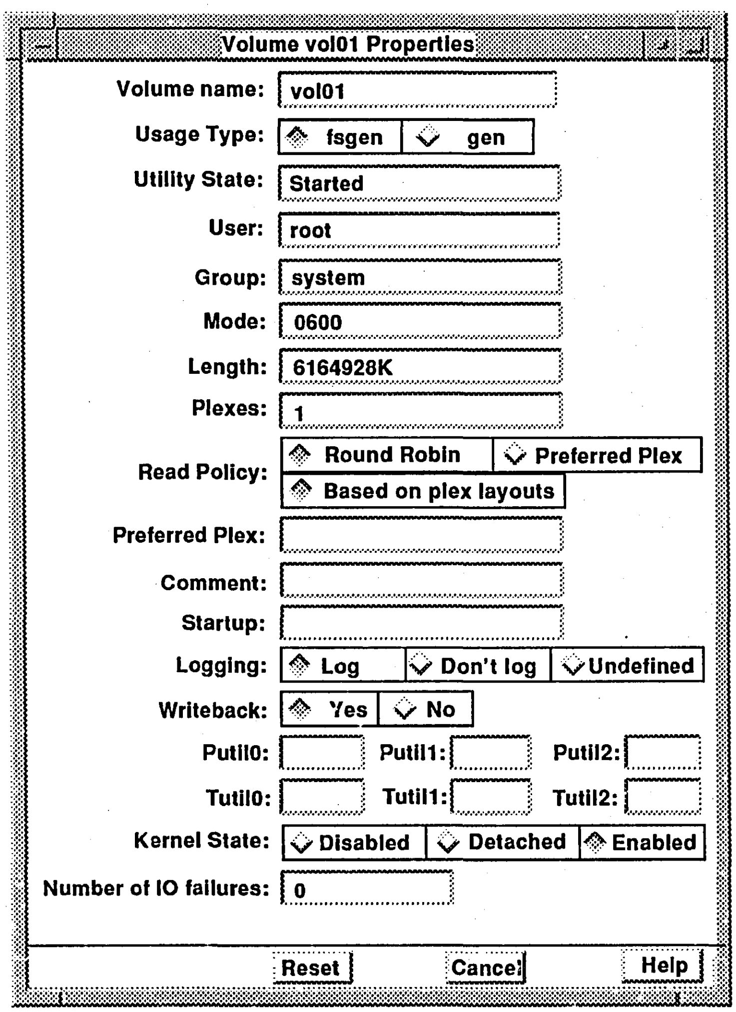
Because file systems are not represented by icons, you access the file system properties form via the File System Operations menu. If the file system happens to be mounted, you can access the file system properties form by clicking MB3 on the mount point name that appears below its volume icon.

5.2 Renaming Objects

The properties form is particularly useful for renaming objects represented by icons. To rename an object, access its properties form and alter the name field for that object. To activate the name change, click on the Apply button or press the Return key. The icon corresponding to the renamed object will adjust its name accordingly.

Figure 5-1 illustrates a volume properties form.

Figure 5-1: Volume Properties Form



5.3 Using the Icon Menu

You can access a set of icon-related options via the Icon menu located in the menu bar of each view window.

The following tables summarizes the icon-related options available through this menu:

Icon Option	Description
Maximize Icons	Maximize the selected minimized icons, so that it shows all of its subicons.
Minimize Icons	Minimize the selected icons, so that it shrinks the size and hides all of its subicons.
Maximize All Icons	Maximize all icons in the current view window at once.
Create Icons	Create a copy of the icons selected from another view and place the icon copy in the current user-created view.
Remove Icons	Remove the selected icons from the current user-created view.

5.4 Color

Depending on the type of monitor you are using, the Visual Administrator employs color or bitmap patterns to indicate the following:

- State of an icon
- Activity level of an icon
- Relationships between icons
- Failure of an operation

It is possible for a single icon to be in multiple states represented by different colors or patterns at once. For example, a given icon may be both selected and under projection at the same time. In such cases, the Visual Administrator reflects the color or pattern that represents the highest priority. The following is the priority list for possible icon states, starting with the highest priority:

1. Blocked
2. Error
3. Selected
4. Projected
5. Analyzed
6. Enabled

An icon that is in the blocked state (highest priority) is one that is currently busy and cannot allow any mouse or keyboard input. The text within a blocked icon is greyed out to indicate that it is inaccessible.

5.4.1 Default Colors and Patterns

If a color monitor is used, the default colors are red, yellow, grey and green. If a monochrome monitor is used, bitmap patterns of varying textures and shades are used instead of colors. By default, standard X Window System bitmaps (typically located in either `/usr/include/X11/bitmaps` or `/usr/bin/X11/bitmaps)` are used to create these patterns.

Table 5-1 describes the values for the default colors and bitmap patterns associated with icons under different conditions.

Table 5-1: Default Colors and Patterns

Situation	Color	Bitmap Pattern
selected icon	royal blue	gray3
disabled icon	light grey	stripe4
alarmed icon	red	gray1
free subdisk icon	light grey	root_weave
projection	deep pink	root_weave
analysis: low	green	cross_weave
analysis: medium	yellow	root_weave
analysis: high	red	wide_weave

Note

This guide assumes that the default colors or bitmaps are being used. However, colors or bitmaps (as well as other interface preferences) can be redefined in the user's `.Xdefaults` file.

5.4.2 Specifying Alternate Colors and Patterns

This section lists X resources that can be used to configure the Visual Administrator according to personal preferences and system requirements.

The Visual Administrator resources and associated preferences can be specified in your `.Xdefaults` file. A file with default dxlsm entries is located in `/usr/lib/X11/app-defaults/DXlsm`. The entries in this file are commented out. You can uncomment the lines that you want to enable. Refer also to your X window system documentation on X resources for further information.

The default values specified here correspond to those defaults compiled into the Visual Administrator. Preferences specified in the system's `app-defaults` file may change these defaults.

The entries in the `.Xdefaults` file should take the following form:

DXlsm* *resource:*   *value*

For example, the color used to represent a disabled icon can be altered from the default color (light grey) to orange by editing the `.Xdefaults` file to include the following line:

DXlsm*disabledPixel: orange

The dxlsm-related resources can also be specified for a single session only by invoking the Visual Administrator as follows:

DXlsm -xrm "dxlsm* *resource: value"*

Section 5.4.2.1 through Section 5.4.2.4 describe the dxlsm-related resources and their default values for color, monochrome, icon, and other resources. The default values can be changed according to user preferences. The resources are listed to the left with their default values to the right. Each resource-value pair is followed by a brief description.

5.4.2.1 Color Resources

The following resources only apply when the Visual Administrator is run on a color monitor:

foreground   black

The color in which all foreground items are displayed. This typically applies to icon outlines and text.

background   white

The color that serves as the background for all windows in the Visual Administrator.

selectedPixel   royal blue

The color of icons that have been selected.

disabledPixel   light grey

The color of icons that are disabled and cannot be used by Visual

Administrator (for example, detached plexes).

alarmPixel	red

The color of icons that have been selected when an error occurs (for example, incorrectly selected icons).

freesdPixel	light grey

The color of subdisk icons that are free (unassociated) when Show Free Subdisks has been turned on.

projectPixel	deep pink

The color of icons that are projecting (displaying object relationships) when Icon Projection has been turned on for that icon or a related icon.

lowPixel	green

The color of icons that have a *low* usage level (as defined in the Analysis Properties Form) when analysis has been turned on for that icon or a related icon.

midPixel	yellow

The color of icons that have a *medium* usage level (as defined in the Analysis Properties Form) when analysis has been turned on for that icon or a related icon.

highPixel	red

The color of icons that have a *high* usage level (as defined in the Analysis Properties Form) when analysis has been turned on for that icon or a related icon.

mono	False

When `True` is specified, the Visual Administrator is forced to operate in monochrome (black and white) mode, whether or not a color monitor is being used.

5.4.2.2 Monochrome Resources

The following resources only apply when the Visual Administrator is run on a monochrome monitor:

foreground black

The color in which all foreground items are displayed. This typically applies to icon outlines and text.

background white

The color that serves as the background for all windows in the Visual Administrator.

selectedPixmap gray3

The bitmap pattern for icons that have been selected.

disabledPixmap stripe4

The bitmap pattern for icons that are disabled and cannot be used by the Visual Administrator (detached plexes, for example).

alarmPixmap gray1

The bitmap pattern for icons that have been selected when an error occurs (incorrectly selected icons, for example).

freesdPixmap root_weave

The bitmap pattern for subdisk icons that are free (unassociated) when Show Free Subdisks has been turned on.

projectPixmap root_weave

The bitmap pattern for icons that are projecting (displaying object relationships) when Icon Projection has been turned on for that icon or a related icon.

lowPixmap cross_weave

The bitmap pattern for icons that have a *low* usage level (as defined in the Analysis Properties Form) when analysis has been turned on for that icon or a related icon.

midPixmap	root_weave

The bitmap pattern for icons that have a *medium* usage level (as defined in the Analysis Properties Form) when analysis has been turned on for that icon or a related icon.

highPixmap	wide_weave

The bitmap pattern for icons that have a *high* usage level (as defined in the Analysis Properties Form) when analysis has been turned on for that icon or a related icon.

5.4.2.3 Icon Resources

The following resources relate to icons.

volumeMinimizeIcons	False

When `True` is specified, volume icons will be minimized when created, by default.

plexMinimizeIcons	False

When `True` is specified, plex icons will be minimized when created, by default. This feature is useful to display structures within volumes, but to hide details about the subdisk structure that makes up the plex.

diskMinimizeIcons	False

When `True` is specified, disk icons will be minimized when created, by default.

phyDiskMinimizeIcons	False

When `True` is specified, physical disk icons will be minimized when created, by default.

autoDeselect	True

When `True` is specified, icons selected for an operation are automatically deselected when the operation completes. If set to `False`, icons are remain selected until the user decides to deselect them, making it possible to perform multiple operations on the same set of selected icons.

5.4.2.4 Miscellaneous Resources

The following are miscellaneous dxlsm-related resources:

lsvaHelp	False

When `True` is specified to IsvalHelp, the Visual Administrator displays a help message (including command line option usage information) in a window at program start up.

title	"Visual Administrator"

This is the title of the application's root window.

fontList	fixed

This describes the font to be used for all text within the Visual Administrator.

commandSilos	50

Use this to specify the number of command silos supported. A *command silo* is a set of sequentially dependent commands (like file system create, followed by file system mount). A larger number of silos supports a larger number of concurrent operations that can be run, but also requires the Visual Administrator to use more memory.

commandHistorySize	20

Use commandHistorySize to specify the number of commands that the Visual Administrator should remember and display in the history portion of the Command Info Window.

defaultViewWindow	rootdg

Use the defaultViewWindow to specify the name of the disk group to be popped up by default when the Visual Administrator is run.

chkMntptInterval	5

Use chkMntptInterval to specify how often, in seconds, the Visual Administrator should check the system mount table to accurately display information about mounted file systems.

twoButtonMouse False

When `True` is specified to twoButtonMouse, the Visual Administrator remaps the mouse buttons for a two button mouse.

5.4.3 Window Adjustments

On small displays (such as those with a graphical resolution of 640x480), some windows or forms may be too long to fit entirely on the screen and the bottom area of these windows/forms may not be visible. If this is the case, the window manager's move function (ALT-F7, by default) can be used to move the window or form so that all areas and form buttons are visible.

Another technique that may allow forms to fit better on a small screen is to start up dxlsm as follows:

dxlsm -xrm "dxlsm*propertiesForm*marginHeight: 1"

This resource specification causes forms to appear shorter than normal. To achieve similar results, you can add the following lines to your `$HOME/.Xdefaults` file:

dxlsm*propertiesForm*marginHeight: 1
dxlsm*propertiesForm*marginWidth: 1

5.5 Projection

Projection is the technique that the Visual Administrator uses to show relationships between icons that represent LSM objects. Projection is illustrated using color (deep pink is the default) or bitmap patterns. Projection highlights those objects that the selected object is composed of and illustrates the relationship between the objects. For example, if a volume is selected for projection, the corresponding subdisks are highlighted within the volume icon and also on the appropriate disk icons. If the selected icon has no associated objects, the Visual Administrator issues a warning to this effect.

To show the projection of a particular icon, click the MB2 button on the icon while holding down the Shift key (Shift-MB2). To stop projection, press Shift-MB2 again. Projection can also be started or stopped by selecting an icon and then using the Icon Projection submenu of the Projection menu.

Note

Volume, plex, subdisk, and LSM disk icons can be selected for projection. Projection does not apply to physical disk or partition icons.

Projection may be requested in any view. When an icon is highlighted by projection, all icons representing that object in all view windows where it appears are highlighted.

Table 5-2 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.

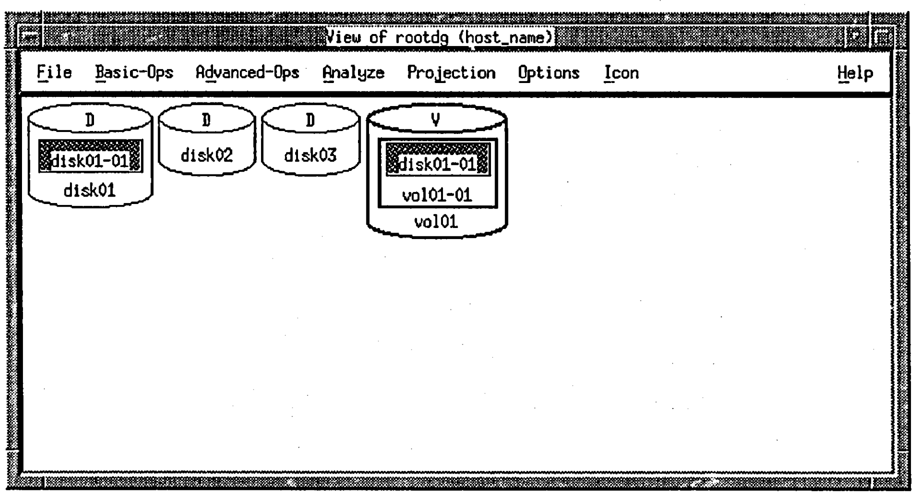
Table 5-2: Projection Behavior

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

When projection is turned on and left on from two different objects, any icon that happens to be related to both of these objects receives two layers of projection highlighting. Projection must then be turned off from both objects that started it (or all projection in the current session must be stopped) before the double-highlighted icon returns to its normal state. For example, if projection is turned on from both a volume and a plex related to the same subdisk, then that subdisk is highlighted twice even though it only appears to have one layer of highlighting.

Figure 5-2 illustrates highlighting that results from the selection of a volume icon for projection.

Figure 5-2: Projection



5.6 Free Subdisks

The Projection menu provides access to a feature that highlights any free subdisk icons. This is useful for identifying subdisks that are not currently associated with any plexes and should either be used or removed to free up the space that they occupy.

To turn on highlighting of free subdisks, access the Projection menu and select Show Free Subdisks, then Start. Once turned on, free subdisks will continue to be highlighted until this feature is turned off.

5.7 Analysis

Through analysis, the Visual Administrator displays statistics about the performance of LSM objects. These statistics are displayed both visually (with different colors or patterns), and numerically (with pop-up statistics forms). Analysis displays activity levels on selected icons representing LSM objects.

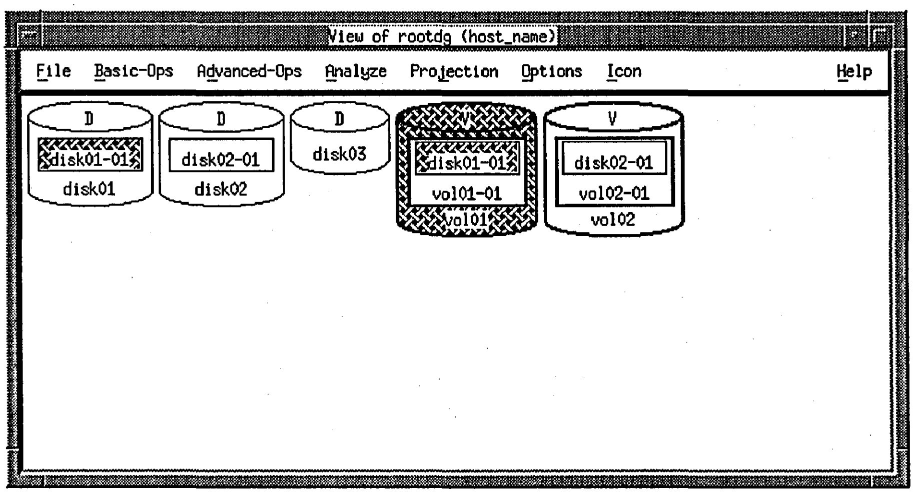
Note

Only volume and LSM disk icons can be selected for analysis.

5.7.1 Starting Analysis

To start analysis, choose one or more LSM disk and volume icons, then choose Start from the Analyze menu. Volumes, LSM disks, and subdisks that are associated with the selected icons change their colors or patterns to reflect their relative activity levels (high, medium, or low). Figure 5-3 illustrates analysis for a selected volume.

Figure 5-3: Analysis



Once started, you can analyze any additional objects by choosing more LSM disk or volume icons.

5.7.2 Stopping Analysis

Analysis can be stopped either completely or for a selected set of objects only.

- To stop analysis for all objects that are currently being analyzed, perform these steps:
  1. Go to the view window corresponding to the disk group in which you want to perform this operation (View of rootdg, by default).

2. From the Analyze menu, choose Stop All.

   Analysis of all objects in the current disk group is terminated and all affected icons return to their nonanalysis colors and patterns. Any open statistics forms are closed.

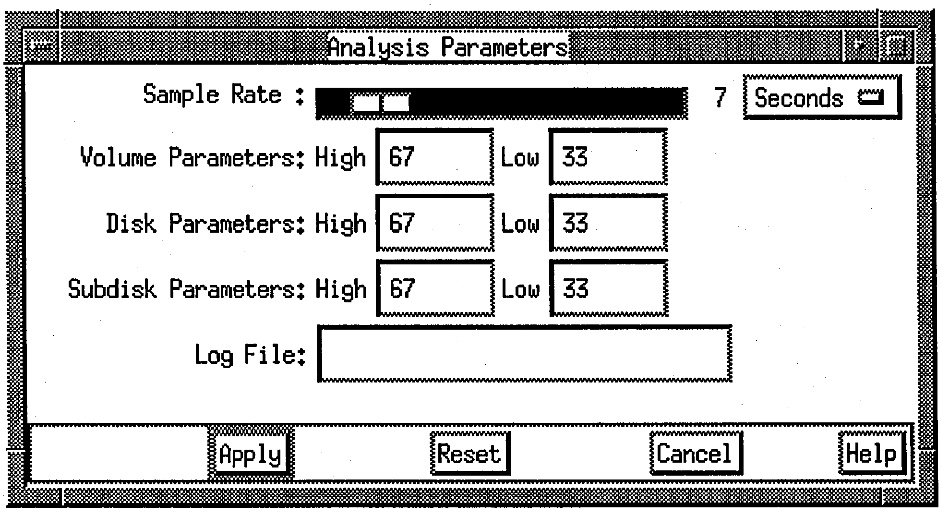
- To stop analysis for specific objects, perform the following steps:
  1. Go to the view window corresponding to the disk group in which you want to perform this operation (View of rootdg, by default).
  2. Select the disks and volumes whose analysis is to be stopped by clicking the MB2 button on the appropriate icons.
  3. From the Analyze menu, select Stop.

Analysis of the selected objects in the current disk group is terminated and their icons return to their nonanalysis colors and patterns. Any open statistics forms for those objects that are no longer being analyzed are closed. Analysis for the remaining, non-selected icons continues.

5.7.3 Setting Parameters

User preferences for analysis are specified via the Analysis Parameters form (see Figure 5-4). The fields in this form allow you to specify the cutoff values for coloring or patterning of the icons under analysis. These values are compared to the Total Read/Write field to see if the icon represents a high, medium, or low use object. You can define a separate high/low pair of parameters for subdisk, volume, and disk analysis. The specified parameters will persist across dxlsm runs.

Figure 5-4: Setting Parameters in Analyze Menu



Set analysis parameters as follows:

1. From the Analyze menu, choose Parameters. The Analysis Parameters form displays.
2. Complete the Analysis Parameters form as follows:

Sample Rate: Use the slider bar to select the time interval, in seconds, between data samples. The interval should be between one and sixty seconds or minutes. A shorter interval means that data will be updated more often, but is also a higher load on the system. The default is 5 seconds.

To convert the interval from seconds (default) to minutes, click on MB1 on the box to the right of the radio button; when the pop-up menu displays, select minutes.

Volume Parameters: Enter the high and low values that will decide the coloring or patterning of the volume icons. The units for the values are total number of read/write operations. Although high and low values appear in this field, change them according to the disk type.

Disk Parameters: Enter the high and low values that will decide the coloring or patterning of the LSM disk icons. The units for the values are total number of read and write operations. Although high and low values appear in this field, they should be changed according to the disk type.

Subdisk Parameters: Enter the high and low values that will decide the coloring or patterning of the subdisk icons. The units for the values are total number of read and write operations. Although high and low values appear in this field, they should be changed according to the disk type.

Log File: Enter the name of the file to be used for statistics logging. If the file does not already exist, it will be created. To stop logging to the file, you must erase the file name text in this field. If logging is left in effect too long, a very large log file could result. Unlike the other fields, this one does not persist across dxlsm sessions.

This log file is a binary file. In order to view the log file, `/usr/bin/lsmlog2text` *`filename`* must be run on this file to process it for viewing.

3. When the form is properly completed, click on the Apply button to set the parameters.

5.7.4 Statistics Forms

When an icon is under analysis, the Analysis Statistics form for that icon can be displayed by clicking the MB3 button on the icon. Because the MB3 button is normally used to access an icon's properties form, use the Shift-MB3 button to access the properties form of an icon undergoing analysis instead.

Because no fields on this form may be changed, the only way to close the form is by clicking on the Cancel button.

The fields of the Analysis Statistics form display the following read-only information, which is continually updated:

Reads: Number of times the object was read from during the last interval.

Writes: Number of times the object was written to during the last interval.

Total R/W: Total number of reads and writes during the last interval.

Blocks Read: Number of disk blocks read from the object during the last interval.

Blocks Written: Number of disk blocks written to the object during the last interval.

Total Blocks: Total number of blocks read from or written to the object during the last interval.

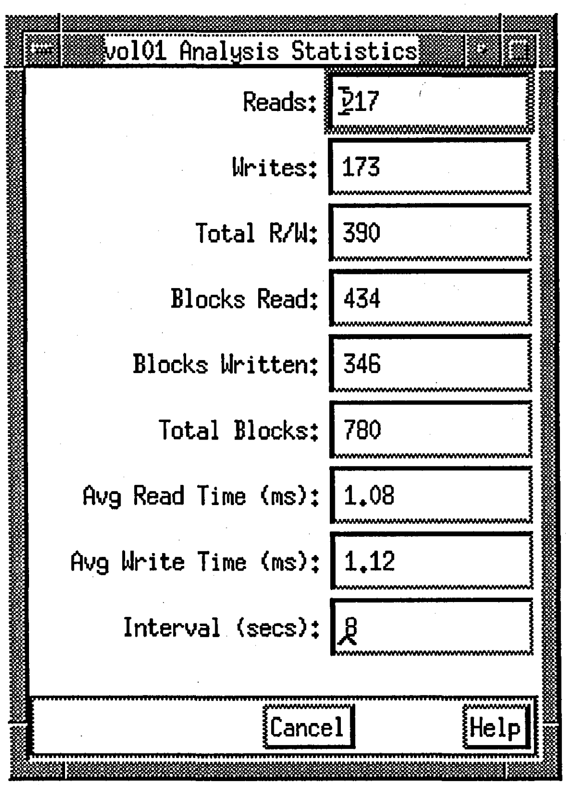
Avg Read Time: Average time, in milliseconds, that it took for a read operation to complete. This is equal to the number of number of reads during the last interval divided by the total time spent on reads.

Avg Write Time: Average time, in milliseconds, that it took for a write operation to complete. This is equal to the number of number of writes during the last interval divided by the total time spent on writes.

Interval: 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.

Figure 5-5 shows a typical Analysis Statistics form.

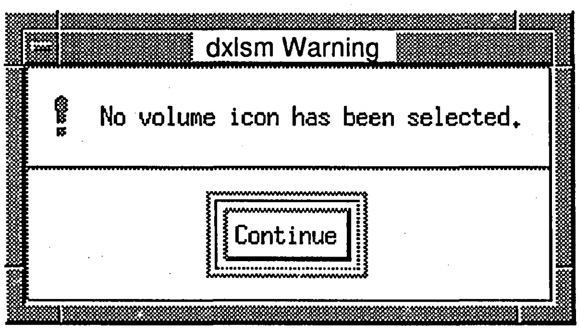
Figure 5-5: Analysis Statistics Form



5.8 Error and Warning Messages

The Visual Administrator uses dialog boxes to present error or warning messages. When a message is displayed in this manner, you must acknowledge it by selecting one of the buttons displayed in the error dialog box before proceeding. Some warning boxes announce that a prerequisite has not been met and require you to acknowledge this by clicking the displayed Continue button before reattempting the operation (see Figure 5-6).

Figure 5-6: Warning Box for dxlsm



5.9 Styles of Operation

Operations can be performed in the following ways within the Visual Administrator:

Style	Description
Select-Operate	An icon representing an LSM object is selected and then the desired operation is performed on that object via menus and forms.
Drag and Drop	An object is manipulated by dragging its icon and then dropping it elsewhere (such as on another object in any view window).

5.9.1 Select-Operate Operation

The select-operate operation works as follows:

1. Select an icon representing a LSM object by positioning the pointer over the icon and clicking the MB1 button (when selecting a single icon) or the MB2 button (when selecting multiple icons).
2. From the Basic-Ops and Advanced-Ops pull-down menus, choose and activate the operation to be performed on the selected icons. Some operations require that a form be completed before the operation is activated.

5.9.2 Drag and Drop Operations

The drag and drop operation works as follows:

1. Drag an icon by placing the pointer on it, pressing and continuing to hold down the MB1 mouse button, and moving the mouse until the outline of the icon is superimposed on its destination.
2. Drop an icon by releasing the MB1 mouse button when the icon reaches its drop location.

When an icon is dragged and dropped, the resulting operation depends on the icon type and drop location. The drop location can be another icon or a different location.

Note

> 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.

Some icons can be copied from one user-created view window to another. Only one copy of a particular icon may appear in a view, however. Attempts to drop a second copy are ignored.

Table 5-3 summarizes the possible drag and drop operations.

Table 5-3: Drag and Drop Operations

Icon Type	Drop Location	Action
Free subdisk	View window	Create a plex and associate the subdisk to the plex.
Free subdisk	LSM disk	Create an identically-sized subdisk on the disk.
Free subdisk	Plex	Associate the subdisk to the plex.
Associated subdisk	Free subdisk	Swap the associated subdisk with the free subdisk. The free subdisk becomes associated and replaces the original subdisk, which is removed.

Table 5-3: (continued)

Icon Type	Drop Location	Action
Associated subdisk	LSM disk	Create an identical free subdisk on the LSM disk, then swap 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	Dissociate the subdisk.
Associated plex	View window	Dissociate the plex.
Dissociated plex	User's view window	Copy the plex icon to the user's view.
Plex	Volume	Associate the plex to the volume.
Disk	User's view window	Copy the physical or LSM disk icon to the user's view.
Volume	User's view window	Copy the volume icon to the user's view.
Partition	Disk group view	Add a LSM disk (corresponding to the partition) to that disk group.

Note

If the user performs a drag and drop operation that is not recognized by the Visual Administrator, that drag and drop operation is either ignored or a warning message box appears. In some cases, the Command Info Window displays an error relating to an unacceptable drag and drop attempt.

5.10 Using Help

The Visual Administrator provides an extensive Help facility to assist users while navigating through the Visual Administrator. Help windows are scattered throughout the various Visual Administrator windows, menus, and forms to provide the user with relevant information. Help information varies according to the location and circumstances.

Depending on the location, Help windows are accessed by:

- Clicking the MB1 mouse button on the Help menu in a window's menu bar

- Clicking the MB1 mouse button on the Help option in a menu
- Clicking the MB1 mouse button on the Help button at the bottom of a form

Once Help is invoked, a Help window containing information relevant to the current window, menu, form, or operation appears.

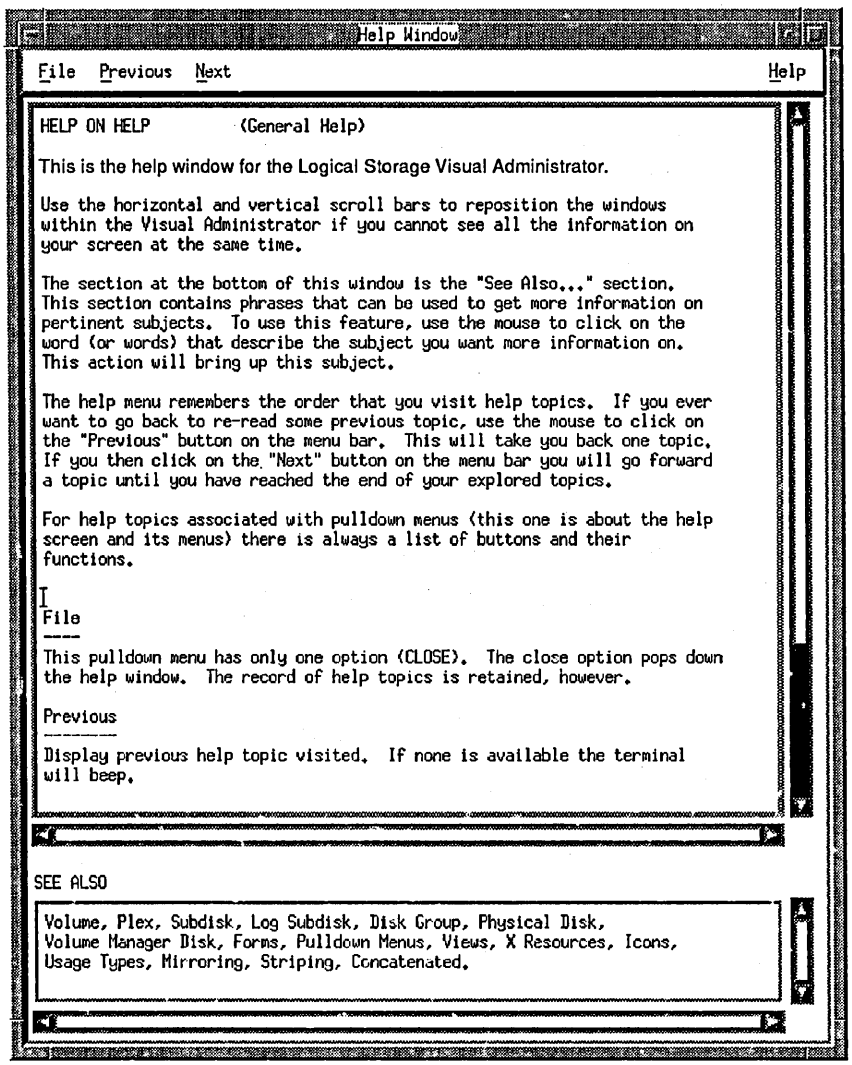
The Help window is equipped with both horizontal and vertical scrollbars, which can be used to scroll through the available Help text.

At the bottom of each Help window is a SEE ALSO area. This area lists similar or related Help topics and operates hypertext-style. To access any of the listed Help topics, simply click the MB1 button on the appropriate words in the SEE ALSO list; a new Help screen containing information on the selected topic replaces the previous help text.

The Help facility keeps track of the order in which Help topics are visited. The user can therefore return to reread a previous topic. Selecting Previous from the menu bar displays the previous topic. Selecting Next displays the next topic.

Figure 5-7 illustrates a typical Help window.

Figure 5-7: Help Window



The Help menu in the menu bar of the Help window itself provides access to the following:

Type	Description
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. Clicking the MB1 button directly over the topic name in the SEE ALSO section causes that topic to appear in the Help window.

The Help window can be closed by selecting the Close option from the File menu. The record of help topics visited is retained, however.

5.11 User Preferences

Most of the Visual Administrator windows contain an Options menu that allows the user to specify preferences on how the Visual Administrator should operate. To access this menu, click the MB1 mouse button on Options in a window's menu bar or press Alt-o within a window displaying this option. Some of the preferences listed have associated submenus that require further selection.

User preferences are saved in the file `.dxlsm_pref`, which is automatically created in the user's home directory. Each user has a personal `.dxlsm_pref` file.

The preferences that can be set through the Options menu are as follows:

- Show Command
- When Commands are Ready
- Logging
- Popup the Command Window
- Units of Size-Related Output
- Help

Some of the available preference settings relate to the Command Info Window. This is a special window that displays the command history, along with the status and results of those commands.

5.11.1 Show Command

This specifies whether the Command Info Window is to be shown before every command is executed (Show at Start), or only when a command fails (Show on Error). If no preference is indicated, the default is Show on Error.

5.11.2 When Commands are Ready

This specifies what the Visual Administrator should do when it is ready to run an LSM or system command. The command can either be run immediately (Execute Commands), or brought up in the Command Info Window for inspection (Show Commands Only). If Show Commands Only is chosen, the command can be executed directly from the Command Info Window once it is approved. If no preference is indicated, the default is Execute Commands.

5.11.3 Logging

This indicates whether logging should be used. Logging keeps a record on disk of all commands sent to LSM or the system by the Visual Administrator. Logging can be started (Start) or stopped (Stop) at any time. When started, a window requesting a log file name appears and a file name must be entered; the Apply button must then be selected to start logging with the designated file. If no preference is indicated, the default is Stop and no logging is in effect.

5.11.4 Pop Up the Command Window

This option is used to bring up the Command Info Window on demand. This window displays the command history, along with the status and results of those commands. Commands can also be executed or repeated from this window. The Command Info Window is discussed in more detail in a later section.

5.11.5 Format of Size

This specifies the units in which size-related output should be displayed (megabytes, kilobytes, or sectors). The unit of size applies to output only and is set to megabytes until the user resets it.

In properties forms, length values are displayed as a number followed by an `s`, `m`, or `k` (representing sectors, megabytes, or kilobytes, respectively). If the size cannot be cleanly converted into megabyte or kilobyte format, it is displayed in sectors instead.

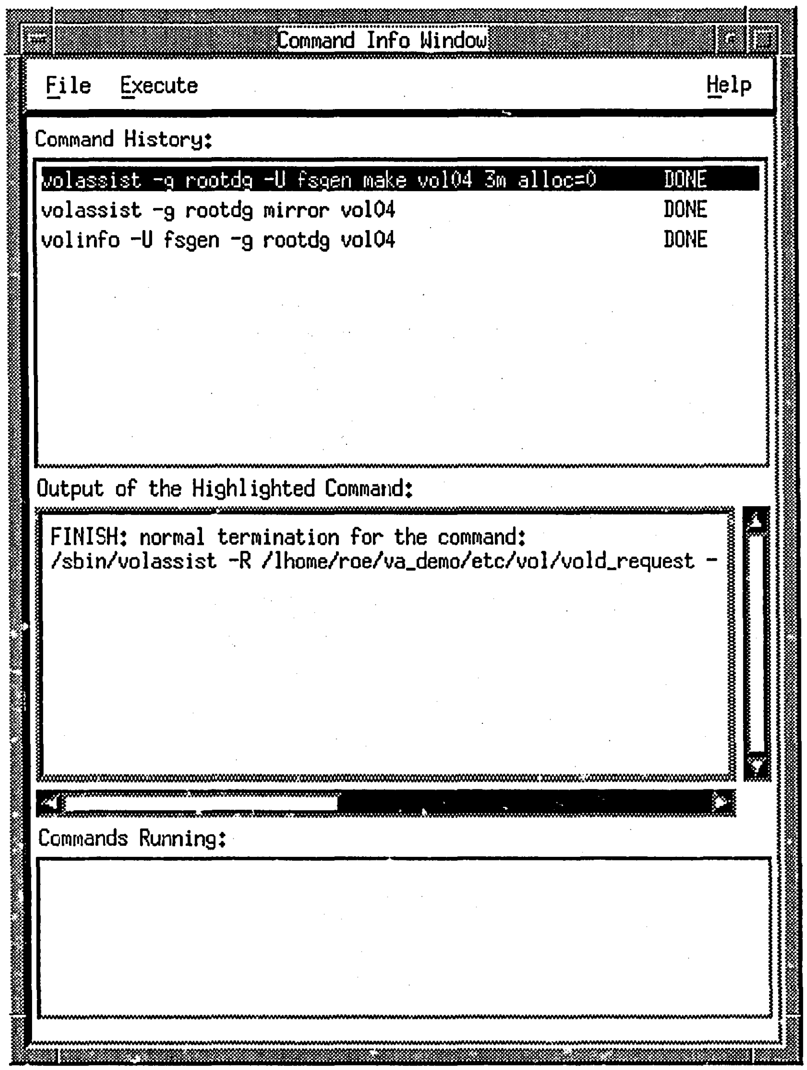
The preferred unit of size does not apply to input. Input typically defaults to sectors, unless megabytes or kilobytes are specified.

5.12 Command Info Window

The Command Info Window displays commands that are currently being executed by the Visual Administrator, as well as previous commands. Both LSM and system commands are displayed in this window as they are invoked by the Visual Administrator. The status and output of these commands is also displayed here. Previously-executed commands can be executed again directly from this window.

Figure 5-8 illustrates the Command Info Window.

Figure 5-8: Command Info Window



5.12.1 Command Info Sections

This window is divided into three sections:

- Command History

- Output of the Highlighted Command
- Commands Running

5.12.1.1 Command History

This section displays a chronological listing (or history) of the commands sent to LSM or the system for execution. The last ten commands are saved and displayed, with the most recent command at the bottom. If a command is too long, only the first few arguments of the command are shown. Each of these commands is listed along with its status, which is displayed to the right of the command. The status of a command is one of the following:

Status	Description
DONE	Command successfully completed
BROWSE	Command not executed, just displayed here
ERROR	Command terminated with error condition
UNKNOWN	Command status cannot be determined by the Visual Administrator (this rarely occurs, but generally results from an internal Visual Administrator error or a command being interrupted unexpectedly)

If a command is selected in this window, it is shown in its entirety in the middle section of this window, along with its output.

5.12.1.2 Output of the Highlighted Command

When a command is highlighted in the Command History section, its information is displayed in the Output of the Highlighted Command section of the window. The results of the command are indicated here (regardless of whether it succeeded or failed), along with the full command. If the command terminated abnormally or exited with an error condition, the error message is also displayed.

5.12.1.3 Commands Running

This section of the window displays the command that is currently running. This command has been sent to the system or LSM, but has not yet terminated. As soon as the command completes, it disappears from this section of the window.

5.12.2 Using the Command Info Window

The Command Info Window is accessed by selecting Popup the Command Window from the Options menu. This window may also display automatically when an error occurs with a command or operation. The Command Info Window can be safely closed by selecting Close from its File menu.

A command can be executed directly from the Command Info Window, as follows:

1. Select a command by clicking the MB1 button on the desired command in the Command History window. Once chosen, the command is highlighted.
2. Access the Execute menu and choose the Execute option.

This procedure sends the selected (highlighted) command to the system or LSM for execution. This is useful for executing a command again, reexecuting a failed command that should now succeed, or executing a command that was only shown (in BROWSE status) before.

In some circumstances, the Execute with Force option may be used rather than Execute. This option adds `-f` to the executed command to force LSM to complete an operation that is considered unsafe and to disregard error messages. The `-f` option is available with some LSM commands only and does not apply to file system operations.

Note

> The Execute with Force option is a *very* dangerous operation, which can result in irreparable loss of data; it should only be used when the user is sure that an operation should succeed, even though LSM error checking prevents it.

5.13 Administrative Operations

The Visual Administrator can be used to perform several types of volume management and other administrative operations. Most of these operations can be invoked from the Basic-Ops and Advanced-Ops menus in the menu bars of the views windows.

5.13.1 Basic Versus Advanced Operations

Although both the basic and advanced operations menus permit volume-related operations, it is important to note that there are two distinct approaches to volume management:

- Basic

  The basic approach typically makes use of the `volassist` utility, which is the LSM one-step automated interface. Because the `volassist` command operates automatically, it takes information about what the user wants to accomplish and then performs the necessary underlying tasks. This approach requires only minimal input from the user, but does permit more detailed specifications.

  You access basic operations via the Basic-Ops menu.

- Advanced

  The advanced approach typically makes use of the LSM command set. The LSM command set consists of a number of fairly complicated commands that typically require the user to specify detailed input. The LSM commands use a building block approach that requires the user to have a detailed knowledge of the underlying LSM structure and components to manually perform the sequences of commands necessary to accomplish a certain task.

  You access advanced operations via the Advanced-Ops menu.

Some operations are available only using one approach, while others can be handled through either the basic or advanced menu.

Note

When creating and manipulating volumes, first select which approach to use and then proceed with that approach only. Do not combine the basic and advanced approaches to perform a single volume operation. Users are advised to opt for the basic approach whenever possible.

5.13.1.1 Basic Operations

Basic operations are used by both LSM and the Visual Administrator to provide for easier system administration. The associated commands are designed to allow the user to perform administrative operations relatively quickly and easily. Basic operations use free space management techniques to automatically locate available space for the desired operation, unless the user specifies some suitable space.

The basic operations available via the Basic-Ops menu are as follows:

Operation	Description
File System Operations	Operations involving general file system maintenance. File systems can be created, mounted, and unmounted.

Operation	Description
Volume Operations	Operations involving general volume maintenance. Volumes can be created from LSM disks, removed, mirrored, resized, and backed up.

5.13.1.2 Advanced Operations

Advanced operations allow the user to manipulate LSM entities manually. It is also necessary to use the Advanced-Ops menu for those operations that are not available via the Basic-Ops menu.

Note

Users should not attempt to create and manipulate volumes using the Advanced-Ops menu unless they have a solid understanding of LSM concepts.

The advanced operations available via the Advanced-Ops menu are as follows:

Operation	Description
Volume	Perform operations on one or more volumes. These include creating a new volume, removing a volume, and changing the state of an existing volume.
Plex	Perform operations on one or more plexes. These include creating a new plex, removing a plex, associating a plex with a volume, disassociating a plex from a volume, and detaching a plex.
Subdisk	Perform operations on one or more subdisks. These include creating a new subdisk, removing a subdisk, associating a subdisk with a plex, disassociating a subdisk from a plex, and joining/splitting subdisks.
Disk Group	Perform various operations on disk groups. These include creating a new disk group, adding disks to a disk group, and removing disks from a disk group.
Disk	Perform various operations on disks. These include initializing a disk, removing a disk from LSM control, and onlining or offlining a disk.

5.13.2 Free Space

For operations that require free disk space, LSM and the Visual Administrator usually allow the user to designate disks with sufficient free space. If no disks are specified, disks with available space are automatically used if an operation is performed via Basic-Ops. With Advanced-Ops, however, users are expected to designate disks with sufficient space.

Before designating a disk to be used, the amount of free space on that disk should be verified. To check the available space on a disk, click the MB3 button on a portion of the LSM disk icon that is not covered by another icon. The disk properties form appears. At the bottom of this form is a field that indicated the maximum free space available for use on this disk.

Menu Interface 6

This chapter describes the LSM Support Operations designed to help you perform LSM disk and disk group operations in the menu interface. You enable LSM Support Operations using the `voldiskadm` command, which starts up a menu-driven interface. The menus are easy to use and provide information about each step to help you decide the correct response for each prompt.

The following sections describe how to start up and use the LSM Support Operations menu interface for some of the most common disk management tasks.

6.1 Starting LSM Support Operations

To start LSM Support Operations, enter the following command:

```
% voldiskadm
```

LSM brings up a Support Operations main menu shown in Figure 6-1 that displays the disk operation options available to you.

Figure 6-1: LSM Support Operations Main Menu

```
Logical Storage Manager Support Operations
Menu LogicalStorageManager/Disk

1     Add or initialize a disk
2     Encapsulate a disk
3     Remove a disk
4     Remove a disk for replacement
5     Replace a failed or removed disk
6     Mirror volumes
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
list  List disk information
?     Display help about menu
q     Exit from menus

Select an operation to perform:
```

Table 6-1 describes the main menu selections.

Table 6-1: Selections on the Main Menu

Option	Description
1	Identifies a disk to LSM and prepares the disk for LSM use. Use this option to initialize disks that are new or that do not contain valid data.
2	Prepares a disk for LSM usage while preserving existing data on the disk.
3	Removes a disk from LSM. This option does not retain the disk name for future disk replacement.
4	Removes a failed disk, retaining the disk name, and replaces it with another disk.
5	Specifies a replacement disk for a disk that either failed or was removed using menu option 4.
6	Duplicates data to give you one or more copies on another disk.
7	Moves data on a disk to another disk or disks on the system.

Table 6-1: (continued)

Option	Description
8	Enables access by this system to a disk group.
9	Disables access to a disk group that is currently enabled (imported) by this system.
10	Places the disk access record in an online state.
11	Places the disk access record in an offline state.
list	Lists disks that are available to the system.
?	Provides help when using the LSM menus. The output of the ? command is a list of operations and a definition of each.
q	Enter the q command: • From any menu to return to the main menu. • From the main menu to exit the menu interface.

6.2 Disk Operations

This section describes the disk operations available with the LSM Support Operations.

6.2.1 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.

To initialize a disk for use with LSM, perform the following steps:

1. At the main menu prompt (shown in Figure 6-1), select menu item 1 to enable the Add or initialize a disk operation.
2. At the prompt on the following Add or initialize a disk screen, enter the address of the disk to be added.

```
Add or initialize a disk
Menu:  LogicalStorageManager/Disk/AddDisk

Help information about this screen
        :
Select disk device to add [<disk/partition name>, list, q, ?]
```

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 available. For example:

DEVICE	DISK	GROUP	STATUS
rz1	disk00	rootdg	online
rz2	–	–	online
rz3	–	–	error

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)
```

Note that you can press the Return key to accept the default disk group name, `rootdg`. You can also enter the word none if either of the following conditions are true:

– 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 screen:

```
The requested operation is to initialize disk rz3 and
to leave this disk free for use as a spare disk.

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

– If you selected `rootdg` as the disk group, LSM displays the following screen:

```
The requested operation is to initialize disk device rz3 and
to add this device to disk group rootdg as disk disk01.

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

Press the Return key to continue.

5. The following screen displays when LSM has successfully completed the disk initialization.

```
Disk initialization for rz3 completed successfully.

Add or initialize another disk? [y,n,q,?] (default: n)
```

6. Press the Return key to return to the main menu.

6.2.2 Displaying Disk Information

The following steps describe how to find information about disks available on the system.

1. From the main menu (see Figure 6-1), enter the letter `l` or 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.

```
DEVICE      DISK      GROUP     STATUS

rz3         disk00    rootdg    online
rz4         -         -         online
rz5         -         -         online

Device to list in detail [<disk/partition name>,none,q,?] (default: none)
```

2. The following screen displays information for the disk device `rz3`:

```
Device:     rz3
devicetag:  rz3
type:       sliced
hostid:
disk:       name= id=723605502, 1095.mysys
group:      name= id=
flags:      online ready private autoconfig
pubpaths:   block = /dev/rz3g char = /dev/rrz3g
privpaths:  block = /dev/rz3h char = /dev/rrz3h
version:    1.1
iosize:     512
public:     slice = 13 offset = 0 len = 660992
private:    slice = 14 offset = 0 len = 512
update:     time = 723605503 seqno = 0.1
header:     0 248
configs:    count = 2 len = 173
logs:       count = 2 len = 26
Defined regions:
  Config priv 000017-000189[000173]: copy = 01 offset = 000000

  log priv 000190-000215[000026]: copy = 01 offset = 000000
  log priv 000296-000321[000026]: copy = 02 offset = 000000
  config priv 000322-000494[000173]: copy = 02 offset = 000000

List another disk device? [y,n,q,?] (default: n)
```

3. Press the Return key to return to the main menu.

6.2.3 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.

You can add a disk to a disk group by performing these steps:

1. Follow the instructions documented in Section 6.2.1.
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 has been initialized, LSM

displays the following screen and asks whether or not you want to reinitialize the disk:

```
Disk device rz3 appears to have been initialized already.
The disk is currently available as a replacement disk.


Do you wish to reinitialize? [y,n,q,?] (default: y) n
```

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.

6.2.4 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 6.2.5).

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, using two consecutive move operations could move volumes from the second disk to the first.

To move volumes from a disk, make sure that the target disks have sufficient space, then do the following:

1. Select menu item 7 from the main menu (shown in Figure 6-1).

2. From the Move volumes from a disk screen, enter the name of the disk whose volumes you want to move:

```
Move volumes from a disk
Menu: LogicalStorageManager/Disk/Evacuate

Help information about this screen
      :

Enter disk name [<disk>, list, q,?]   disk01
```

3. Verify the information that LSM displays on the following screen:

```
  Requested operation is to move all volumes from disk
  disk01 in group rootdg

  NOTE: This operation can take a long time to complete.
Continue with operation? [y,n,q,?] (default: y)
```

   Press Return to move the volumes.

4. As LSM moves the volumes from the disk, it displays the status of the operation:

```
Move volume voltest ...

Move volume voltest-bk00 ...
```

5. When the volumes have all been moved, LSM displays the following success screen:

```
Evacuation of disk disk01 is complete.

Move volumes from another disk? [y,n,q,?] (default: n)
```

Press Return to return to the main menu.

6.2.5 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. LSM Support Operations provides two methods (menu items 3 and 4 on the main menu shown in Figure 6-1) 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 also Section 6.2.4 which describes how to move data from a disk to another disk on the system, and see Section 6.2.6 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 6.3.2.

6.2.5.1 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 (shown in Figure 6-1).
2. LSM displays the following Remove a disk screen and prompts you to enter the disk name of the disk to be removed.

```
Remove a disk
Menu: LogicalStorageManager/Disk/RemoveDisk
Help information about this screen
        :
Enter disk name [<disk>, list, q, ?] disk02
```

This example removes the disk device `disk02`.

3. LSM displays a verification screen and asks whether or not to continue:

```
Requested operation is to remove disk disk02 from group newdg.

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 disk02 is complete.

Remove another disk? [y,n,q,?] (default: n)
```

Press the Return key to return to the main menu.

6.2.5.2 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 (shown in Figure 6-1). LSM displays the Remove a disk for replacement screen.

```
Remove a disk for replacement
Menu: LogicalStorageManager/Disk/RemoveForReplace

Help information about this screen
        :
        :
Enter the disk name [<disk>, list, q, ?] l
```

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:

```
DM NAME      DEVICE      TYPE     PRIVLEN   PUBLEN   PUBPATH
dm disk00    rz2         sliced   512       143632   /dev/rrz2g
dm disk01    rz4         sliced   512       660992   /dev/rrz4g

Enter disk name [<disk>,list,q,?] disk01
```

3. 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.

```
The following devices are available as replacements:

rz3

You can choose one of these disks now, to replace disk01.
Select "none" if you do not wish to select a replacement disk.

Choose a device, or select "none"
[<device>,none,q,?] (default: rz3)
```

Press the Return key if you want to use the default disk.

4. LSM then displays the following verification screen:

```
Requested operation is to remove disk disk01 from group rootdg.
The removed disk will be replaced with disk device rz3.

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

Press the Return key to continue.

5. When LSM successfully replaces the disk, LSM displays the following screen:

```
Removal of disk disk01 completed successfully.
Proceeeding to replace disk01 with device rz3.
Disk replacement completed successfully.
Remove another disk? [y,n,q,?] (default: n)
```

Press the Return key to return to the main menu.

6.2.6 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 6.2.5.2), 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 (shown in Figure 6-1).
2. The following Replace a failed or removed disk screen asks you to enter the name of the disk to be replaced. You can choose an uninitialized disk, or you can choose a disk that you have already initialized (skip to the next step).

```
Replace a failed or removed disk
Menu:  LogicalStorageManager/Disk/ReplaceDisk

Help information about this screen
          :

Select a removed or failed disk  [<disk>, list, q,?]  disk01
```

3. If there are any initialized disks available that are not part of a disk group, LSM displays the following screen to list the disks available to be used as a replacement. Select the replacement disk from the list provided.

```
The following devices are available as replacements:

rz3

You can choose one of these disks now, to replace disk01.
Select "none" if you do not wish to select a replacement disk.

Choose a device, or select "none"
[<device>,none,q,?] (default: rz3)
```

   Press Return to select the default device or enter the device name of the device of your choice.

4. LSM displays a confirmation screen:

```
The requested operation is to use the initialized device rz3
to replace the removed or failed disk disk01
in disk group rootdg.

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

   Press Return to replace the disk.

5. LSM displays the following success screen:

```
Replacement of disk disk01 in group rootdg with disk
device rz3 completed successfully.

Replace another disk? [y,n,q,?] (default: n)
```

Press Return to return to the main menu.

6.2.7 Renaming a Disk

To rename a disk, perform the following steps:

1. Perform the steps to remove the disk as described in Section 6.2.5.
2. Select menu item 1 from the main menu (shown in Figure 6-1) to bring up the Add or initialize a disk menu.
3. Select the address of the disk you just removed. This is the disk that you will add with a new name:

```
Add or initialize a disk
Menu: LogicalStorageManager/Disk/AddDisk

Help information about this screen
        :
Select disk device to add [<disk/partition name>, list, q, ?] rz3
```

4. LSM displays a screen that tells you the device is already initialized, and is available as a replacement disk, and asks whether or not you want to reinitialize the disk. Press the Return key to reinitialize the disk. Otherwise, enter `n` to avoid reinitialization.
5. Enter the name of the disk group in which you want to include the disk. In the following example, assume that the disk group, `newdg`, has already been created.

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

6. Enter the new disk name. The following example uses `disk03`:

```
Enter disk name: [<diskNN>,*,q,?] (default: disk02) disk03
```

7. LSM then displays the following verification screen:

```
The requested operation is to add disk device rz3 to disk
group newdg as disk disk03.

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

8. Press the Return key if the information is correct and you want LSM to continue.

9. Look for a message screen similar to the following that displays when LSM finishes the renaming operation:

```
Remove a disk
Menu: VolumeManager/Disk/RemoveDisk
Help information about this screen

Enter disk name [<disk>, list, q, ?] disk02
```

Press the Return key to return to the main menu.

6.2.8 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 (shown in Figure 6-1) to Disable (offline) a removable disk.
2. On the next screen, select the disk you want to disable:

```
Disable (offline) a disk device
Menu: LogicalStorageManager/Disk/OfflineDisk

Help information about this screen
        :

Select a disk device to disable [<disk/partition name>, list, q, ?] rz3
```

   This example shows that disk `rz3` has been selected.

3. LSM disables disk `rz3` 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.

6.3 Disk Group Operations

This section describes the disk group operations that can be performed with the LSM Support Operations.

6.3.1 Importing (Enabling) a Disk Group

Use this menu operation to enable access by this system 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 6.3.2), 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 (shown in Figure 6-1). From the Enable access to (import) a disk group menu, select the name of the disk group to import:

```
Enable access to (import) a disk group
Menu: LogicalStorageManager/Disk/EnableDiskGroup

Help information about this screen
        :

Select disk group to import [<group>, list, q,?] (default: list) newdg
```

2. Once the import is complete, LSM displays the following success screen:

```
    The import of newdg was successful.

  Select another disk group? [y,n,q,?] (default: n)
```

Press Return to return to the main menu

6.3.2 Deporting (Disabling) 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 (shown in Figure 6-1).
2. From the following Remove access to (deport) a disk group menu, enter the name of the disk group to be deported:

```
Remove access to (deport) a disk group
Menu: LogicalStorageManager/Disk/CreateDiskGroup

Help information about this screen
        :

Enter name of disk group [<group>, list, q,?] (default: list) newdg
```

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 newdg. This group is stored
on the following disks:
  newdg on device rz3
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)
```

Press Return to deport the disk group.

4. On the next screen display, verify that you want to continue with the operation:

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

Press Return to continue.

5. Look for the following screen that displays once the disk group is deported:

```
Removal of disk group newdg was successful.

Disable another disk group? [y,n,q,?] (default: n)
```

Press Return to go to the main menu.

6.4 Volume Operations: Mirroring Volumes on a Disk

Mirroring the volumes on a disk gives you one or more copies of your volumes in another disk location. By creating mirror copies of your volumes, you protect yourself against loss of data in case of a disk failure.

LSM cannot mirror volumes that are already mirrored, or if they 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 (shown in Figure 6-1).
2. On the Mirror volumes on a disk menu, enter the name of the disk whose volumes you want to mirror:

```
Mirror volumes on a disk
Menu: LogicalStorageManager/Disk/Mirror

Help information about this screen
      :
      :
Enter disk name [<disk>, list, q,?] disk02
```

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 disk02 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 [<disk>, list, q,?] (default: any) disk01
```

4. LSM displays the verification screen:

```
The requested operation is to mirror all volumes on disk disk02
in disk group rootdg onto available disk space on disk disk01.

NOTE: This operation can take a long time to complete.

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

Press Return to make the mirror.

5. LSM displays the status of the operation as it performs the mirroring:

```
Mirror volume voltest-bk00 ...
Mirroring of disk disk01 is complete.

Mirror volumes on another disk? [y,n,q,?] (default: n)
```

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.

6.5 Exiting LSM Support Operations

When you have completed all of your disk administration activities, exit the LSM Support Operations by selecting menu option q from the main menu.

Command Line Interface 7

This chapter presents some of the most frequently-used administrative commands provided by LSM. The discussions are organized within system administration for disk and disk group operations, volume operations, and file system operations. Each command is accompanied by the appropriate procedures necessary to complete an operation.

For details on any of the commands mentioned in this chapter, refer to the appropriate reference pages.

The functions described in this chapter can also be performed using the LSM Support Operations menu interface or the graphical user interface, dxlsm.

7.1 Commands and Utilities

Table 7-1 introduces the commands and utilities discussed in this chapter.

Table 7-1: Commands and Utilities

Utility	Description
`volassist`	A one-step utility that finds space for and creates simple volumes, adds simple plexes to existing volumes, extends and shrinks existing volumes, provides for the migration of data from a specified disk, and provides facilities for the online backup of existing volumes. The command is supplied a keyword that selects the action to perform.
`voldisk`	Defines special disk devices, initializes information stored on disks that LSM uses to identify and manage disks, and performs additional special operations.
`voldiskadd`	Adds standard disks to LSM. This utility leads you through the process of initializing a new disk by displaying information and asking questions.

Table 7-1: (continued)

Utility	Description
`voldiskadm`	Invokes the LSM Support Operations menu that allows you to perform both physical disk administration and volume and file system administration. Each entry in the main menu leads you through a particular operation by providing you with information and asking you questions. Default answers are provided for many questions so that common answers can be selected easily. This chapter describes uses for the `voldiskadm` utility but does not describe its use in detail. For more information about `voldiskadm` refer to chapter titled "Menu Interface Operations."
`voldg`	Creates new disk groups, adds and removes disks from disk groups, and enables (imports) or disables (deports) access to disk groups.
`voledit`	Sets and changes various attributes for LSM configuration records that do not depend upon volume usage types. (See the `volume` command utility to set attributes that are dependent upon usage types.) Each invocation can be applied to only one disk group at a time.
`volevac`	Moves subdisks off the specified disk *medianame* to the specified destination disks *new_medianame*. If *new_medianame* operands are not specified, then any nonvolatile, nonreserved disks can be used as destination disks. Subdisks that are part of unmirrored striped plexes will be moved by moving the entire plex to a new location.
`volinfo`	Reports a usage-type-dependent condition on one or more volumes in a disk group. A report for each volume specified by the volume operand is written to the standard output. If no volume operands are given, then a volume condition report is provided for each volume in the selected disk group.
`volmake`	Creates subdisks, plex, and volume records.
`volplex`	Performs LSM operations on plexes and volume-and-plex combinations. The `volplex` utility takes several operands. The first operand is a keyword that determines the specific operation to perform. The remaining operands specify the configuration objects to which the operation is to be applied. Each operation can be applied to only one disk group at a time.
`volprint`	Displays complete or partial information from records in LSM disk group configurations. Records can be selected by name or with special search expressions. Additionally, record association hierarchies can be displayed in an orderly fashion so that the structure of records is more apparent.

Table 7-1: (continued)

Utility	Description
`volrecover`	Recovers plexes and volumes after disk replacement.
`volsd`	Performs LSM operations on subdisks and on plex and subdisk combinations. This utility can associate or disassociate each named subdisk operand with a specific plex, split subdisks into two, or combine subdisks to form a new subdisk, move the contents of an old subdisk to a new one, and reduce the system impact of copy operations by throttling copy I/O operations.
`volstat`	Prints and resets statistics information about one or more volumes, plexes, subdisks, or disks. The volstat utility reads statistics from the volume device files in the directory `/dev/rvol` and prints them to standard output. These statistics represent volume, plex, subdisk, and disk activity since boot time. If no object (volume, plex, subdisk, or disk) operands are given, then LSM reports statistics from all volumes in the configuration database.
`volume`	Performs LSM operations on volumes, including initialization, set the read policy (round robin, preferential, select a default policy based on plex association), enable disabled or detached volumes, stop volumes, and change volume characteristics.

In addition, the `mount`, `unmount`, and `newfs` commands are described for the purpose of making and mounting a file system on an LSM volume.

7.2 Disk and Disk Group Operations

This section summarizes how to perform various disk and disk group operations. Many of these operations can be performed more easily using the LSM Support Operations menus (`voldiskadm`) discussed in Chapter 6. The `voldiskadm` menus are recommended for users new to LSM.

7.2.1 Adding a Disk

Add a disk by entering the command:

voldiskadd *devname*

To add the device rz3 to LSM control, do the following:

1. Enter:

   voldiskadd rz3

2. `voldiskadd` displays the following screen:

```
Add or initialize a disk
Menu: Logical Storage Manager/Disk/AddDisk

  Use this operation to add a disk to a disk group.  You can
  select an existing disk group or create a new disk group.
  You can also initialize a disk without adding it to a disk
  group, which leaves the disk available for use as a
  replacement disk. This operation takes, as input, a disk
  device, for example rz3, a disk group (or none to leave the
  disk available for as a replacement disk).  If you are
  adding the disk to a disk group, you will be asked to
  give a name to the disk.
```

3. LSM prompts you for a disk group name.

```
  You can choose to add this disk to an existing disk group,
  to create a new disk group.  To create a new disk group,
  select a disk group name that does not yet exist.

Which disk group [<group>,list,q,?] (default: rootdg)
```

   Press Return to assign the disk to the default disk group `rootdg`. Otherwise, enter the name of the disk group to which you want the disk assigned or enter `none` to assign the disk as a spare.

   Note

   You can create other disk groups if necessary. (Instructions for creating a disk group are included later in this chapter.) However, for most systems, the use of disk groups is not necessary. Therefore, LSM supports the default disk group, `rootdg`, in which all volumes are created if no further specification is given. All commands default to `rootdg` as well.

4. LSM now prompts you for a disk name (unless you entered `none` for a disk group, because spare disks do not get named).

```
  You must now select a disk name for the disk.
  This disk name can be specified to disk removal,
  move, or replacement operations.  If you move the
  disk, such as between host bus adapters, the disk
  will retain the same disk name, even though it will
  be accessed using a different disk device address name.

Enter disk name [<name>,q,?] (default: disk01)
```

   If you are initializing a new disk or reinitializing a previously-partitioned disk and selected the default disk group and disk name, the following screen appears:

```
  The requested operation is to initialize disk device rz3
  and to add this device to disk group rootdg as disk disk01.

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

In either case, press Return if you want to continue with the chosen operation.

5. LSM returns the following display to notify you that the initialization was a success.

```
 Disk initialization for rz3 completed successfully.

Add or initialize another disk? [y,n,q,?] (default: n)
```

Press Return to exit `voldiskadd`.

7.2.2 Reserving Disks

By default, `volassist` operations will allocate space from any disk that has free space. You may want to reserve some set of disks for special purposes, such as to avoid general use of a particularly slow or a particularly fast disk. To reserve a disk for special purposes, enter:

voledit set reserve=yes *diskname*

After you enter this command, `volassist` will not allocate space from the selected disk unless that disk is specifically mentioned on the `volassist` command line. For example, if disk `disk03` is reserved, the following command creates a 20 megabyte volume on `disk03`:

volassist make vol03 20m disk03

However, this command does not use `disk03`, even if there is no free space on any other disk:

volassist make vol04

To turn off reservation of a disk, enter:

voledit set reserve=no *diskname*

7.2.3 Display Disk Information

Before you use a disk, you need to know if it has been initialized. You need to know if the disk is part of a disk group, since you cannot create volumes on a disk that is not part of a disk group. The `list` command displays device names for all recognized disks, the disk names, the disk group names associated with each disk, and the status of each disk.

Display information on all disks that are defined to LSM as follows:

voldisk list

LSM returns the following display:

```
DEVICE    TYPE      DISK          GROUP         STATUS
rz8       sliced    rz8           rootdg        online
rz9d      simple    -             -             online
```

Display details on a particular disk defined to LSM as follows:

voldisk list rz8

7.2.4 Rename a Disk

It is not necessary to give your disks special names. LSM gives the disk a default name when you add the disk to LSM control. The disk name is used by LSM to identify the disk's location or type. If you want to change the disk name to reflect a change of ownership or use, enter:

voledit rename *old_diskname new_diskname*

To rename disk03 to disk01 enter:

voledit rename disk03 disk01

To see if the name change took place, enter:

voldisk list

LSM displays the following:

```
DEVICE        TYPE      DISK          GROUP         STATUS
rz8           sliced    rz8           rootdg        online
rz9           sliced    disk01        rootdg        online
rz9d          simple    -             -             online
```

7.2.5 Initialize a New Disk Group

There can be situations in which all data related to a particular set of applications of a particular group of users needs to be made accessible on another system. Examples of this are:

- A system has failed and its data need to be moved to other systems.
- The work load must be balanced across a number of systems.

In such cases, it is important that the data related to particular applications or users be located on an identifiable set of disk drives, such that when these drives are moved, all data of the applications or group of users is moved and no other data.

Note

> On most systems, the use of disk groups is not necessary. Therefore, LSM supports a default disk group, `rootdg`, in which all volumes are created if no further specification is given. All commands will default to `rootdg` as well.

To create the disk group `newdg` associated with disk02, enter the following command:

voldiskadd rz9d

1. Because rz9d has already been initialized, LSM asks if you want to reinitialize it.

```
Add or initialize a disk
Menu: Logical Storage Manager/Disk/AddDisk

  Use this operation to add a disk to a disk group.
  You can select an existing disk group or create
  a new disk group. You can also initialize a disk
  without adding it to a disk group, which leaves
  the disk available for use as a replacement disk.
  This operation takes, as input, a disk device,
  for example rz3, a disk group (or none to leave
  the disk available for as a replacement disk).
  If you are adding the disk to a disk group, you
  will be asked to give a name to the disk.

  Disk device rz9d appears to have been initialized
  already. The disk is currently available as a
  replacement disk.

Do you wish to reinitialize rz9d? [y,n,q,?] (default: y)n
```

   Enter `n` to add the disk to a disk group without reinitializing it.

2. LSM prompts you for a disk group.

```
  You can choose to add this disk to an existing disk
  group, to create a new disk group, or you can choose
  to 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)
newdg
```

   Enter the new disk group name and press Return .

3. LSM responds with:

```
  There is no active disk group named newdg.

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

   Press Return to continue.

4. LSM asks for a disk name.

```
  You must now select a disk name for the disk.  This disk
  name can be specified to disk removal, move, or
  replacement operations.  If you move the disk, such as
  between host bus adapters, the disk will retain the same
  disk name, even though it will be accessed using a
  different disk device address name.

Enter disk name [<name>,q,?] (default: disk02)
```

   Enter the disk name of your choice or press Return to select the default name.

5. LSM displays a confirmations window:

```
  The requested operation is to create a new disk group
  named newdg containing disk device rz9d.  The disk
  will be named newdg01 within the disk group.

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

   Press Return to continue.

6. Once the operation is complete, LSM returns the following display:

```
  Disk initialization for rz9d completed successfully.

Goodbye.
```

7. Enter the following command to see if the disk group was created:

   voldisk list

LSM displays this output which shows that `disk02` is a part of the disk group.

DEVICE	TYPE	DISK	GROUP	STATUS
rz8	sliced	rz8	rootdg	online
rz9	sliced	disk01	rootdg	online
rz9d	simple	disk02	newdg	online

7.2.6 Display Disk Group Information

To use disk groups, you need to know what they are and what disks belong to each group.

Display information on existing disk groups as follows:

voldg list

LSM returns the following:

NAME	STATE	ID
rootdg	enabled	730344554.1025.harley
newdg	enabled	731118794.1213.harley

7.2.7 Using Disk Groups

Most LSM commands allow a disk group to be specified using the `-g` option. For example, to create a volume in disk group `mktdg` you could use the command:

volassist -g mktdg make mktvol 50m

The volume device for this volume is:

/dev/vol/mktdg/mktvol

In many cases the disk group does not have to be specified. Most LSM commands use object names specified on the command line to determine the disk group for the operation. For example, a volume can be created on disk `mktdg01` without specifying the disk group name:

volassist make mktvol 50m mktdg01

This works for many commands as long as two disk groups do not have objects with the same name. For example, LSM allows you to create volumes named `mktvol` in both `rootdg` and in `mktdg`. If you do this, you must add `-g mktdg` to any command where you want to manipulate the volume in the `mktdg` disk group.

7.2.8 Display Free Space

Before you add volumes and file systems to your system, you may want to make sure you have enough free disk space to adequately meet your needs. LSM lets you request a display of free space.

To display free space for a group, enter:

voldg free *disk_group*

To see the free space in the default disk group, `rootdg`, enter:

voldg free

LSM displays the following:

```
GROUP    DISK     DEVICE   TAG    OFFSET  LENGTH   FLAGS
rootdg   rz8      rz8      rz8    726400  102672   -
rootdg   disk01   rz9      rz9    0       102128   -
```

The free space is measured in 512-byte sectors.

7.2.9 Disable (Offline) a Disk

Occasionally, you may need to disable (offline) a disk. If the disk is corrupted, you need to disable it and remove it. You also must disable a disk before moving the physical disk device to another location to be connected to another system.

To disable a disk, first remove the disk from its disk group. Then place a disk in an offline state as follows:

voldisk offline *devname*

To take the device rz8 offline, enter:

voldisk offline rz8

Note

The device name is used because the disk is no longer in a disk group and does not have an administrative name.

7.2.10 Remove a Disk

You can remove a disk to move it to another system or you may remove the disk because the disk is failing or has failed. However, before removing the disk from the current system, you must move the subdisks to a different physical disk (see `volevac`(8)). If there is not enough free space to move the subdisks, then you must:

- Unmount the file system (see later in this chapter).
- Stop the volumes (see later in this chapter).
- Back up the volumes to tape (see Section 7.3.8).

  Alternatively, if the volumes are no longer needed, they can be removed.

Removing a disk involves the following steps:

1. Remove the disk from its disk group.

   Remove a disk from a disk group as follows:

   voldg [-g *groupname***] rmdisk** *diskname*

   where:

groupname	is the name of the group to which the disk belongs.
diskname	is the name of the disk to be removed.

For example, to remove disk01 from `rootdg`, enter:

voldg rmdisk disk01

Because `rootdg` is the default disk group, you do not need to specify it.

2. Remove the disk from LSM and the system.

   After removing a disk from its disk group, remove it from the system as follows:

   voldisk rm *devname*

   For example, to remove rz9d from LSM control, enter:

   voldisk rm rz9d

7.2.11 Detecting Failed Disks

If one plex of a volume encounters a disk I/O failure (for example, because the disk has an uncorrectable format error), LSM may detach the plex. If a disk fails completely, LSM may detach the disk from its disk group. If a plex is detached, I/O stops on that plex but continues on the remaining plexes of the volume. If a disk is detached, all plexes on the disk are disabled. If there are any unmirrored volumes on a disk when it is detached, those volumes are disabled as well.

If a volume, a plex, or a disk is detached by failures, mail is sent to `root` indicating the failed objects. For example, if a disk containing two mirrored volumes fails you might receive a mail message like:

```
To: root
Subject: Logical Storage Manager failures on mobius.lsm.com

Failures have been detected by LSM on host
mobius.lsm.com:

failed plexes:
 home-02
 src-02

No data appears to have been lost.  However, you should replace
the drives that have failed.
```

To determine which disks are causing the failures in the above message, run the command:

volstat -sff home-02 src-02

This will produce output such as:

```
                                       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` (the basename for the displayed subdisks).

Sometimes these errors are caused by cabling failures. You should look at the cables connecting your disks to your system. If there are any obvious problems, correct them and recover the plexes with the command:

volrecover -b home src

This command will start a recovery of the failed plexes in the background (the command will return before the operation is done). If an error message appears later, or if the plexes become detached again, you should replace the disk.

If you do not see any obvious cabling failures, then the disk probably needs to be replaced.

If a disk fails completely, the mail message will list the disks that have failed, all plexes that use the disk, and all volumes defined on the disk that were disabled because the volumes were not mirrored. For example:

```
To: root
Subject: Logical Storage Manager failures on mobius.lsm.com

Failures have been detected by LSM on host
mobius.lsm.com:

failed disks:
 disk02

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

failed volumes:
 mkting

The contents of failed volumes may be corrupted, and should be
restored from any available backups.  To restart one of
these volumes so that you can restore it from backup, replace
disks as appropriate then use the command:

     volume -f start <volume-name>

You can then restore or recreate the volume.
```

This message indicates that `disk02` was detached by a failure; that plexes `home-02`, `src-02`, and `mkting-01` were also detached (probably because of the failure of the disk); and that the volume `mkting` was disabled because of the failure.

Again, the problem may be a cabling error. If the problem is not a cabling error then you must replace the disk.

7.2.12 Replacing Disks

Disks that have failed completely (that have been detached by failure) can be replaced by running `voldiskadm` and selecting item 5, `Replace a failed or removed disk` from the main menu. If there are any initialized but unadded disks, you will be able to select one of those disks as a replacement. Do not choose the old disk drive as a replacement; 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 then 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 command:

volinfo

Any volumes that are listed as `Unstartable` must be restored from backup. For example, `volinfo` might display:

```
home            fsgen    Started
mkting          fsgen    Unstartable
src             fsgen    Started
```

To restart volume `mkting` so that it can be restored from backup, use the command:

volume -obg -f start mkting

The `-obg` option causes any plexes to be recovered in a background task.

If failures are starting to occur on a disk, but the disk has not yet failed completely, you should replace the disk. This involves two steps: detaching the disk from its disk group and replacing the disk with a new one. To detach the disk, run `voldiskadm` and select item 4, `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 selecting item 5, `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:

        lhome 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 do this? [y,n,q,?] (default: n)
```

If any volumes would be disabled, you should quit from `voldiskadm` and save the volume. Either backup the volume or move the volume off of the disk. To move the volume `mkting` to a disk other than `disk02`, use the command:

volassist move mkting disk02

After the volume is backed up or moved, run `voldiskadm` again and continue to remove the disk for replacement.

After the disk has been removed for replacement, a replacement disk can be used by specifying item 5, `Replace a failed or removed disk,` from the main menu in `voldiskadm`.

7.2.13 Removing a Disk Group

To remove a disk group, close or unmount any volumes in the disk group and then run the command:

voldg deport *diskgroupname*

Deporting a disk group does not actually remove the disk group. It disables use of the disk group by the system. However, disks that are in a deported disk group can be reused, reinitialized, or added to other disk groups.

If you do not want to remove a disk group, but do want to reorganize by moving a disk between disk groups, remove the disk from one disk group and add it to the other. For example, to move the physical disk `rz3g` (attached with the disk name `disk04`) from disk group `rootdg` and add it to disk group `mktdg`, you could use the commands:

voldg rmdisk disk04
voldg -g mktdg adddisk mktdg02=rz3g

This can also be done using `voldiskadm` by selecting item 3, `Remove a disk,` from the main menu, and then selecting item 1, `Add or initialize a disk.`

7.2.14 Moving Disk Groups Between Systems

An important feature of disk groups is that they can be moved between systems. If all disks in a disk group are moved from one system to another, then the disk group can be used by the second system without having to respecify the configuration.

There are two steps in moving a disk group between systems.

On the first system:

1. Close and unmount all volumes in the disk group, then deport (disable local access to) the disk group with the command:

   voldg deport *diskgroupname*

2. Then, move all the disks to the second system, online the disk devices with the command:

voldisk online *diskdevicename ...*

then import (enable local access to) the disk group on the second system with the command:

voldg import *diskgroupname*

3. After the disk group is imported, start all volumes in the disk group with the command:

   volrecover -g *diskgroupname* **-sb**

You may want to move disks from a system that has crashed. In this case, you will not be able to deport the disk group from the first system. When a disk group is created or imported on a system, that system writes a lock on all disks in the disk group. The purpose of the lock is to ensure that dual-ported disks (disks that can be accessed simultaneously by two systems) will not be used by both systems at the same time. If two systems try to manage the same disks at the same time, configuration information stored on the disk will be corrupted and the disk will become unusable.

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. 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, use the command:

voldisk clearimport *diskdevicename ...*

Be careful when using this command on systems that really do have dual-ported disks.

In some cases, you may want to import a disk group when some disks are not available. The `import` operation normally fails if some disks for the disk group cannot be found among the disk drives attached to the system. If the `import` operation fails, one of the following error messages will display:

```
voldg: Disk group groupname: import failed: Disk for \
        disk group not found
```

or

```
voldg:Disk group groupname: import failed: Disk group has \
         no valid configuration copies
```

If some of the disks in the disk group have failed, you can force the disk group to be imported with the command:

voldg -f import *diskgroupname*

7.3 Volume Operations

This section summarizes various volume operations using the `volassist` command, which is the simplest and most powerful LSM command.

Note

You can create a file in `/etc/default/volassist` that contains default options for `volassist`. This file can be edited to change the default behavior of `volassist` on your system. See Section 8.1.1.3 for an example of this file.

The `volassist` command contains an online help program to help you use `volassist` effectively. To activate the online help, enter:

volassist help

The `volassist help` command displays the following information:

```
volassist - Perform simple general administrative actions

Usage: volassist [-D debugopt[,...]] [-g diskgroup] \
              [-U usetype] [-d file] -b keyword arg ...
Recognized keywords:
    make volume_name length [options]
    mirror volume_name [options]
    move volume_name [options]
    growto volume_name new_length [options]
    growby volume_name length_change [options]
    shrinkto volume_name new_length [options]
    shrinkby volume_name length_change [options]
    snapstart volume_name [options]
    snapwait volume_name
    snapshot volume_name snapshot_name
    help [debug | flags | options]
```

7.3.1 Create a Volume

A volume is a logical disk device on which file systems and databases can be created.

Create a volume using system defaults as follows:

volassist make *volume_name length*

To create the volume `voldef` enter:

volassist make voldef 10m

This creates a 10MB volume named voldef.

7.3.1.1 Concatenated Volume

A concatenated volume is a volume using one or more sections of disk space. On a fragmented disk, this allows you to put together a volume larger than any individual section of free disk space available. Create a concatenated volume as follows:

volassist make *volume_name length* **layout=concat**

To create the concatenated volume `volcat`, enter:

volassist make volcat 10m layout=concat

Concatenation can be set as the default in the `volassist` defaults file `/etc/defaults/volassist`. This file is described in Section 8.1.1.3.

7.3.1.2 Spanned Volume

A spanned volume is a concatenated volume with sections of disk space spread across more than one disk. A spanned volume can be larger than the single largest disk, since it takes space from more than one disk. Create a spanned volume as follows:

volassist make *volume_name length* **layout=concat,span**

The concatenated, spanned layout is the default.

To create the spanned volume `volspan`, enter:

volassist make volspan 1000m layout=concat,span

7.3.1.3 Striped Volume

A striped volume consists of a number of equal sized subdisks, each located on a separate disk drive. Create a striped volume as follows:

volassist make *volume_name length* **layout=stripe**

To create the striped volume `volzebra`, enter:

volassist make volzebra 10m layout=stripe

This creates a volume with the default stripe width on the default number of drives.

7.3.1.4 Volume on Specific Disk

LSM automatically selects the disk or disks each volume will reside on, unless you specify otherwise. If you want a volume to reside on a specific disk, you must designate the disk for LSM. Create a volume on a specific disk as follows:

volassist make *volume_name length diskname* **[...]**

More than one disk can be specified.

To create the volume `volspec` on disk03, enter:

volassist make volspec 3m disk03

7.3.2 Display Volume Configuration Information

If you are administering a volume created by someone else, you may want to know how this volume is configured.

Display the volume, plex, and subdisk record information for all volumes as follows:

volprint -ht

For example:

```
DG NAME     GROUP-ID
DM NAME      DEVICE        TYPE     PRIVLEN  PUBLEN PUBPATH
V  NAME      USETYPE       KSTATE   STATE    LENGTH READPOL...
PL NAME      VOLUME        KSTATE   STATE    LENGTH LAYOUT...
SD NAME      PLEX          PLOFFS   DISKOFFS LENGTH DISK-MEDIA...

dg rootdg      730344554.1025.harley

dm rz8         rz8         sliced   512    829072 /dev/rrz8e
dm disk01      rz8d        simple   512    202240 /dev/rrz8d
```

```
dm disk02       rz9          sliced   512    202240 /dev/rrz9e

pl volmir-02    -            DISABLED -      10240  CONCAT -   RW
sd disk02-02    volmir-02    0        16384  10240  disk02     rz9

v  homevol      fsgen        ENABLED  ACTIVE 275184 SELECT -
pl homevol-01   homevol      ENABLED  ACTIVE 275184 CONCAT -   RW
sd rz8-04       homevol-01   0        410256 275184 rz8        rz8

v  rootvol      root         ENABLED  ACTIVE 314496 SELECT -
pl rootvol-01   rootvol      ENABLED  ACTIVE 314496 CONCAT -   RW
sd rz8-01       rootvol-01   0        95760  314496 rz8        rz8

v  snapvol      fsgen        ENABLED  -      20480  ROUND  -
pl voltest-03   snapvol      ENABLED  ACTIVE 20480  CONCAT -   RW
sd disk01-01    voltest-03   0        143088 20480  disk01     rz8d

v  standvol     gen          ENABLED  ACTIVE 31248  SELECT -
pl standvol-01  standvol     ENABLED  ACTIVE 31248  CONCAT -   RW
sd rz8-03       standvol-01  0        0      31248  rz8        rz8

v  swapvol      swap         ENABLED  ACTIVE 64512  SELECT -
pl swapvol-01   swapvol      ENABLED  ACTIVE 64512  CONCAT -   RW
sd rz8-02       swapvol-01   0        31248  64512  rz8        rz8

v  volcat       fsgen        ENABLED  ACTIVE 20480  SELECT -
pl volcat-01    volcat       ENABLED  ACTIVE 20480  CONCAT -   RW
sd rz8-06       volcat-01    0        705920 20480  rz8        rz8

v  volspec      fsgen        ENABLED  ACTIVE 6144   SELECT -
pl volspec-01   volspec      ENABLED  ACTIVE 6144   CONCAT -   RW
sd disk02-01    volspec-01   0        10240  6144   disk02     rz8d

v  volzebra     fsgen        ENABLED  ACTIVE 20480  SELECT volstripe

pl volzebra-01  volzebra     ENABLED  ACTIVE 20480  STRIPE 128 RW
sd disk01-02    volzebra-01  0        102128 10240  disk01      rz9
sd rz8-01       volzebra-01  10240    0      10240  disk02      rz9

v  voltest      fsgen        ENABLED  ACTIVE 20480  SELECT -
pl voltest-01   voltest      ENABLED  ACTIVE 20480  CONCAT -    RW
sd rz8-05       voltest-01   0        685440 20480  rz8         rz8
pl voltest-02   voltest      ENABLED  ACTIVE 20480  CONCAT -    RW
sd disk01-04    voltest-02   0        122608 20480  disk01      rz8d
```

In the example, the following abbreviations are as follows:

Abbreviation	Stands for...
dg	Disk group
dm	Disk
pl	Plex

Abbreviation	Stands for...
sd	Subdisk
v	Volume

The syntax to display volume-related information for a specific volume is as follows:

volprint -t *volume_name*

To display the information about `volspec`, enter the following command:

volprint -t volspec

```
DG NAME  GROUP-ID
DM NAME  DEVICE    TYPE      PRIVLEN  PUBLEN   PUBPATH
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 volspec fsgen    ENABLED   ACTIVE   6144     SELECT   -
```

7.3.3 Mirror a Volume

A plex contains a copy of the volume's data. A volume must have two plexes for the data to be mirrored. The mirrored copy is not stored on the same disk as the original copy of the volume. Mirroring a volume assures you that the data in that volume will not be lost if one of your disks fails.

Create a new volume with a plex as follows:

volassist make *volume_name length* **mirror=yes**

To create the mirrored volume, `volmir`, enter:

volassist make volmir 5m mirror=yes

Create a plex for an existing volume as follows:

volassist mirror *volume_name*

The following example creates a plex of the volume `voltest`:

volassist mirror voltest

7.3.4 Remove a Plex

A plex contains a copy of the volume's data. A volume is mirrored when there are two or more plexes attached to the volume. If you no longer want to mirror a volume, you must disassociate the extra plex copy or copies.

Removing a plex involves the following procedure:

1. Disassociate the plex from the volume as follows:

   volplex dis *plex_name*

   For example, the volume `volmir` is a mirrored volume.

   ```
   TYPE NAME        ASSOC        KSTATE     LENGTH COMMENT
   vol  volmir      fsgen        ENABLED     10240
   plex volmir-01   volmir       ENABLED     10240
   sd   disk01-03   volmir-01    -           10240
   plex volmir-02   volmir       ENABLED     10240
   sd   disk02-02   volmir-02    -           10240
   ```

   To disassociate `volmir-02` from the parent volume, enter:

   volplex dis volmir-02

2. If you do not plan to use the plex and its subdisks again, you can remove them from the LSM configuration database.

After disassociating the plex from the volume, remove the plex and any associated subdisks as follows:

voledit -r rm *plex_name*

Remove `volmir-02` as shown below:

voledit -r rm volmir-02

The output of `volprint -h` for `volmir` is:

```
TYPE NAME        ASSOC        KSTATE     LENGTH COMMENT
vol  volmir      fsgen        ENABLED     10240
plex volmir-01   volmir       ENABLED     10240
sd   disk01-03   volmir-01    -           10240
```

Because the volmir volume has only one plex attached to it, volmir is no longer mirrored.

7.3.5 Extend a Volume

If the volume is not large enough for the amount of data that needs to be stored in it, you need to extend the volume's length.

Extend a volume *to* a specific length as follows:

volassist growto *volume_name length*

To extend `volcat` to 2000 512-byte sectors, enter:

volassist growto volcat 2000

Extend a volume *by* a specific length as follows:

volassist growby *volume_name length*

To extend `volcat` by 100 sectors, enter:

volassist growby volcat 100

Note

The `volassist growto` and `volassist growby` commands cannot be used with striped volumes.

7.3.6 Shrink a Volume

If you find that your volume is much larger than you really need it to be, you can shrink the volume's size. Shrink a volume *to* a specific length as follows:

volassist shrinkto *volume_name length*

This command can be safely used on empty volumes. To shrink `volcat` to 1300 sectors, enter:

volassist shrinkto volcat 1300

Shrink a volume *by* a specific length as follows:

volassist shrinkby *volume_name length*

To shrink `volcat` by 300 sectors, enter:

volassist shrinkby volcat 300s

Note

The `volassist shrinkto` and `volassist shrinkby` commands cannot be used with striped volumes.

7.3.7 Remove a Volume

Once a volume is no longer necessary (it is inactive and archived, for example), you can remove the volume and free up the disk space for other uses.

To remove a volume, all references to the volume must be closed (that is, the file system in the volume must be unmounted, and the volume stopped using the `volume stop` command). The `volume stop` command stops all LSM activity to the volume.

After stopping the volume, enter the following command:

voledit -rf rm *volume_name*

To remove the volume `volspan`, enter:

voledit -rf rm volspan

7.3.8 Reducing Backup Downtime Using volassist

LSM provides the ability to perform snapshot backups of volume devices. This capability is provided through the `volassist` and other utilities. There are various possible procedures for doing backups, depending upon requirements for integrity of the volume contents. These procedures have the same starting requirement: a plex that is large enough to store the complete contents of the volume. The plex can be larger than necessary, but if a smaller plex is used, an incomplete copy results. The recommended approach to volume backup involves the use of the `volassist` utility. The `volassist` procedure is convenient and relatively simple.

The `volassist snapstart`, `snapwait`, and `snapshot` operations provide a way to do online backup of volumes with minimal interruption of data change and access activity.

The `snapstart` operation creates a write-only backup plex which gets attached to and synchronized with the volume. When synchronized with the volume, the backup plex is ready to be used as a snapshot plex. The end of the update procedure is signified by the new snapshot plex changing its state to SNAPDONE. This change can be tracked by `volassist snapwait` operation, which waits until at least one of the `snapshot` plexes changes its state to SNAPDONE. If the attach process fails, the snapshot plex is removed and its space is released.

Once the snapshot plex is synchronized, it continues being updated until it is detached. The system administrator can then select a convenient time at which to create a snapshot volume as an image of the existing volume. The system administrator can also ask users to refrain from using the system during the brief time required to perform the snapshot (typically less than a minute). The amount of time involved in creating the snapshot plex is long and indefinite in contrast to the brief amount of time that it takes to create the snapshot volume.

The backup procedure is completed by running a `volassist snapshot` command on a volume with a SNAPDONE plex. 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 and the state of the snapshot is set to ACTIVE.

If the snapshot procedure is interrupted, the snapshot plex is automatically removed when the volume is started.

Follow these steps to perform a complete *`volassist`* backup:

1. Create a snapshot plex for a volume as follows:

   volassist snapstart *volume_name*

2. When the snapstart operation is complete and the plex is in a SNAPDONE state, select a convenient time to complete the snapshot operation. Inform users of the upcoming snapshot and warn them to save files and refrain from using the system briefly during that time.

   Note

   For UFS volumes, Digital recommends the file system be unmounted briefly to ensure the snapshot data on disk is consistent and complete.

3. Create a snapshot volume that reflects the original volume as follows:

   volassist snapshot *volume_name temp_volume_name*

   After the snapshot is completed, you can resume normal use of the volume. For example, if the volume contained a UFS, it can be remounted.

4. Use `fsck`, or some utility appropriate to the application running on the volume, to clean the temporary volume's contents. For example:

   fsck -p /dev/rvol/ *temp-volume-name*

5. Copy the temporary volume to tape, or to some other appropriate backup media. For example, a UFS on the temporary volume could be backed up

on the temporary volume with the following command:

dump 0 /dev/rvol/ *temp-volume-name*

6. After the back up has completed, remove the temporary volume as follows:

   voledit -rf rm *temp_volume_name*

Warning

The `-r` option of `voledit` removes multiple objects. Exercise caution when using it.

Advanced Volume Operations 8

This chapter describes advanced Logical Storage Manager (LSM) system administration using the LSM command line interface. Information is included about how to manipulate the configuration and create, remove, and maintain LSM records.

8.1 Individual Utility Descriptions

The following sections describe the LSM utilities that are most commonly used to perform system administration and maintenance functions. This section also describes the function of each command. Detailed information for each of these utilities can be found in their respective reference pages.

8.1.1 Using the volassist Command

The `volassist` command provides a convenient, one-step interface to LSM and is especially useful for basic and commonly used administrative operations. The `volassist` command automatically finds space for and creates simple volumes or plexes for existing volumes, resizes volumes, and provides online backup of volumes.

The `volassist` command accomplishes alone many tasks that would otherwise require the use of a sequence of several other LSM utilities. The `volassist` command does not conflict with the existing LSM utilities or preclude their use. Objects created by the `volassist` command are compatible and interoperable with objects created manually using the other LSM utilities and interfaces, and vice versa.

In general, it is more convenient to use the `volassist` command than a series of other LSM commands. Some of the advantages of using the `volassist` command include:

- The use of the `volassist` command involves only one step on the part of the user. The `volassist` command automatically takes care of all underlying and related operations (such as creating associated subdisks and plexes) that would otherwise need to be performed manually by the user through additional commands.
- The user is required to specify only minimal information to `volassist`, yet can optionally specify additional parameters to modify its actions.

- The volassist operations result in a set of configuration changes that either succeed or fail as a group, rather than individually. Most volassist operations therefore function in such a way that system crashes or other interruptions do not leave intermediate states that need to be cleaned up. If volassist encounters an error or some other exceptional condition, it will exit without leaving behind partially changed configurations; the system will be left in the same state as it was prior to the attempted volassist operation.

8.1.1.1 How the volassist Command Works

The volassist utility allows users to create and modify simple volumes. The user specifies the basic volume creation or modification requirements and volassist proceeds to perform all of the necessary underlying tasks.

The amount of information that volassist requires from the user is minimal because volassist obtains most of the information it needs from other sources. The volassist command obtains information about the existing objects and their layouts from the objects themselves. For operations requiring new disk space, volassist seeks out available disk space and tries to allocate it in the configuration that conforms to the layout specifications and offers the best use of free space.

8.1.1.2 The volassist Command Defaults

The volassist command invocation is designed to be as simple as possible, while allowing its behavior to be tailored when necessary. The volassist command uses a set of tunable parameters that you can specify in defaults files or on the command line. The tunable parameters default to reasonable values if they are not defined anywhere. Any tunable parameters listed on the command line override those specified elsewhere. The tunable parameters are specified as follows:

- Internal defaults

  The built-in defaults are used when the value for a particular tunable is not specified elsewhere (on the command line or in a defaults file).

- Systemwide defaults file

  The systemwide defaults file contains default values that may be altered by the system administrator. These values are used for tunables that are not specified on the command line or in the user's defaults file.

- User defaults file

  The user can create a personal defaults file. If a personal defaults file exists, the values therein are used for tunables that are not specified on the command line. These values override those in the systemwide defaults file.

- Command line

  The tunable values specified on the command line override any values specified internally or in defaults files.

8.1.1.3 Defaults File

The default behavior of `volassist` is controlled by the tunables specified in the `/etc/default/volassist` file that you must create. The format of the defaults file is a list of `attribute=`*`value`* pairs separated by new lines. These `attribute=`*`value`* pairs are the same as those specified as options on the command line.

The following is a sample `volassist` defaults file:

```
# LSM Vn.n
# volassist defaults file.  Use '#' for comments

# layout
layout=concat,noncontig,span

# mirroring
nmirror=2
mirror=no

# allocation policies
align=4k
alloc=20m

# striping
stripewidth=64k

# logging
logtype=none

# volume usage type
usetype=fsgen
```

For detailed information about how to use `volassist`, refer to the `volassist` reference page.

8.1.2 Using the voldctl Utility

The volume configuration daemon (`vold`) is the interface between the other LSM utilities and the kernel `volconfig` device driver. The `volconfig` device is a special device file created by LSM that interacts with `vold` to make LSM configuration changes.

Some `voldctl` operations involve modifications to the `volboot` file, which indicates the locations of root configuration copies.

The `voldctl` utility is the interface to `vold` and is used for:

- Performing administrative tasks related to the state of the daemon
- Managing boot information and various aspects of the LSM root configuration initialization
- Manipulating the contents of the `volboot` file, which contains a list of disks containing root configuration databases

For detailed information about how to use `voldctl`, refer to the `voldctl`(8) reference page.

8.1.3 Using voledit to Remove and Modify LSM Objects

The `voledit` utility has two functions:

1. Allows the system administrator to modify certain records in the LSM configuration databases. Only fields that are not volume usage-type-dependent can be modified.

2. Removes or renames LSM objects.

In general, LSM objects that are associated are not removable. This means that:

- A subdisk that is associated with a plex cannot be removed.
- A plex that is attached to a volume cannot be removed.

Note

Using the recursive suboption (`-r`) to the removal option of the `voledit` command removes all objects from the specified object downward. In this way, a plex and its associated subdisks, or a volume and its attached plexes and their associated subdisks, can be removed by a single invocation of this command.

Because the `-r` option of `voledit` removes multiple objects, exercise caution when using it.

For detailed information about how to use `voledit`, refer to the `voledit`(8) reference page.

8.1.4 Using volmake to Create LSM Objects

The `volmake` utility is used to add a new volume, plex, or subdisk to the set of objects managed by LSM. The `volmake` command adds a new record for that object to the LSM database. Records can be created entirely from parameters specified on the command line, or they can be created using a description file.

If operands are specified, then the first operand is a keyword that determines the kind of object to be created, the second operand is the name given to that object, and additional operands specify attributes for the object. If no operands are specified on the command line, then a description file is used to specify what records to create.

A *description file* is a file that contains plain text describing the objects to be created with `volmake`. A description file can contain several commands, and can be edited to perform a list of `volmake` utility operations. You can create a description file of existing LSM objects with the `volprint` utility, and then later recreate these LSM objects using the `volmake` utility. The following example shows a sample description file:

```
#rectyp #name    #options
sd       3s1-01  dmname=disk03 offset=0 len=20480
sd       4s1-01  dmname=disk04
plex     db-dsk  layout=STRIPE st_width=16k sd=3s1-01,4s1-01
sd       mem1-01 dmname=memdisk01 len=640h
                 comment="Hot spot for dbvol"
plex     db-mem  sd=mem1-01:40320

vol      db      use_type=gen plex=db-dsk,db-mem
                 read_pol=PREFER pref_name=memdbplx
                 comment="Uses mem1 for hot spot in last 5m"
```

By default, this description file is read from standard input. However, by using the `-d` option, a filename can be specified. For detailed information about how to use `volmake`, as well as detailed descriptions and definitions of the object-specific fields specified with `volmake`, refer to the `volmake`(8) reference page.

8.1.5 Using the volmend Utility

The `volmend` utility performs miscellaneous LSM usage-type-specific operations on volumes, plexes, and subdisks. These operations are used to:

- Clear utility fields
- Change the state of a volume or plex
- Take a volume or plex off line
- Place a volume or plex on line
- Perform specialized actions for objects with a particular usage type

Use the `volmend` utility primarily to escape from a state that was accidentally reached. The offline and online functions are also available with disk-related commands.

For detailed information about how to use `volmend`, refer to the `volmend`(8) reference page.

8.1.6 Using volplex to Perform Plex Operations

The `volplex` utility performs LSM operations on plex or on volume-and-plex combinations. The first operand is a keyword that determines the specific operation to perform. The remaining operands specify the configuration objects to which the operation is to be applied. The `volplex` utility can attach a plex to a volume and detach a plex from a volume. A detached plex does not participate in I/O activity to the volume, but remains associated with the volume. The semantics of a detached plex are somewhat different dependent upon usage type, but in the case of the standard usage types, `fsgen` and `gen`, a detached plex is reattached when a volume is next started.

Another capability provided by `volplex` is disassociation of a plex from the volume with which it is associated. When a plex is disassociated, its relationship to the volume is completely broken. At that point, the plex is available for other uses and can be associated with a different volume. This functionality is useful as part of a backup procedure.

Additionally, `volplex` provides options to copy the contents of the specified volume onto all of the named plexes, moving the contents of one plex onto a new plex, and allows for other usage-type-dependent operations to be added.

For detailed information about how to use `volplex`, refer to the `volplex`(8) reference page.

8.1.7 Using volprint to Print Configuration Information

The `volprint` utility provides a flexible method of displaying information from records in the LSM configuration database. This command can be used to display partial or complete information about any or all objects. The format can be hierarchical to clarify relationships or `volprint` can tailor the output for use by `volmake` or UNIX system utilities such as `awk`, `sed`, or `grep`.

For detailed information about how to use `volprint`, refer to the `volprint`(8) reference page.

8.1.8 Using volsd to Perform Subdisk Operations

The `volsd` utility is used to maintain subdisk-plex associations. The `volsd` utility can associate a subdisk with a plex or disassociate a subdisk from its associated plex, to move the contents of a subdisk to another subdisk, to split one subdisk into two subdisks that occupy the same space as the original, or to join two contiguous subdisks into one.

Note

Users should be aware that some `volsd` operations can take a considerable amount of time to complete.

For detailed information about how to use `volsd`, refer to the `volsd`(8) reference page.

8.1.9 Using volstat to Print Volume Statistics

The `volstat` utility prints statistics information about LSM objects and block devices under LSM control. The `volstat` utility reads the summary statistics from the volume device files in the directory `/dev/rvol` and formats them to the standard output. These statistics represent LSM activity from the time the system was initially booted. If no LSM object name is specified, then statistics from all volumes in the configuration database are reported.

For detailed information about how to use `volstat`, refer to the `volstat`(8) reference page.

8.1.10 Using voltrace to Trace Volume Operations

The `voltrace` utility prints formatted event log records and sets event trace masks. A *trace mask* is used to determine which type of events will be tracked, such as I/O events, configuration changes, or I/O errors. These events are then recorded by the volume driver in a *trace file*. The `voltrace` utility reads the volume event-log device (`/dev/volevent`) and writes formatted log entries to standard output. The `voltrace` utility prints log entries from all volumes in the database unless a specific volume to trace is named.

The `voltrace` utility can also be used to set trace masks as described above for a specific volume. The system default trace mask can also be changed, and will be used as the trace mask for any volumes for which a trace mask has not been set.

For detailed information about how to use `voltrace`, refer to the `voltrace`(8) reference page.

8.1.11 Using the volume Utility

The volume operations that the `volume` utility performs are:

- Initializing a volume
- Starting a volume

- Stopping a volume
- Establishing the read policy for a volume

Starting a volume changes its kernel state from DISABLED or DETACHED to ENABLED; stopping a volume changes its state from ENABLED or DETACHED to DISABLED. One of three read policies can be selected:

1. Round — Prescribes round-robin reads of enabled plexes.
2. Prefer — Prescribes preferential reads from a specified plex.
3. Select — Selects a default policy based on plex association to the volume.

For detailed information about how to use `volume`, refer to the `volume` reference page.

8.2 Subdisk Operations

LSM volumes are composed of two types of objects, plexes, and subdisks. A plex is composed of a series of subdisks linked together in an address space. A subdisk is a portion of a physical disk and is defined by disk media, offset, and length. Subdisks are low-level building blocks of an LSM configuration. The following sections describe each of the operations that can be performed in relation to subdisks. These subdisk operations are:

- Creating a subdisk
- Removing a subdisk
- Displaying a subdisk
- Associating a subdisk
- Disassociating a subdisk
- Changing a subdisk
- Moving a subdisk
- Splitting a subdisk
- Joining a subdisk

8.2.1 Creating Subdisks

The command to create LSM objects is `volmake`. The steps to create a subdisk include specifying the following:

1. Name of the subdisk
2. Length of the subdisk

3. Starting point (offset) of the subdisk within the disk
4. Disk media name

To create a subdisk, use either of the following commands:

volmake sd *name disk, offset, len*

or

volmake sd *name* **len=***length* **offset=***disk_offset* **disk=***lsm_disk*

For example, use either of the following command lines to create a subdisk labeled `disk02-01`, that starts at the beginning of disk disk02, and has a length of 8000 blocks:

volmake sd disk02-01 disk02,0,8000

or

volmake sd disk02-01 len=8000 offset=0 disk=disk02

Note

Commands take sizes in blocks. Adding a suffix changes the unit of measure. A `k` suffix specifies 1024-byte blocks. To preserve (encapsulate) data that exists on the disk, a plex and volume must be created to cover that data.

8.2.2 Removing Subdisks

You can remove subdisks when they are no longer in use or needed. To remove a subdisk, use the following command:

voledit rm *subdisk_name*

For example, the command line to remove a subdisk labeled `disk02-01` looks like this:

voledit rm disk02-01

8.2.3 Displaying Subdisks

The `volprint` utility displays information about LSM objects. To display general information for all subdisks, use the following command:

volprint –st

The `-s` option instructs `volprint` to get information about subdisks. The `-t` option prints a single-line output record that depends on the type of object being listed.

To display complete information about a particular subdisk, use the following command:

volprint –l ***subdisk_name***

For example, the command line to obtain all database information on a subdisk labeled `disk02-01` looks like this:

volprint –l disk02-01

8.2.4 Associating Subdisks

Associating a subdisk with a plex places the amount of disk space defined by the subdisk at a specific offset within the plex. In all cases, the entire area that the subdisk fills must not be occupied by any portion of another subdisk. There are several different ways that subdisks can be associated with plexes, depending on the overall state of the configuration.

If the system administrator has already created all the subdisks needed for a particular plex, subdisks are associated at plex creation by using a command similar to the following:

volmake plex home-1 sd=disk02-01,disk03-00,disk03-01

This command creates a plex `home-1` and associates subdisks disk02-01, disk03-00, and disk03-01 with the plex `home-1` during the plex creation process. Subdisks are associated in order starting at offset 0. Using a command like this one eliminates the need to specify the multiple commands necessary to create the plex and then associate each of the subdisks with that plex. In the previous example, the subdisks are associated to the plex in the order they are listed (after the `sd=`); the disk space defined as `disk02-01` will be first, the disk space of `disk03-00` is second, and `disk03-01` is third.

This method of associating subdisks is convenient during initial configuration. Subdisks can also be associated with a plex that already exists. One or more subdisks can be associated with an existing plex as follows:

volsd assoc *plex_name sd_name [sd_name2 sd_name3 ...]*

For example, the command line to associate subdisks labeled `disk02-01`, `disk03-00`, and `disk03-01` with a plex labeled `home-1` looks like this:

volsd assoc home-1 disk02-01 disk03-00 disk03-01

If the plex is not empty, the new subdisks are added after any subdisks that are already associated with the plex, unless the `-l` option is specified with the `volsd` command. The `-l` option provides a way to associate subdisks at a specific offset within the plex.

The `-l` option is needed in a case where a system administrator has created a sparse plex for a particular volume, and wants to make this plex complete. To make the plex complete, it is necessary to create a subdisk of exactly the size needed to fill the hole in the sparse plex, and then associate the subdisk with the plex by specifying the offset of the beginning of the hole in the plex. Use the following command to accomplish this task:

volsd -l *offset* **assoc** *sparse_plex_name exact_size_subdisk*

Note

The subdisk must be exactly the right size because LSM does not allow for the space defined by two subdisks to overlap within a single plex.

8.2.5 Associating Logging Subdisks

Logging subdisks are one-block long subdisks that are defined for and added to a plex that is to become part of a volume using block-change logging. Block-change logging is enabled for a volume when the volume has at least two active plexes that include a logging subdisk. Logging subdisks are ignored as far as the usual plex policies are concerned, and are only used to hold the *block change log* (described in Section 9.1.4).

Because block-change logs are one block in length, only a limited number of recent changes can be kept in the log. Older entries are removed from the log as newer I/O changes take place and are added to the log. Since all I/O must follow a log write, if no unused or completed log entries are available for use in the log block, then further I/O is delayed until some of the pending I/O completes and releases log space. This delay could adversely affect system performance.

Note

Only one logging subdisk can be associated with a plex. Because this subdisk is frequently written, care should be taken to position it on a disk that is not heavily used. Placing a logging subdisk on a heavily-used disk can result in severe degradation of system performance.

To add a logging subdisk to a plex, use the following command:

volsd aslog *plex subdisk*

For example, the command line to associate a subdisk labeled `disk02-01` with a plex labeled `vol01-02` (which is already associated with volume `vol01`) looks like this:

volsd aslog vol01-02 disk02-01

8.2.6 Disassociating Subdisks

To break an established relationship between a subdisk and the plex to which it belongs, the subdisk is *disassociated* from the plex. A subdisk is disassociated when the subdisk is to be removed or used in another plex. To disassociate a subdisk, use the following command:

volsd dis *subdisk_name*

To disassociate a subdisk labeled `disk02-01` from the plex with which it is currently associated, use the following command:

volsd dis disk02-01

Subdisks can also be removed with the command:

volsd -orm dis *subdisk_name*

8.2.7 Changing Subdisk Information

The `voledit` utility changes information related to subdisks. To change information relating to a subdisk use the following command:

voledit set *field=value ... subdisk_name ...*

For example, the command line to change the comment field of subdisk labeled `disk02-01` looks like this:

voledit set comment= *"New comment"* **disk02-01**

The subdisk fields that can be changed using `voledit` are:

- The `putil` *[n]* fields
- The `tutil` *[n]* fields
- The `len` field (only if the subdisk is disassociated)

Table 8-1 describes these fields.

Table 8-1: The `putil[n]` and `tutil[n]` Fields

Field	Description
`putil0`	This utility field is reserved for use by the LSM utilities and is retained on reboot.
`putil1`	This utility field is reserved for use by high-level utilities such as Visual Administrator interface (dxlsm) and the LSM Support Operations interface (voldiskadm). This field is retained on reboot.
`putil2`	This utility field is reserved for use by the system administrator or site-specific applications. This field is retained on reboot.
`tutil0`	This utility field is reserved for use by the LSM utilities and is cleared on reboot.
`tutil1`	This utility field is reserved for use by high-level utilities such as dxlsm and voldiskadm. This field is cleared on reboot.
`tutil2`	This utility field is reserved for use by the system administrator or site-specific applications. This field is cleared on reboot.

Note

Entering data in the `putil0` field prevents the subdisk from being used as part of a plex, if it is not already.

8.2.8 Moving Subdisks

Moving a subdisk copies the disk space contents of a subdisk onto another subdisk. If the subdisk being moved is associated with a plex, then the data stored on the original subdisk is copied to the new subdisk, the old subdisk is disassociated from the plex, and the new subdisk is associated with the plex, at the same offset within the plex as the source subdisk. To move a subdisk, use the following command:

volsd mv *old_subdisk_name new_subdisk_name*

For the subdisk move operation to perform correctly, the following conditions must be met:

- The subdisks involved must be the same size.
- The subdisk being moved must be part of an active plex on an active (ENABLED) volume.
- The new subdisk must not be associated with any other plex.

8.2.9 Splitting Subdisks

Splitting a subdisk divides an existing subdisk into two subdisks. The `-s` option is required to specify the size of the first of the two subdisks that will be created. To split a subdisk, use the following command:

volsd –s *size* **split** *sd newsd newsd2*

In the above example, *sd* is the name of the original subdisk, *newsd* is the name of the first of the two subdisks that will be created, and *newsd2* is the name of the second subdisk to be created.

If the existing subdisk is associated with a plex before the `split` operation, upon completion of the split, both of the resulting subdisks will be associated to the same plex.

8.2.10 Joining Subdisks

Joining a subdisk combines two or more existing subdisks into one subdisk. To join subdisks, the subdisks must be contiguous on the same disk; if the selected subdisks are associated, they must be associated with the same plex, and be contiguous in that plex. The command to join a subdisk is:

volsd join *subdisk1 subdisk2 new_subdisk*

8.3 Plex Operations

Plexes are logical groupings of subdisks that create an area of disk space independent of any physical disk size. Replication of disk data can be accomplished by defining multiple plexes that will be attached to one volume. The replication provided by multiple plexes prevents data loss in the event of a single-point disk-subsystem failure. Multiple plexes also provide increased data integrity and reliability.

Plex operations include:

- Creating a plex

- Removing a plex
- Associating a plex
- Disassociating a plex
- Listing all plexes
- Displaying plexes
- Changing plex attributes
- Changing plex status
- Moving plexes
- Copying plexes

8.3.1 Creating Plexes

Plexes are created by identifying subdisks and associating them to the plex that you want to create. The `volmake` command creates LSM objects. To create a plex from existing subdisks, use the following command:

volmake plex *plex_name* **sd=***subdisk_name,...*

For example, the command line to create a plex labeled `vol01-02` using two existing subdisks labeled `disk02-01` and `disk02-02` looks like this:

volmake plex vol01-02 sd=disk02-01,disk02-02

8.3.2 Backup Using a Plex

If a volume is mirrored, backup can be performed on that volume by taking one of the volume's plexes offline 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 for the duration of the time needed for the backup.

Perform a backup of a mirrored volume on an active system as follows:

1. Stop I/O activity and flush any buffers to improve the consistency of the backup. For example, Digital 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 operation should only take a few seconds. It will leave the device `/dev/plex/vol-01` available as an image of the volume frozen at the time of the disassociation.

3. At this point, resume I/O activity. For example, if the volume contained UFS, it can be remounted.
4. Create a temporary value. For example:

   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. For example:

   dump 0 /dev/rvol/vtmp

7. Reattach the plex to the volume in order to regain redundancy of the volume. For example:

   volplex dis vol-01

   volplex att *volume-name* **vol-01**

8.3.3 Associating Plexes

A plex becomes a participating plex for a volume by associating the plex with the volume. To associate a plex with an existing volume, use the following command:

volplex att *volume_name plex_name*

For example, the command line to associate a plex labeled `vol01-02` with volume labeled `vol01` looks like this:

volplex att vol01 vol01-02

Alternately, if the volume has not been created, a plex (or multiple plexes) can be associated with the volume to be created as part of the volume create command:

volmake -U *usetype* **vol** *vol_name* **plex=***plex_name1, plex_name2***...**

For example, the command line to create a mirrored, `fsgen`-type volume labeled `home`, and associate two existing plexes labeled `home-1` and `home-2` to the volume `home` looks like this:

volmake -Ufsgen vol home plex=home-1,home-2

8.3.4 Removing Plexes

When a plex is no longer needed, it can be removed. Examples of operations that require plexes to be removed are:

- Reducing the number of plexes in a volume to increase the length of another plex and its associated volume; the plex and subdisks are removed, then the resulting space can be added to other volumes
- Removing a temporary plex that was created to backup a volume and is no longer required
- Changing the layout of a plex from concatenated to striped, or vice versa

Caution

To save the data on a plex that is to be removed, 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 such a plex is removed, its configuration should be recorded.

A plex can be disassociated from a volume and removed with the following command:

volplex -o rm dis *plex_name*

To disassociate and remove a plex labeled `vol01-02` use the following command:

volplex -o rm dis vol01-02

This removes the plex `vol01-02` and all associated subdisks.

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, use the following command:

voledit -r rm *plex_name*

Because the `-r` option of `voledit` removes multiple objects. Exercise caution when using it.

8.3.5 Listing All Plexes

Listing plexes helps identify free plexes that can be used for building volumes. Using the `volprint` utility with the plex (`-p`) option lists information about all plexes; the `-t` option prints a single line of information about the plex. To list free plexes, use the following command:

volprint -pt

8.3.6 Displaying Plexes

To display detailed information about all plexes, use the following command:

volprint -lp

To display detailed information about a specific plex, use the following command:

volprint -l *plex_name*

8.3.7 Changing Plex Attributes

The `comment` field and the `putil` and `tutil` fields are used by the utilities after plex creation. The `putil` fields attributes are maintained on reboot; the `tutil` fields are temporary and are not retained on reboot. Both `putil` and `tutil` have three uses and are numbered according to those uses. These fields can be modified as needed. LSM uses the utility fields marked `putil0` and `tutil0`; other products use those marked `putil1` and `tutil1`; and those marked `putil2` and `tutil2` are user fields. Table 8-1 in Section 8.2.7 describes the uses for the `putil` and `tutil` fields.

To change plex attributes, use the following command:

voledit set *field=value* **...** *plex_name* **...**

The following steps demonstrate how the `voledit` command can be used to modify comment fields:

1. Set the `comment` field that identifies what the plex (`vol01-02`) is used for, to `my_plex`.
2. Enter `u` in `tutil2` to indicate that the subdisk is in use.
3. Use the following command to change the user ID to `admin` in the `putil2` field:

voledit set comment="my plex" tutil2="u" putil2="admin" vol01-02

To prevent a particular plex from being associated with a volume, set the `putil0` field to a non-null string as specified in the following command:

voledit set putil0="DO-NOT-USE" vol01-02

8.3.8 Changing Plex Status: Detaching and Attaching Plexes

Once a volume has been created and placed online (ENABLED), LSM provides mechanisms by which plexes can be temporarily disconnected from the volume. This is useful, for example, when the hardware on which a plex resides needs repair or when a volume has been left unstartable and a source plex for the volume revive must be chosen manually.

Resolving a disk or system failure includes taking a volume off line and attaching and detaching its plexes. The two commands used to accomplish disk failure resolution are `volmend` and `volplex`.

To take a plex OFFLINE so that repair or maintenance can be performed on the physical disk containing that plexes' subdisks, use the following command:

volmend off *plex_name* **...**

If a disk drive suffered a head crash, the system administrator should put all plexes that have associated subdisks represented on the affected drive OFFLINE. For example, if plexes `vol01-02` and `vol02p1` had subdisks on a drive to be repaired, use the following command:

volmend off vol01-02 vol02p1

This command places `vol01-02` and `vol02-01` in the OFFLINE state, and they remain in that state until explicitly changed.

8.3.8.1 Detaching Plexes

To temporarily detach one plex in a mirrored volume, use the following command:

volplex det *plex_name* **...**

For example, the command line to temporarily detach a plex labeled `vol01-02` and place it in maintenance mode looks like this:

volplex det vol01-02

This command temporarily detaches the plex, but maintains the association between the plex and its volume; however, the plex will not be used for I/O. A plex detached with the preceding command will be recovered on a system reboot. The plex state is set to STALE, so that if a `volume start` command is run on the appropriate volume (for example, on system reboot), the plex will be revived and made ACTIVE.

When the plex is ready to return as an active part of its volume, follow this procedure:

1. If the volume is not ENABLED, start it by using

   volume start vol_name.

   If it is unstartable, set one of the plexes to CLEAN using

   volmend plex clean *plex_name,*

   and then start the volume.

2. If the plex does not yet have a *kernel state* of ENABLED, issue the following command:

   volplex att *volume_name plex_name* **...**

   As with returning an OFFLINE plex to ACTIVE, this command starts a revive of the plexes stated, and when each revive completes, sets the plex state to ACTIVE.

8.3.8.2 Attaching Plexes

When the disk has been repaired or replaced and is again ready for use, the plexes must be put back online (plex state set to ACTIVE). The following steps describe putting plexes back on line.

1. If the volume is currently ENABLED, use the following command:

   volplex att *volume_name plex_name* **...**

   For example, the command line for a plex labeled `vol01-02` on a volume labeled `vol01` looks like this:

   volplex att vol01 vol01-02

   This starts a revive of the plex `vol01-02` and, after the revive is complete, sets the plex utility state to ACTIVE.

2. If the volume is not in use (not ENABLED), use the following command:

   volmend on *plex_name*

   For example, the command line for a plex labeled `vol01-02` looks like this:

volmend on vol01-02

In this case, the state of `vol01-02` is set to STALE, so that when the volume is next started, the data on the plex will be 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, refer to Section 8.4.10. See the `volmake`(8) and `volmend`(8) reference pages for more information about these commands.

8.3.9 Moving Plexes

Moving a plex copies the data content from the original plex onto a new plex. In order for a move operation to be successful, the following criteria must be met:

- The old plex must be an active part of an active (ENABLED) volume.
- The new plex should be at least the same size or larger than the old plex.
- The new plex must not be associated with another volume.

The size of the plex has several important implications. If the new plex is smaller, or more sparse, than the original plex, an incomplete copy of the data on the original plex results. If this is the desired action, then the `-o force` option is required. If the new plex is longer, or less sparse, than the original plex, then the data that exists on the original plex will be copied onto the new plex and then any area that was not on the original plex, but is represented on the new plex, will be filled from other complete plexes associated with the same volume. If the new plex is longer than the volume itself, then the remaining area of the new plex above the size of the volume will not be initialized.

The command to move data from one plex to another is:

volplex mv *original_plex new_plex*

8.3.10 Copying Plexes

This operation copies the contents of a volume onto a specified plex. The volume to be copied must not be enabled. The plex must not be associated with any other volume. To copy a plex, the following command is used:

volplex cp *vol_name new_plex*

After the copy operation is complete, *`new_plex`* will not be associated with the specified volume *`vol_name`*. The plex contains a complete copy of the volume data. The plex that is being copied should be the same size or larger than the volume, otherwise an incomplete copy of the data results. For this same reason, `new_plex` also should not be sparse.

8.4 Volume Operations

A *volume* is a collection of from one to eight plexes and appears as a block and character device in the `/dev/vol` directory. A volume can be used as a partition device.

Volume operations include:

- Creating a volume
- Removing a volume
- Listing all volumes
- Displaying volumes
- Changing volume attributes
- Resizing a volume
- Changing volume read policy
- Starting and stopping volumes
- Listing unstartable volumes
- Mirroring an existing volume
- Displaying volume plexes
- Recovering a disabled volume
- Initializing a volume

The following sections describe how to perform common volume operations. In some cases, either `volassist` or one or more other commands can be used to accomplish the same task; in such cases, both approaches are described. Detailed descriptions about the commands used to perform volume operations are contained in the LSM reference pages.

8.4.1 Creating a Volume

Volumes can be created with either `volassist` or `volmake`.

The length of a new volume can be specified in sectors, megabytes, or kilobytes. The unit of measure is indicated by adding the appropriate suffix to the length (`s`, `m`, or `k`). If no unit is specified, sectors are assumed.

8.4.1.1 Creating Volumes with the volassist Command

The volassist command can be used to create volumes with default settings or with user-specified attributes. The volassist command automatically creates and attempts to enable new volumes. If the volume fails to be enabled, volassist attempts to remove it and releases the space used to allocate that volume.

To create a simple volume using volassist and its default settings, enter the following command:

volassist make *volume_name length*

For example, the following command creates a volume named *voldef* with a length of 10 megabytes on any available disks:

volassist make voldef 10m

Additional parameters can be specified to volassist to reflect the new volume's attributes. Refer to the volassist(8) reference page for details. The following example illustrates the creation of a volume named volzebra that is striped across disk03 and disk04, has the fsgen usage type, and is 10 megabytes long:

volassist -Ufsgen make volzebra 10m layout=stripe disk03 disk04

8.4.1.2 Creating Volumes with the volmake Command

To create a volume using volmake, use the following command:

volmake –U *usage_type* **vol** *volume_name* **len=***length* **plex=***plex_name,...*

Either the plex= parameter or the len= parameter can be omitted. If the len= parameter is specified with the plex= parameter, then the volume created from an existing plex takes the length of that plex. If the len= parameter is specified without the plex= parameter, then at least one plex of the appropriate length must later be specified.

Examples of commands for creating a fsgen-type volume called vol01 are as follows:

volmake –Ufsgen vol vol01 len=100000

or

volmake vol vol01 use_type=fsgen plex=vol01-01,vol01-02

Note that the first command shown above creates a volume that has no plex.

The usage type for a volume can be specified in either of two ways: `-Ufsgen` or `use_type=fsgen`. If a length is not specified (for example, `len=100000`) or associated plexes are not identified (for example, `plex=vol01-01,vol01-02`), the volume length will be zero.

Instead of specifying parameters on the command line, you can use a `volmake` description file to create a volume, as well as associated subdisks and plexes, by using the following command:

volmake –d *description_file*

For detailed information about how to use `volmake`, and an example of the `volmake` description files, refer to Section 8.1.4 or to the `volmake`(8) reference page.

8.4.2 Initializing Volumes

LSM automatically sets the states of plexes and volumes based on usage type. The `volume start` command performs a default volume initialization on a newly-created volume; the `volume init` command is used for spanned volumes. However, should manual initialization be necessary (for example, when a multiple-plex volume is first configured and needs to be started), use the following command:

volume init *how volume_name [plex_name]*

The *how* variable determines what the initialization does and what condition the volume and plexes have after the volume has been initialized. The most common form of manual initialization is setting the state of the volume and one of its plexes to CLEAN. To set the state of a volume and plex to CLEAN, use the following command:

volume init clean *volume1 plex*

The following commands illustrate a specific example of when the system administrator would want to set a volume and plex to the CLEAN state during manual initialization:

volmake -Ufsgen home plex=home-1,home-2

volume init clean home home-2

volume start home

In the preceding example, the system administrator has created a new two-plex volume, and then realized that the data that existed on plex `home-2` was the data that needed to be mirrored onto the plex labeled `home-1`. By setting the state of `home-2` to CLEAN, LSM will correctly copy the data onto (revive) `home-1` during the `volume start` operation.

To make all the plexes in a volume ACTIVE and enable the volume, use the following command:

volume init active *volume_name*

To enable a volume to temporarily allow I/O to dump the contents of a tape into a volume, use the following command:

volume init enable *volume_name*

This enables the volume and all its plexes, but leaves the plex utility state as EMPTY.

After the I/O is complete, issue the `volume init active` version of the command.

To zero out all the plexes, set them to ACTIVE, and enable the volume, use the following command:

volume init zero *volume_name*

Caution

For both mirrored and nonmirrored volumes, this action destroys any existing data residing on the volume. Do not use the `zero` option unless you are sure that no critical data exists on any plex in the volume. Most initializations will destroy data on some plexes. To preserve data on a volume, use either `volume init active` or `volume init clean`, as appropriate.

8.4.3 Removing Volumes

Removing a volume is the same as removing a physical disk partition on a non-volume-managed system. If the volume has been used, and critical data may still reside on the disk space defined by that volume, the system administrator should make a backup of the data, and take care to ensure that this data is retained.

To remove a volume, use the following command:

voledit rm *volume_name*

or

voledit –rf rm *volume_name*

The `-r` option indicates recursive removal, which means the removal of all plexes associated with the volume and all subdisks associated with those plexes. The `-f` option forces removal, and is necessary if the volume is enabled.

Warning

> The `-r` option of `voledit` removes multiple objects. Exercise caution when using it.

8.4.4 Displaying Volumes

It is possible to list information related to volumes under LSM control. This information includes the name of the volume, its usage type, state, length, user and group IDs, and mode. To list information on all volumes, use the following command:

volprint –vt

The utility used to display information about LSM objects is `volprint`. To display detailed information about a specific volume, use the following command:

volprint –l *volume_name*

If no volume is specified, detailed information is given for all volumes by using the following command:

volprint -vl

8.4.5 Changing Volume Attributes

Volume attributes such as read policy, error policies, ownership, permissions, and the values in the comment and utility fields for existing volumes can be changed. These attributes are changed whenever the use of the volume or users' needs change.

There are two LSM commands associated with setting volume attributes:

- The `voledit` command sets those attributes that are not usage-type-dependent.
- The `volume` command sets only those attributes that are usage-type-dependent.

Examples of how to use each of these commands follow:

volume set *field=value ... volume_name ...*

or

voledit set *field*= *value* ... *volume_name* ...

Table 8-2 details which attributes can be set by each command.

Table 8-2: Setting Volume Attributes

Command	Attribute	Description
voledit		The comment field
	tutil0, tutil1, tutil2	Descriptive string of volume contents
	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	(blkno/undef) specifier of block-change logging mode for volume
	log	Length of the block-change logging log
	start opts	Options to be executed to the volume start operation

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:

voledit set user=susan group=staff mode=0666 vol01

8.4.5.1 Resizing a Volume

Resizing a volume is an instance of changing volume attributes that can be handled via either volassist or volume. Striped volumes cannot be resized.

The new size of a volume can be specified in sectors, megabytes, or kilobytes. The unit of measure is indicated by adding the appropriate suffix to the length (s, m, or k). If no unit is specified, sectors are assumed.

8.4.5.1.1 Using the volassist Command: – `volassist` can resize a volume in any of the following ways:

`growto`	Increase volume to specified length
`growby`	Increase volume by specified amount
`shrinkto`	Reduce volume to specified length
`shrinkby`	Reduce volume by specified amount

If the volume is increased in size, `volassist` automatically seeks out available disk space.

To increase a volume to a specified length, use the command:

volassist growto *volume_name new_length*

To increase a volume by a certain amount, use the command:

voassist growby *volume_name length_change*

To reduce a volume to a specified length, use the command:

volassist shrinkto *volume_name new_length*

To reduce a volume by a certain amount, use the command:

volassist shrinkby *volume_name length_change*

8.4.5.1.2 Using the volume Command: – To change the length of a volume using `volume set`, use the following command:

volume set len=*value* **...** *volume_name* **...**

For example, to change the length to 100000 sectors, use the following command:

volume set len=100000 vol01

Note

Note the following when using the `volume` command:

- The `volume set len` command cannot increase the size of a volume unless the needed space is available in the plexes of the volume.
- When a volume's size is reduced using the `volume set len` command, the freed space is not released into the free space pool.

8.4.5.2 Changing 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 has been labeled as the preferred plex. The read policy can be changed 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. To set a read policy, use one of the following commands:

volume rdpol round *volume_name*

or

volume rdpol prefer *volume_name preferred_plex_name*

For example, the command line to set the read policy for volume `vol01` to a round-robin read looks like this:

volume rdpol round vol01

The command line to set the policy for the same volume to read preferentially from the plex `vol01-02` looks like this:

volume rdpol prefer vol01 vol01-02

8.4.6 Starting and Stopping Volumes

Like mounting and unmounting a file system, starting and stopping a volume affects its availability to the user. Starting a volume changes its state and makes it available for use. Stopping a volume makes it unavailable.

Starting a volume changes the volume state from DISABLED or DETACHED to ENABLED. The success of this operation depends on the ability to enable a volume. If a volume cannot be enabled, it remains in its current state. To start a volume, use the following command:

volrecover -s *volume_name* **...**

To start all DISABLED volumes, use the following command:

volrecover -s

Stopping a volume changes the volume state from ENABLED or DETACHED to DISABLED. If the command cannot stop it, the volume remains in its current state. To stop a volume, use the following command:

volume stop *volume_name* **...**

For example, the command line to stop a volume labeled `vol01` looks like this:

volume stop vol01

To stop all ENABLED volumes, use the following command:

volume stopall

If all plexes of the volume become STALE, put the volume in maintenance mode so that the plexes can be looked at while the volume is DETACHED and determine which plex to use for reviving the others. To place a volume in maintenance mode, use the following command:

volume maint *volume_name*

To assist in choosing the revival source plex, list the unstarted volume and displays its plexes.

To take plex `vol01-02` offline, use the following command:

volmend off vol01-02

For ENABLED volumes, save a step by using `volplex att` without first invoking `volmend on`. This command works on an OFFLINE plex of an ENABLED volume (designated as `vol01` in the example):

volplex att vol01 vol01-02

The `volmend` utility can change the state of an OFFLINE plex of a DISABLED volume to STALE, after which a `volume start` on the volume would revive the plex. To put a plex labeled `vol01-02` in the STALE state, use the following command:

volmend on vol01-02

To make other state changes in a plex or a volume, refer to the subsequent sections on volume recovery options.

8.4.7 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 [*volname*]

8.4.8 Mirroring Existing Volumes

A plex can be added to an existing volume. This can be done with the `volassist` command as follows:

volassist mirror *volume_name*

For example, the following command creates a plex of a volume `voltest`:

volassist mirror voltest

Another way to mirror an existing volume is by first creating a plex and then associating it to a volume, using the following commands:

volmake plex *plex_name sd=subdisk_name ...*

volplex att *volume_name plex_name*

8.4.9 Displaying Plexes Within a Volume

To limit the display of `volprint` to a single object, specify the object name after the command. To display the plexes for a volume labeled `vol01`, use the following command:

volprint -tv vol01

To display the volume, its plexes, and the subdisks in those plexes.

You can also display the plexes for `vol01` with the command:

volprint –e 'assoc="vol01"'

This displays only the plexes associated with `vol01`.

8.4.10 Volume Recovery

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, use the following command:

volmend fix clean *plex_name*

For example, the command line to place one plex labeled `vol01-02` in the CLEAN state looks like this:

volmend fix clean vol01-02

Refer to the `volmend`(8) reference pages for more information.

Advanced Volume Management Concepts 9

The two basic functions of the LSM software is to satisfy read and write requests, and ensure data is available. LSM automatically ensures that when you mirror disks, the corresponding plexes contain the same data. Under certain circumstances, LSM must perform a copy operation to ensure that corresponding plexes are mirrors of each other.

The state of LSM plexes and volumes can vary during the life of the LSM configuration. Changes in the composition of the volume are inevitable because:

- Disk drives occasionally need corrective maintenance
- New disks are added to replace other disks
- System failures occur, requiring copy operations to take place within the LSM volume

This chapter describes how LSM manages plexes and volumes, and provides information for administrators who want to understand LSM plex and volume states, usages, and policies.

9.1 Plex States

System administrators look at *plex states* to see whether or not plexes are complete and consistent copies of the volume contents. Although the LSM utilities automatically maintain a plex's state, it is possible for you to modify the state of a plex if necessary. For example, if a disk with a particular plex located on it begins to fail, you can temporarily disable the plex.

LSM utilities use plex states to:

- Indicate whether volume contents have been initialized to a known state
- Determine if a plex contains a valid copy of the volume contents
- Track whether a plex was in active use at the time of a system failure
- Monitor operations on plexes

Plex states are an important aspect of high data availability. The following subsections describe all of the different plex states, how they change to indicate abnormalities, and what LSM does to normalize the plex state again.

Plexes that are associated with a volume will be in one of the states shown in Table 9-1.

Table 9-1: LSM Plex States

Plex State	Description
ACTIVE	A plex can be in the ACTIVE state in two situations: 1) When the volume is started and the plex fully participates in normal volume I/O (meaning that the plex contents change as the contents of the volume change) and, 2) When the volume was stopped as a result of a system crash and the plex was ACTIVE at the moment of the crash. Because of the impossibility of making atomic changes to more than one plex, a system failure may leave plex contents in an inconsistent state. When a volume is started, LSM performs a recovery action to guarantee that the contents of the plexes that are marked as ACTIVE are made identical. **Note** On a system that is performing well, you should see most volume plexes in the ACTIVE plex state.
CLEAN	A plex is in a CLEAN state when it contains a consistent copy of the volume contents and a `volume stop` operation has disabled the volume. As a result, when all plexes of a volume are CLEAN, no action is required to guarantee that the plexes are identical when that volume is started.
EMPTY	Volume creation sets all plexes associated with the volume to the EMPTY state to indicate to the usage type utilities (volume usage types are discussed later in this chapter) that the volume contents have not yet been initialized.
IOFAIL	On the detection of a failure of an ACTIVE plex, `vold` places that plex in the IOFAIL state so that it is disqualified from the recovery selection process at volume start time.
OFFLINE	The `volmend off` operation indefinitely detaches a plex from a volume by setting the plex state to OFFLINE. Although the detached plex maintains its association with the volume, changes to the volume do not update the OFFLINE plex until the plex is reattached with the `volplex att` operation. When this occurs, the plex is placed in the STALE state, which causes its contents to be recovered.

Table 9-1: (continued)

Plex State	Description
STALE	If there is a possibility that a plex does not have the complete and current volume contents, that plex is placed in the STALE state. Also, if an I/O error occurs on a plex, the kernel stops using and updating the contents of that plex, and a `volume stop` operation sets the state of the plex to STALE. A `volume start` operation revives STALE plexes from an ACTIVE plex. Atomic copy operations copy the contents of the volume to the STALE plexes. The system administrator can force a plex to the STALE state with a `volplex det` operation.
TEMP	Setting a plex to the TEMP state facilitates some plex operations that cannot occur in a truly atomic fashion. For example, attaching a plex to an enabled volume requires copying volume contents to the plex before it can be considered fully attached. A utility will set the plex state to TEMP at the start of such an operation and to a final state at the end of the operation. If the system goes down for any reason, a TEMP plex state indicates that the operation is incomplete; a `volume start` will disassociate plexes in the TEMP state.
TEMPRM	A TEMPRM plex state resembles a TEMP state except that at the completion of the operation, TEMPRM plex is removed. Some subdisk operations require a temporary plex. Associating a subdisk with a plex, for example, requires updating the subdisk with the volume contents before actually associating the subdisk. This update requires associating the subdisk with a temporary plex, marked TEMPRM, until the operation completes and removes the TEMPRM plex. If the system goes down for any reason, the TEMPRM state indicates that the operation did not complete successfully. A subsequent `volume start` operation will disassociate and remove TEMPRM plexes.

9.1.1 Plex State Cycle

Plex states change as a normal part of disk operations. However, deviations in plex state indicate abnormalities that LSM must normalize. Table 9-2 describes possible failure scenarios and the actions LSM takes to fix deviations.

Table 9-2: How LSM Handles Changes in Plex States

Cycle	During Normal Operations	When a Crash Occurs
Startup	The `volume start` operation makes all CLEAN plexes ACTIVE. The plexes remain marked as ACTIVE unless a crash occurs.	If a crash occurs between startup and shutdown, the volume-starting operation does not find any CLEAN plexes, only ACTIVE plexes. The operation then establishes one plex as an up to date and suitable source for reviving the other plexes. LSM marks that source plex ACTIVE and marks all others as STALE. The volume-usage type determines which plex is selected as the source plex.[a] [b]
Shutdown	If all goes well until shutdown, the volume-stopping operation marks all ACTIVE plexes CLEAN and the cycle continues. Having all plexes CLEAN at startup (before `volume start` makes them ACTIVE) indicates a normal shutdown and optimizes startup.	If an I/O error occurred and caused a plex to become disabled, the volume-stopping operation marks the plex in which the error occurred as STALE. Any STALE plexes require recovery. When the system restarts, a utility copies data from an ACTIVE to a STALE plex and makes the STALE plex ACTIVE.

Table Note:

a. Any plex can serve as the source for generic (`gen`) usage type volumes. The most up-to-date plex is selected for file system generic (`fsgen`) volumes. If the startup operation finds neither a CLEAN nor an ACTIVE plex, the system administrator must use `volmend` to select a plex to be set to CLEAN.

b. If the volume has the writeback-on-read flag set, all ACTIVE plexes are attached to the volume as ACTIVE plexes. A process is forked which reads the entire volume and the read data is written to the remaining plexes. Refer to Section 9.1.6 for more details.

9.1.2 Plex Kernel State

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), and online (ENABLED) mode of operation. These modes are described in the following table.

Mode	Description
DISABLED	The plex may not be 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.
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.

Note

User intervention is not required to set these states.

9.1.3 Plex Layout Policy

Plexes are logical groupings of subdisks that create an area of disk space that is independent of any physical disk size. The subdisks that make up a plex can be filled with data in two ways, *concatenation* or *striping*:

- *Concatenation* places sequentially written data on subdisks in the order that the data was created. The first subdisk is filled, then the second, and so on.
- *Striping* alternates sections of plex data among multiple disks. To accomplish this, the subdisks forming a plex must all be of equal size and are divided into stripe-blocks of equal size (the stripe width). Data is placed on the subdisks by stripe; sequentially written data fills stripe block 0 on Subdisk 1 first, then stripe block 1 on Subdisk 2, and so on.

  For striping to be effective, subdisks must be spread across multiple disks, with one stripe per disk. Striping provides a performance advantage over concatenation because striping allows parallel I/O activity. You can use striping to distribute hot spots or areas of high I/O traffic across multiple devices.

See also Chapter 2 for more information about concatenation and striping.

9.1.4 Block-Change Logging

Block-change logging is a method used to dramatically reduce synchronization overhead of a mirrored volume during recovery in case of a system failure. Block-change logging keeps a log of the blocks that have changed due to I/O writes to a plex. If block-change logging is not enabled and a system failure occurs, LSM must restore all plexes to a consistent state by copying the full contents of an ACTIVE plex to the STALE plexes. This process can be lengthy and I/O intensive.

Block-change logging tracks writes by identifying and logging the block number that has changed, and stores this number in a *logging subdisk*. The block-change log maintains a record of all pending I/O records. Because the block-change log is only one block long, and because the log stores the I/O information until a process completes, the log may not be able to store all I/O changes as they occur. Once a process completes, it is flushed from the log and pending I/O information is then placed in the log.

Log records are written before the data is written. Thus, if the system experiences a crash, on system restart LSM searches the log and uses the log IDs to determine which plex contains the latest data written before the crash. In this way, plexes remain consistent, and except for possibly the last write before the crash, data is intact and up to date.

Block-change logging is enabled if two or more plexes in a mirrored volume have a logging subdisk associated with them. In addition, only the blocks recorded in the log of the ACTIVE plex need to be copied to restore the STALE plexes and maintain data integrity. For example:

volmake sd rz1-01 len=1 rz1

volsd aslog vol01-01 rz1-01

Note

Block-change logging can add some overhead to your system, because LSM must perform an extra I/O for every write operation.

9.1.5 Persistent State Logging

Persistent state logging ensures that only active plexes are used for recovery purposes and prevents failed plexes from being selected for recovery. Table 9-3 describes how persistent state logging solves problems due to plex failures and recovery.

Table 9-3: Recovering from Plex Failures

Problem	Resolution
When LSM error policies sometimes detach failing plexes, the state information about the failed plex is maintained on disk and is kept by the kernel as a dynamic record of the state of the configuration. However, in the event of a system failure the kernel state of the plex at the time of the failure is unknown. Thus, during recovery, LSM might select a plex that has been detached from the volume (due to an error) some time before the system failed, and the data contained in the selected failed plex could be significantly out of date.	When LSM detaches a failed plex, it immediately writes a record to the persistent state log. This way, even if a system failure occurs between the time that the plex is detached and the state change is logged, the detached plex is then disqualified from the recovery selection process. Persistent state logging can therefore guarantee that only active plexes are selected for recovery purposes.[a]
Without persistent state logging, a system crash causes all plexes to go through a recovery process, regardless of whether or not the plexes had been accessed.	Persistent state logging prevents unnecessary plex recoveries associated with started volumes that have never been accessed. Persistent state logging maintains a record of the first write to a volume and also of the last close of the volume so that no plex recovery is attempted following a crash.[b]

Table Notes:

a. A special plex state, IOFAIL, exists for failed plexes. As soon as the failure of an active plex is detected, `vold` places that plex in the IOFAIL state to ensure that it is disqualified from the selection process for any subsequent volume start operation.

b. With persistent state logging, a transaction completion record is logged whenever a transaction completes. In this way, a later recovery will be able to determine the state of pending transactions and perform the appropriate recovery action.

9.1.6 Plex Resynchronizing Policy

During a `volume start` operation for an LSM volume, it might be necessary to resynchronize plexes that have become out of date using the following steps:

1. A set of the volume's plexes are chosen as being up to date or most up to date and the volume is made available with the up-to-date plexes in the ACTIVE state.
2. The remaining plexes are then set to the STALE state and are made unavailable for reading. These plexes are brought up to date with the ACTIVE plexes through a VOL_COPY process that copies the contents of the ACTIVE plexes to the out-of-date plexes and changes their state to ACTIVE.
3. The volume is then available with all plexes active.

During the resynchronization process, however, it is possible that only one plex is active for a volume. For the duration of the resynchronization, the volume is therefore vulnerable to I/O failures because there is no reliable redundancy of the contents of the volume. This might leave the volume in an irrecoverable state if an error is encountered.

The writeback-on-read mode avoids such problems. When a read is received for a volume, the `writeback-on-read` flag causes the data read to be written back to all other plexes in the volume. The `volume start` command uses the writeback-on-read mode to start volumes with only ACTIVE plexes from which to recover.

The writeback-on-read model works as follows:

1. All plexes that are marked as ACTIVE are attached to the volume as ACTIVE upon volume startup, and the `writeback-on-read` flag is set for the volume.
2. The start operation then forks off a process which generates read I/O for the entire volume. The reads are serviced normally (for example, a plex from among the ACTIVE plexes is chosen via the volume read policy) and the read data is then written back to the remaining plexes.
3. When the read loop has finished, the `writeback-on-read` flag is unset and all the plexes of the volume are now consistent.

The success of writeback-on-read depends on the existence of Persistent State Logging, which guarantees that plexes marked as ACTIVE and not marked as DETACHED in the persistent state log area were all active at the time of the system crash. If this is the case, then the only areas of the ACTIVE plexes where data may disagree are those places that had active write I/O at the time of the crash.

Data in these areas is not guaranteed to be correct in terms of representing the data either before or after the write, but LSM guarantees that the data areas are consistent across all plexes. Writeback-on-read supports this consistency.

In addition, the failure of any plex during the synchronization process results in the normal error processing being performed without entering an irrecoverable state, because other plexes are available for use.

9.2 Volume States

Volume states indicate whether or not the volume is initialized, written to, and the accessibility of the volume.

The interpretation of these volume states during volume startup is modified by the persistent state log for the volume (for example, the dirty/clean state). If the flag for the clean state is set, this means that an ACTIVE volume was not written to by any processes or was not open at the time of the reboot; therefore, it can be considered CLEAN. The flag for the clean state will always be set in any case where the volume is marked CLEAN.

Table 9-4 describes the volume states, some of which are similar to plex states.

Table 9-4: LSM Volume States

State	Description
ACTIVE	The volume has been started (kstate is currently ENABLED) or was in use (kstate was ENABLED) when the machine was rebooted. If the volume is currently ENABLED, the state of its plexes 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.
CLEAN	The volume is not started (kstate is DISABLED) and its plexes are synchronized.
NEEDSYNC	The volume is not started (kstate is DISABLED) and its plexes are not synchronized. This can occur after a power failure or system failure.
EMPTY	The volume contents are not initialized. The kstate is always DISABLED when the volume is EMPTY.
SYNC	The volume is either in read-writeback mode (kstate is currently ENABLED) or was in read-writeback mode when the machine was rebooted (kstate is DISABLED). 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 needs to be synchronized.

The following subsections describe the different volume states, how they change to indicate abnormalities, and what LSM does to normalize the volume state.

9.2.1 Volume Kernel State

The *volume kernel state* indicates the accessibility of the volume. The volume kernel state allows a volume to have an offline (DISABLED), maintenance (DETACHED), and online (ENABLED) mode of operation. These modes are described in the following table:

Mode	Description
DISABLED	The volume cannot be accessed.
DETACHED	The volume cannot be read or written, but plex device operations and `ioctl` functions are accepted.
ENABLED	The volumes can be read and written.

9.2.2 Volume Usage Types

A *volume usage type* is a type label given to each volume under LSM control. Just as a file system type establishes and enforces policies for file operations, a volume usage type establishes and enforces policies for volume operations. The rules and capabilities differ for different usage types. The volume usage types affect such things as plex synchronization and error handling.

LSM provides the options described in the following table for volume usage types.

Option	Description
fsgen (file system generic)	The `fsgen` usage type assumes the volume is being used by a file system. This usage type assumes there is a way to synchronize file system data to a volume during `volplex` or snapshot procedures. It uses the file system time stamp to see which plex is most up to date. It determines the file system type and calls an appropriate procedure to do a synchronization just prior to the plex split off.
gen (generic)	The `gen` usage type makes no assumptions regarding the data content of the volume. This usage type does not handle synchronization. The `gen` option is useful for databases that reside directly on volumes.

Operations that are dependent on usage-type must determine the usage type of a volume before switching control to a utility customized for that usage type. For example, following a failure, the `gen` and `fsgen` usage types — using different algorithms — guarantee that all plexes of a volume are identical.

Note

Use the `fsgen` usage type when creating a volume if the volume is to be used by a file system. Otherwise, use the `gen` usage type.

9.2.3 Volume Read Policy

Starting a volume changes its state from DISABLED or DETACHED to ENABLED; stopping a volume changes its state from ENABLED or DETACHED to DISABLED.

Table 9-5 describes the three volume-read policies that can be selected.

Table 9-5: LSM Volume Read Policies

Policy	Description
Round	Prescribes round-robin reads of enabled plexes (this is the default read policy). If the read policy for a volume has been set to `round`, the read policy for that volume evenly distributes I/O read requests between all plexes of that volume. Reads are distributed by alternating read requests to each plex in a volume. This policy is preferred in cases where read access performance for all plexes is the same.
Prefer	Prescribes preferential reads from a specified plex. Setting the read policy to `prefer` designates a specific plex of a particular volume to be used for I/O read requests. This policy is preferred if read access performance of one plex is better than the other mirrors. For example, if one of the plexes is striped and the other plexes are concatenated.
Select	Selects a default read policy, based on the plex associations to the volume. If the volume contains a single, enabled, striped plex, the default is to prefer that plex. For any other set of plex associations, the default is to use a round-robin policy.

9.2.4 Managing Available Disk Space

Utilities such as the `volassist` utility obtain information on available disk space and use the information to calculate acceptable layouts for LSM objects. The concept of free space management is based on the idea that free space mapping can be derived by mapping out the existing allocations from the total space on the disk. The methods and specifications for dealing with the allocations and layouts may be provided at the command line; otherwise, they are obtained from defaults specified in a default file or internally.

9.3 Implementing LSM Configuration Changes

When a system administrator makes changes to a set of LSM objects, LSM groups the changes into a transaction. For any transaction, LSM ensures that either all related changes occur successfully or none of the changes are made. LSM makes all configuration changes appear to occur simultaneously and any intermediate stages of change are invisible. If a problem is encountered during the transaction, LSM does not allow any changes to occur and returns the configuration to its original state.

To achieve these atomic, all-or-nothing configuration changes, the LSM volume daemon (the `vold` utility) envelops all configuration changes into a transaction by performing the following steps:

1. Locks all affected objects
2. Gets information about the locked records
3. Records prospective changes
4. Makes all the changes
5. Unlocks the changed objects

The result is that atomic transactions:

- Permit several system administrators to make concurrent changes to the configuration.
- Prevent inconsistent LSM configurations from occurring when there is a system failure.

If a system failure occurs during a transaction, restarting the system causes the `vold` utility to back out the partial changes. This prevents the disks maintained by LSM from becoming inconsistently configured.

Refer to the `vold`(8) reference page for additional information about the volume daemon.

LSM Performance Management 10

This chapter suggests performance priorities and guidelines for use with LSM. It also provides information about monitoring LSM and gathering performance data.

10.1 Performance Strategies

Achieving optimal performance by balancing input/output (I/O) load among several disks on a system without LSM may be limited because it is difficult to anticipate future disk usage patterns, and it is not always possible to split file systems across drives. For example, if a single file system receives most of the disk accesses, placing that file system on another drive moves the bottleneck to another drive.

LSM provides flexibility in configuring storage to improve system performance. Table 10-1 describes two basic strategies available to optimize performance.

Table 10-1: Strategies for Improved Performance

Strategy	Result
Assign data to physical drives to evenly balance the I/O load among the available disk drives	Achieves a finer level of granularity in data placement because LSM provides a way for volumes to be split across multiple drives. After measuring actual data-access patterns, you can adjust file system placement decisions. Volumes can be reconfigured online after performance patterns have been established or have changed, without adversely impacting volume availability.

Table 10-1: (continued)

Strategy	Result
Identify the most-frequently accessed data and increase access bandwidth to that data through the use of mirroring and striping	Achieves a significant improvement in performance when there are multiple I/O streams. If you can identify the most heavily-accessed file systems and databases, then you can realize significant performance benefits by striping the high traffic data across portions of multiple disks, and thereby increasing access bandwidth to this data. Mirroring heavily-accessed data not only protects the data from loss due to disk failure, but in many cases also improves I/O performance.

10.1.1 Improving Mirrored Disk Performance

The use of mirroring to store multiple copies of data on a system improves the chance of data recovery in the event of a system crash or disk failure, and in some cases can be used to improve system performance. However, mirroring degrades write performance slightly. On most systems, data access patterns conform to the 80/20 concept: Twenty percent of the data is accessed 80 percent of the time, and the other 80 percent of the data is accessed 20 percent of the time.

The following sections describe some guidelines for configuring mirrored disks, improving mirrored system performance, and using block-change logging to speed up the recovery of mirrored volumes.

10.1.1.1 Configuring Mirrored Disks for Performance

When properly applied, mirroring can provide continuous data availability by protecting against data loss due to physical media failure. Use the following guidelines when using mirroring:

- Never place subdisks from different mirrors of a mirrored volume on the same physical disk; this action compromises the availability benefits of mirroring and significantly impacts performance.
- To provide optimum performance improvements through the use of mirroring, at least 30 percent of the physical I/O operations should be reads; a higher percentage of read operations results in a higher benefit of performance. Mirroring may provide no performance increase or result in a decrease of performance in a write-intensive workload environment.

Note

The DEC/OSF1 UNIX operating system implements a file system cache. Because read requests frequently can be satisfied from this cache, the read/write ratio for physical I/O's through the file system can be significantly more biased toward writing than the read/write ratio at the application level.

- Where feasible, use disks attached to different controllers when mirroring or striping. Although most disk controllers support overlapped seeks (allowing seeks to begin on two disks at once), do not configure two mirrors of the same volume on disks attached to a controller that does not support overlapped seeks. This is very important for older controllers or SCSI disks that do not do caching on the drive. It is less important for many newer SCSI disks and controllers.
- If one plex exhibits superior performance — either because the disk is being striped or concatenated across multiple disks, or because it is located on a much faster device — then the read policy can be set to the preferred read policy (described in Table 10-2) for the faster plex. By default, a volume with one striped plex should be configured with preferred read of the striped plex.

10.1.1.2 Using Mirroring to Improve System Performance

Mirroring can also improve system performance. Unlike striping, however, performance gained through the use of mirroring depends on the read/write ratio of the disk accesses. If the system workload is primarily write-intensive (for example, greater than 70 percent writes), then mirroring can result in somewhat reduced performance.

Because mirroring is most often used to protect against loss of data due to drive failures, it may be necessary to use mirroring for write-intensive workloads. In these instances, combine mirroring with striping to deliver both high availability and performance.

To provide optimal performance for different types of mirrored volumes, LSM supports the read policies shown in Table 10-2:

Table 10-2: LSM Read Policies

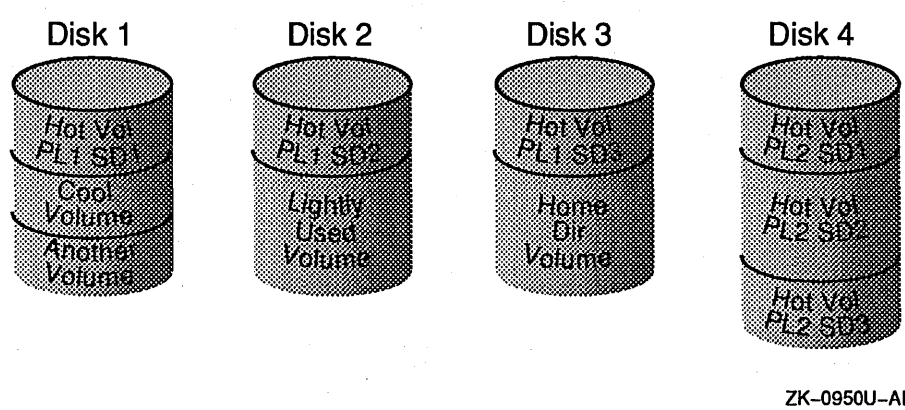
Policy	Description
Round-robin read	Satisfies read requests to the volume in a round-robin manner from all plexes in the volume

Table 10-2: (continued)

Policy	Description
Preferred read	Satisfies read requests from one specific plex (presumably the plex with the highest performance)

For example, in the configuration shown in Figure 10-1, the read policy of the volumes labeled Hot Vol should be set to the preferred read policy from the striped mirror labeled PL1. In this way, reads going to PL1 distribute the load across a number of otherwise lightly used disk drives, as opposed to a single disk drive.

Figure 10-1: Improving System Performance Using Mirroring and Striping



To improve performance for read-intensive workloads, up to eight plexes can be attached to the same volume, although this scenario results in a decrease of effective disk space use. Performance can also be improved by striping across half of the available disks to form one plex and across the other half to form another plex.

10.1.1.3 Improving Mirrored-Volume Recovery with Block-Change Logging

LSM *block-change logging* keeps track of the blocks that have changed as a result of writes to a mirror. Block-change logging does this by identifying the block number of changed blocks, and storing this number in a *log subdisk.* Block-change logging can significantly speed up recovery of mirrored volumes following a system crash.

Note

Using block-change logging can significantly decrease system performance in a write-intensive environment.

Logging subdisks are one-block long subdisks that are defined for and added to a mirror that is to become part of a volume that has block-change logging enabled. They are ignored as far as the usual mirror policies are concerned and are only used to hold the *block-change log*.

Follow these guidelines when using block-change logging:

- Make sure that the subdisk that will be used as the log subdisk does not contain necessary data.
- Ensure that the logging subdisks are one block in length.
- If possible, do not place the log subdisk on a heavily-used disk.
- If persistent (nonvolatile) RAM disks are available, use them for log subdisks.
- Make sure that all mirrors within the volume have a block-change log. If only one plex has a block-change log, logging will be disabled for the volume.

10.1.2 Improving Striped Disk Performance

Striping can improve serial access when I/O exactly fits across all subdisks in one stripe. Better throughput is achieved because parallel I/O streams can operate concurrently on separate devices.

The following sections describe how to use striping as a way of slicing data and storing it across multiple devices to improve access bandwidth for a mirror.

10.1.2.1 Configuring Striped Disks for Performance

Follow these guidelines when using striping:

- Calculate stripe sizes carefully. If it is not feasible to set the stripe width to the track size, use 64 kilobytes for the stripe width, which is the default.
- Avoid small stripe widths; small stripe widths can result in poor system performance *unless* its total width exactly matches the size of the I/O.
- Never put more than one subdisk of a striped mirror on the same physical disk.
- Typically, the greater the number of physical disks in the stripe, the greater the improvement in I/O performance. However, this reduces the

effective mean-time-between-failures (MTBF) of the volume. If this is an issue, striping can be combined with mirroring to provide a high-performance volume with improved reliability.

- If only one mirror of a mirrored volume is striped, be sure to set the policy of the volume to `preferred read` for the striped mirror. (The default read policy, `select`, does this automatically.)
- When striping is used with mirroring, never place subdisks from one mirror on the same physical disk as subdisks from the other mirror.
- If more than one mirror of a mirrored volume is striped, make sure the stripe width is the same for each striped mirror.
- Where possible, distribute the subdisks of a striped volume across drives connected to different controllers and buses.
- Avoid the use of controllers that do not support overlapped seeks.

The `volassist` command automatically adopts many of these rules when it allocates space for striped plexes in a volume.

10.1.2.2 Improving Access Bandwidth with Striped Plexes

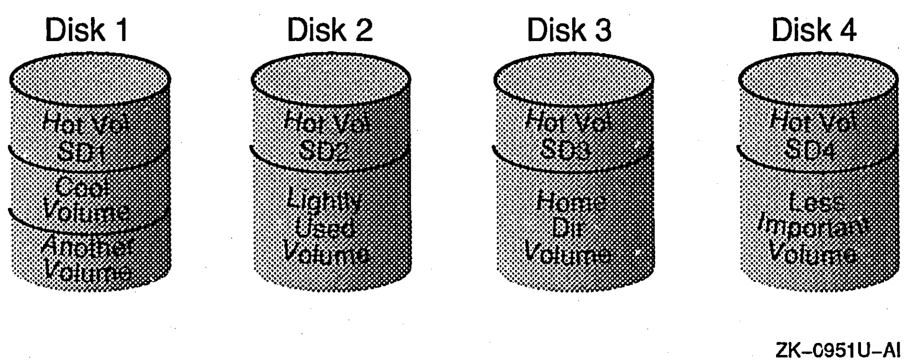
Striping can provide increased access bandwidth for a plex. Striped plexes exhibit improved access performance for both read and write operations. Where possible, disks attached to different controllers should be used to further increase parallelism.

One disadvantage of striping is that some configuration changes are harder to perform on striped plexes than on concatenated plexes. For example, it is not possible to move an individual subdisk of a striped plex, or to extend the size of a striped plex, except by creating a completely new plex and removing the old striped plex. This can be done with the `volassist move` command or the `volplex mv` command.

While these operations can be performed on concatenated plexes without copying through a plex, striping offers the advantage that load balancing can be achieved in a much simpler manner.

Figure 10-2 is an example of a single file system that has been identified as a data-access bottleneck. This file system was striped across four disks, leaving the remainder of those four disks free for use by less-heavily used file systems.

Figure 10-2: Use of Striping for Optimal Data Access



10.2 Monitoring LSM Performance

The following sections suggest ways to prioritize your performance requirements, and how to obtain and use performance data and recorded statistics to help you gain the performance benefits provided by LSM.

Table 10-3 describes the two sets of performance priorities for a system administrator.

Table 10-3: LSM Performance Priorities

Priority	Description
Physical (hardware)	Addresses the balance of the I/O on each drive and the concentration of the I/O within a drive to minimize seek time. Based on monitored results, it may be necessary to move subdisks around to balance the disks.
Logical (software)	Involves software operations and how they are managed. Based on monitoring, certain volumes may be *mirrored* (multiple plexes) or *striped* to improve their performance. Overall throughput may be sacrificed to improve the performance of critical volumes. Only you can decide what is important on a system and what tradeoffs make sense.

10.2.1 Statistics Recorded by LSM

LSM records the following three I/O statistics:

- A count of operations
- The number of blocks transferred (one operation could involve more than one block)
- The total active time

LSM records these statistics for logical I/Os for each volume. The statistics are recorded for the following types of operations: reads, writes, atomic copies, verified reads, verified writes, mirror reads, and mirror writes.

For example, one write to a two-mirror volume will result in the following statistics being updated:

- One operation for each plex
- One operation for each subdisk
- One operation for the volume

Similarly, one read that spans two subdisks results in the following statistics being updated: — one read for each subdisk, one for the mirror, and one for the volume.

LSM 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.

10.2.2 Gathering Performance Data

LSM provides two types of performance information — I/O statistics and I/O traces:

- I/O statistics are retrieved using the `volstat` utility
- I/O tracing can be retrieved using the `voltrace` utility

Each type of performance information can help in performance monitoring. The following sections briefly discuss these utilities.

10.2.2.1 Obtaining I/O Statistics

The `volstat` utility provides access to information for activity on volumes, plexes, subdisks, and disks under LSM control. The `volstat` utility reports statistics that reflect the activity levels of LSM objects since boot time. Statistics for a specific LSM object or all objects can be displayed at one time. A disk group can also be specified, in which case statistics for objects in that disk group only are displayed; if you do not specify a particular disk group on the `volstat` command line, statistics for the default disk group (rootdg) are displayed.

The amount of information displayed depends on what options are specified to volstat. For detailed information on available options, refer to the volstat(8) reference page.

The volstat utility is also capable of resetting the statistics information to zero. This can be done for all objects or for only those objects that are specified. Resetting just prior to a particular operation makes it possible to measure the impact of that particular operation afterwards.

The following example shows typical output from a volstat display:

```
                      OPERATIONS          BLOCKS         AVG TIME(ms)
TYP NAME            READ    WRITE     READ    WRITE   READ  WRITE
vol blop               0        0        0        0    0.0    0.0
vol foobarvol          0        0        0        0    0.0    0.0
vol rootvol        73017   181735   718528  1114227   26.8   27.9
vol swapvol        13197    20252   105569   162009   25.8  397.0
vol testvol            0        0        0        0    0.0    0.0
```

10.2.2.2 Tracing I/O Operations

The voltrace command is used to trace operations on volumes. Through the voltrace utility, 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 utility 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, refer to the voltrace(8) reference page.

10.2.3 Using Performance Data

Once performance data has been gathered, you can use the data to determine an optimum system configuration that makes the most efficient use of system resources. The following sections provide an overview of how you can use I/O statistics and I/O tracing.

10.2.3.1 Using I/O Statistics

Examination of the I/O statistics may suggest reconfiguration. There are two primary statistics to look at: volume I/O activity and disk I/O activity. The following steps describes how to record and examine I/O statistics:

1. Before obtaining statistics, consider clearing (resetting) all existing statistics. Clearing statistics eliminates any differences between volumes

or disks that might appear due to volumes being created, and also removes statistics from booting (which are not normally of interest). Use the following command to clear all statistics:

```
volstat -r
```

2. After clearing the statistics, gather I/O statistics during a time span when typical system activity is occurring. You may also want to gather I/O statistics during the time that you run a specific application to obtain statistics specific to that application or workload.

   When monitoring a system that is used for multiple purposes, try not to exercise any application more than it would be exercised under typical circumstances. When monitoring a time-sharing system with many users, try to let the I/O statistics accumulate during typical usage for several hours during the day.

3. To display volume statistics, enter the `volstat` command without any arguments. The output might appear as shown in the following example:

```
                      OPERATIONS         BLOCKS         AVG TIME(ms)
TYP NAME            READ   WRITE     READ     WRITE    READ    WRITE
vol archive          865     807     5722      3809    32.5     24.0
vol home            2980    5287     6504     10550    37.7    221.1
vol local          49477   49230   507892    204975    28.5     33.5
vol src            79174   23603   425472    139302    22.4     30.9
vol swapvol        22751   32364   182001    258905    25.3    323.2
```

4. To display disk statistics, enter the `volstat -d` command. The resulting output might appear as shown in the following example:

```
                    OPERATIONS          BLOCKS          AVG TIME(ms)
TYP NAME          READ   WRITE       READ     WRITE    READ    WRITE
dm  disk01       40473  174045     455898    951379    29.5     35.4
dm  disk02       32668   16873     470337    351351    35.2    102.9
dm  disk03       55249   60043     780779    731979    35.3     61.2
dm  disk04       11909   13745     114508    128605    25.0     30.7
```

5. Check the displays for volumes with an unusually large number of operations or excessive read or write times.

6. To move the volume `archive` onto the boot disk (`disk01` in the previous example), identify which disks it is on using the `volprint -tvh archive` command. The display from this command is similar to the following example:

```
V  NAME        USETYPE     KSTATE  STATE     LENGTH READPOL...
PL NAME        VOLUME      KSTATE  STATE     LENGTH LAYOUT...
SD NAME        PLEX        PLOFFS  DISKOFFS  LENGTH DISK-MEDIA...

v  archive     fsgen       ENABLED ACTIVE    204800 SELECT    -
pl archive-01  archive     ENABLED ACTIVE    204800 CONCAT    -  RW
sd disk03-03   archive-01  0       409600    204800 disk03      rz2
```

Looking at the associated subdisks indicates that the `archive` volume is on disk `disk03`. To move the volume off `disk03` and onto `disk01`, use one of the following commands:

a. For K-shell users, enter:

   volassist move archive !disk03 disk01

b. For C-shell users, enter:

   volassist move archive disk03 disk01

These commands indicate that the volume should be reorganized so that no part is on `disk03`, and that any parts to be moved should be moved to `disk01`.

Note

> The easiest way to move pieces of volumes between disks is to use the Logical Storage Manager Visual Administrator (dxlsm). If dxlsm is available on the system, you may prefer to use it instead of the command-line utilities.

7. If there are two busy volumes, try to move them so that each volume is on a different disk.
8. If there is one volume that is particularly busy (especially if it has unusually large average read or write times), consider striping the volume (or splitting the volume into multiple pieces, with each piece on a different disk). Converting a volume to use striping requires sufficient free space to store an extra copy of the volume.

   To convert to striping, create a striped mirror of the volume and then remove the old mirror. For example, to stripe the volume `archive` across disks `disk02` and `disk04`, enter the following commands:

   volassist mirror archive layout=stripe disk02 disk04
   volplex -o rm dis archive-01

9. After reorganizing any particularly busy volumes, check the disk statistics. If some volumes have been reorganized, first clear statistics and then gather statistics for a sufficient period of time.

IF...	THEN...
Some disks appear to be used excessively (or have particularly long read or write times)	Reconfigure some volumes.

IF...	THEN...
There are two relatively busy volumes on a disk	Consider moving the volumes closer together to reduce seek times on the disk.
There are too many relatively busy volumes on one disk	Try to move the volumes to a disk that is less busy.

10. Use I/O tracing (or perhaps subdisk statistics) to determine whether volumes have excessive activity in particular regions of the volume. If such regions can be identified, try to split the volume and to move those regions to a less busy disk.

Note

File systems and databases typically shift their use of allocated space over time, so this position-specific information on a volume often is not useful. For databases, it may be possible to identify the space used by a particularly busy index or table. If these can be identified, they are reasonable candidates for moving to disks that are not busy.

11. Examine the ratio of reads and writes to identify volumes that can be mirrored to improve their performance.

IF...	THEN...
The read-to-write ratio is high	Mirroring could increase performance as well as reliability. The ratio of reads to writes where mirroring can improve performance depends greatly on the disks, the disk controller, whether multiple controllers can be used, and the speed of the system bus.
A particularly busy volume has a ratio of reads to writes as high as 5:1	It is likely that mirroring can dramatically improve performance of that volume.

Warning

Use mirroring to substantially reduce the chances that a single disk failure will result in failure of a large number of volumes. This is because 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 will require that all five volumes be restored from a backup. If each volume were on a separate disk, only one volume would have to be restored.

10.2.3.2 Using I/O Tracing

Whereas I/O statistics provide the data for basic performance analysis, I/O traces provide for more detailed analysis. With an I/O trace, the focus of the analysis is narrowed to obtain an event trace for a specific workload. For example, you can identify exactly where a hot spot is, how big it is, and which application is causing it.

By using data from I/O traces, you can simulate real workloads on disks and trace the results. By using these statistics, you can anticipate system limitations and plan for additional resources.

Logical Storage Manager Error Messages A

The Logical Storage Manager 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. When `vold` is unable to recognize and fix system problems, you need to handle the task of problem solving.

The following sections cover 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.

A.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 since 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 LSM. 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 preceeding 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 LSM 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 since 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 unselected 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 reference page for `voldisk`(8). 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 reccur.

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 since 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.

A.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 disk *disk_name* **in group** *disk_group*.
   If it is removable media (like a floppy), it may not be mounted or ready. Otherwise, there may be problems with the drive.
   Kernel error code *error_number*

   `Clarification:` The named disk cannot be accessed in the named disk group.

   `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.

Recovery B

Disk failures can cause two types of problems: loss of data on the failed disk and loss of access to your system due to the failures of a key disk (a disk involved with system operation). LSM provides the ability to protect your system from either type of problem. LSM uses *mirroring* to protect the data. By mirroring your data, you prevent data loss from a disk failure. By mirroring drives critical to booting, you ensure that no single disk failure will leave your system unusable.

B.1 Protecting Your System

To maintain system availability, the data important to running and booting your system must be mirrored. Furthermore, it must be preserved in such a way that it can be used in case of failure. The most difficult part of this is the ability to boot the system after a failure of a disk that is critical to the boot process.

To preserve data, create and use volumes that have at least two mirrors (plexes). The mirrors must be on different disks and preferably on different controllers for an added margin of safety. The `volassist` utility locates the mirrors such that the loss of one disk will not result in a loss of data. Edit the file `/etc/default/volassist` to set the default number of mirrors for newly created volumes to two.

You must also do regular backups (of all data except the root file system) to protect your data. Backups are necessary if all copies of a volume are lost or corrupted in some way. For example, a power surge could damage several (or all) disks on your system. Alternately, a mistyped command could remove critical files or damage a file system directly. For information about backing up your data, see Section 4.6.9 to use the Visual Administrator, or Section 7.3.8 to use the `volassist` command for backup.

B.2 Reinstallation Recovery

Occasionally, your system may need to be reinstalled after some types of failures. Reinstallation is necessary if all copies of your root (boot) disk are damaged, or if certain critical files are lost due to file system damage. When a failure of either of these types occurs, you must reinstall the entire system, since there is currently no method of restoring the root file system from backup.

Note

> Support for LSM volumes for root file system is not currently available.

If these types of failures occur, you should attempt to preserve as much of the original LSM configuration as possible. Any volumes not directly involved in the failure may be saved. You do not have to reconfigure any volumes that are preserved.

This section describes the procedures used to reinstall LSM and preserve as much of the original configuration as possible after a failure.

B.2.1 General Recovery Information

System reinstallation completely destroys the contents of any disks that are reinstalled. Any LSM related information, such as data in the LSM private areas on removed disks (containing the disk identifier and copies of the LSM configuration), is removed during reinstallation. The removal of this information makes the disk unusable as an LSM disk.

If a disk was placed under LSM control (either during the LSM installation or by later encapsulation), that disk and any volumes or volume plexes on it are lost during reinstallation. If a disk is not under LSM control prior to the failure, no volumes are lost at reinstallation. Any other disks to be replaced can be replaced following the procedures in Section 6.2.6 and Section 7.2.12 for information using the LSM Support Operations or the LSM command-line interfaces respectively.

When reinstallation is necessary, the only volumes saved are those that reside on, or have copies on, disks that are not directly involved with the failure, the reinstallation, or both; volumes on disks involved with the failure or reinstallation are lost during reinstallation. If backup copies of these volumes are available, you can restore these volumes after reinstallation.

B.2.2 Reinstallation and Reconfiguration Procedures

To reinstall the system and recover the LSM configuration, perform the following procedure:

1. Prepare the system for installation. This includes replacing any failed disks or other hardware, and detaching any disks not involved in the reinstallation.
2. Install the operating system. Do this by reinstalling the base system and any other non-LSM packages.

3. Install LSM.
4. Recover the LSM configuration.
5. Cleanup the configuration. This includes restoring any information in volumes affected by the failure or reinstallation.

The previous steps are described in the following sections.

B.2.2.1 Preparing the System for 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, it may still be recoverable. Removing the second drive ensures that the file system remains intact.

B.2.2.2 Reinstalling the Operating System

Once any failed or failing disks have been replaced and disks uninvolved with the reinstallation have been detached, reinstall the operating system as described in the manuals for your operating system. Install the basic operating system prior to installing LSM.

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.

Note

Several of the *Automatic* options for installation access drives other than the root drive without requiring confirmation from the administrator. Therefore, it is advised that you disconnect all other disks from the system prior to installing the operating system.

B.2.2.3 Reinstalling LSM

The installation of LSM has two parts:

- Loading LSM from CD–ROM or over the network
- Initializing LSM

If you want to reconstruct the LSM configuration left on the nonroot disks, *do not* initialize LSM after the reinstallation.

To reinstall LSM, follow the instructions for loading LSM (from CD–ROM or over the network) in the *Installation Guide* and the instructions in Section B.2.2.4.

B.2.2.4 Recovering the LSM Configuration

Once the LSM package has been loaded recover the LSM configuration by doing the following:

1. Shut down the system.
2. Reattach the disks that were removed from the system.
3. Reboot the system.
4. When the system comes up, bring the system to single-user mode by entering the following command:

   shutdown now

5. You need to remove some files involved with installation that were created when you loaded LSM but are no longer needed. To do this, enter the command:

   rm -rf /etc/vol/bin/install-db

6. Once these files are removed, you must start some LSM daemons. Start the daemons by entering the command:

   /sbin/voliod set 2

7. Start the LSM Configuration Daemon, `vold`, by entering the command:

   /sbin/vold -m disable

8. Initialize the `vold` daemon by entering:

   voldctl init

9. Add one or more disks that have configuration databases to the `/etc/vol/volboot` file. You must do this otherwise LSM cannot restart after a reboot.

   To reenable the previous LSM configuration, you need to determine the name of one of the disks that was in the `rootdg` disk group. If you do not know the name of one of the disks, you can scan the disk label on the disks available on the system for LSM disk label tags such as `LSMpubl` or `LSMsimp`. If you find the `LSMpubl` disk label tag on a disk, add the disk as an LSM sliced disk. If you find the `LSMsimp` disk label tag, add the partition as an LSM simple disk.

voldctl add disk rz3

10. Enable `vold` by entering:

voldctl enable

The configuration preserved on the disks not involved with the reinstallation has now been recovered.

Visual Administrator Reference C

Various Visual Administrator operations and options can be accessed using the main menus located in the menu bars of the Visual Administrator root window, view windows, or both. Some of these menus lead to submenus; some lead to forms; and some simply execute operations directly.

This chapter provides reference information for managing LSM volumes with the LSM Visual Administrator.

C.1 Introduction and Setup

C.1.1 Views

The LSM Visual Administrator root window contains a Views menu that is used to create and modify view windows. The following list describes the views available using the Views menu (located in the LSM Visual Administrator root window only).

- Create a View

  Views → Create a View

  `Description:` Creates a new user-specified view (commonly referred to as a *user-created view* or a *user view*). The user specifies a unique name for this new view. Once created, icons can be added to this new view window by copying them over from existing views.

  User-created views differ from default views (disk, volume, disk group, and world views) in that they contain copies of icons from the default views. Any operations performed from user-created views are automatically activated in the corresponding default views. Icons that appear in default views are not necessarily duplicated in user-created views.

- Rename a View

  Views → Rename a View

  `Description:` Renames an existing user-created view. The user specifies a new, unique name for the view.

  `Requirement:` Only a user-created view can be renamed.

- Remove Views

  Views → Remove Views

  `Description:` Permanently removes a user-created view. The removal of a user-created view results in the removal of its icons, but the objects represented by those icons are unaffected.

  `Requirement:` Only a user-created view can be removed.

C.1.2 Views Forms

The following list shows the forms accessed via the Views menu (located in the LSM Visual Administrator root window only).

- Create a View

  Views → Create a View

  `Description:` Creates a new view window. Once created, this view is represented by a new view button in the Visual Administrator root window. This user-created view is also retained across future Visual Administrator sessions. The following table describes the field for this form.

Field	Description
View Name:	The name of the view to be created. This name must be unique. The maximum length of the name is 14 characters. This field is required.

  `Requirement:` The name specified for the new view must be unique.

- Rename View Form

  Views → Rename a View

  `Description:` This form is used to change the name of an existing user-created view. The following table describes the fields for this form.

Field	Description
Old View Name:	The name of the view to be renamed. This name must belong to a user-created view. This field is required.
New View Name:	The name to which the existing name is to be changed. This name must be unique. The maximum length of the name is 14 characters. This field is required.

`Requirements:` Only user-created views can be renamed, and the new view name must be unique.

- Remove View Form

  Views → Remove a View

  `Description:` This form is used to remove a user-created view. This form contains no fields. Instead, it lists all user-created views that currently exist.

  The user selects one or more views for removal by highlighting the desired view names with the MB1 button. If a view is mistakenly selected, the MB1 button can be used again to clear that view name. Removal is activated by clicking MB1 on the Apply button once, then confirming the removal by selecting Apply again (this ensures that a view is not accidentally removed).

C.1.3 Options

User preferences for how the LSM Visual Administrator should operate are set via the Options menu.

Once set, most preferences for each user are saved to the `.dxlsm_pref`, file located in each user's home directory. The `.dxlsm_pref` file should not be edited by users. Once set, these preferences are maintained across future Visual Administrator session.

C.1.3.1 Options Menu

The following list shows the menu selections accessible via the Options menu:

- Show Command

  Options → Show Command

  `Description:` This menu selection is used to specify whether the Command Info Window is to be displayed before each command is executed or only upon command failure.

  From the Show Command menu, a submenu allows the user to indicate when to display the Command Info Window. The following table describes the available options.

Option	Description
Show on Error	Display the Command Info Window only when an LSM or system command has failed. This is the default behavior.

Option	Description
Show at Start	Display the Command Info Window whenever an LSM or system command is ready to be sent. This enables the user to view the actual commands being executed via the Visual Administrator.

- When Commands Are Ready

  Options → When Commands Are Ready

  `Description:` This menu selection is used to specify whether commands should be executed immediately or simply displayed for user review.

  From the When Commands Are Ready menu, a submenu allows the user to indicate what the Visual Administrator is to do when a command is issued:

  Execute Commands
  : Automatically execute commands as soon as they are issued. This is the default behavior.

  Show Commands Only
  : Display commands in the Command Info Window for user review rather than executing them. Upon approval, the user can execute the displayed command directly from the Command Info Window by highlighting that command and then using the Execute menu.

- Logging

  Options → Logging

  `Description:` This menu selection is used to start or stop logging of Visual Administrator commands. Logging records all commands sent to LSM or the system by the LSM Visual Administrator in a specified file.

  From the Logging menu, a submenu allows you to indicate whether logging should be activated or deactivated:

Option	Description
Start	Begin recording all commands to a log file. A log file is created if one does not already exist. The file to be used for logging must be specified in the resulting form. If the file exists, the user must have permission to write to that file. The log information will be appended to the end of the specified file.

Option	Description
Stop	Stop recording all commands to the log file. When logging is discontinued, the user is responsible for remembering the name of the log file that was used.

Note

Unlike other user preferences, the logging setting is not saved across Visual Administrator sessions.

`Form:` Log File Form (described in Section C.1.3.2).

- Popup the Command Window

  Options → Popup the Command Window

  `Description:` This menu selection is used to access and display the Command Info Window. This window displays current and previous commands, along with the status of each command. Once accessed in this way, the Command Info Window remains visible until it is closed via its File menu.

  See Section C.1.4 for further details.

- Format of Size

  Options → Format of Size

  `Description:` This menu selection is used to specify the units (megabytes, kilobytes, or sectors) to be used for size-related output. The unit of size is set to megabytes until the user resets it.

  In properties forms, length values are displayed as sectors (s), kilobytes (k), or megabytes (m). If the size cannot be cleanly converted into kilobytes or megabytes, it is displayed in sectors instead (even though another format of size preference may be set).

  The preferred format of size applies to output only and does not impact input in any way. Input typically defaults to sectors, unless megabytes or kilobytes are specified.

  From the Format of Size menu, a submenu allows the user to select the units of size:

Units	Description
Mbytes	Use megabytes when displaying size-related output.
Kbytes	Use kilobytes when displaying size-related output.
Sectors	Use sectors when displaying size-related output. This is the default.

C.1.3.2 Options Forms

The following forms are accessible via the Options menu:

- Log File Form

  Options → Logging → Start

  `Description:` This form is used to specify the file to be used for logging purposes. The following table describes the field for this form.

Field	Description
Log File:	The name of the file (and path name) to be used to store the command log. If no path is specified, the file is created in the directory from which the Visual Administrator session was started. The maximum length of the path specified here is 127 characters.

`Requirements:`

- The user must have privileges appropriate to access and write to the named file (and any directories in its path).
- If a path name is included, it must be valid.

C.1.4 Command Info Window

You access the Command Info Window via the Options menu. Once accessed, this window displays information on current and previous commands executed by the Visual Administrator.

In addition to viewing or previewing commands, you can use the Command Info Window to execute commands directly through its Execute menu. This is useful for reexecuting commands or for executing commands that were previously only shown in this window. The following list describes using the Execute and Execute With Force options:

- Execute

Execute → Execute

Description: This menu selection is used to execute the command highlighted in the Command History section of the window. The Visual Administrator sends the highlighted command to LSM or the system for execution.

Requirement: A single command in the Command History section of the window must be highlighted.

- Execute With Force Option

  Execute → Execute With Force Option

  Description: This menu selection is used to forcefully execute the LSM command highlighted in the Command History section of the window (using the `-f` option). This option effectively forces LSM to complete an operation that is considered unsafe and should therefore be used only when the user is certain that an operation should be performed in this way.

  Note

  This is a dangerous operation that can cause irreparable loss of data. Use this option only when absolutely necessary.

  Requirements:

  – The force option works for some LSM commands only (those that take the `-f` option). The `-f` option does not apply to file system operations offered through the Visual Administrator.

  – Use this option only when you are sure that an operation will succeed, even though LSM error checking prevents it.

C.1.5 Icons

You can access a set of icon-related options via the Icon menu located in views.

The following list shows the menu selections you can access via the Icon menu:

- Maximize Icons

  Icon → Maximize Icons

  Description: This operation maximizes the selected minimized icons, making it show all of its subicons.

  Requirement: At least one minimized icon must be selected.

- Minimize Icons

  Icon → Minimize Icons

  `Description:` This operation minimizes the selected icons, making it shrink down in size and hide all of its subicons.

  Minimized icons occupy less space and are displayed with their names in reverse type.

  `Requirement:` You must select at least one maximized icon.

- Maximize All Icons

  Icon → Maximize All Icons

  `Description:` This operation maximizes all icons in the current view window, making them show all of their subicons. No icons need to be selected.

- Create Icons

  Icon → Create Icons

  `Description:` This operation creates a copy of the icons selected from another view and places the new copy in the current user-created view. Icons that already exist in this user-created view will not be duplicated.

  `Requirement:` This option is only available in user-created views.

- Remove Icons

  Icon → Remove Icons

  `Description:` This operation removes the selected icons from the current user-created view.

  `Requirement:` This option is only available in user-created views.

C.1.6 Help

The LSM Visual Administrator provides an extensive Help facility, which is accessible from most windows or operations.

C.1.6.1 Help from Menus

Almost all Visual Administrator windows contain a Help menu that provides access to information relevant to the contents of that window. The Help menu contains a list of available Help options, each of which brings up a Help window containing information on the selected topic.

Most Visual Administrator menus contain a Help option (listed at the bottom of the menu). When the Help option is selected from a given menu, a Help window containing information relevant to the operations or options listed in the menu appears.

C.1.6.2 Help from Forms

Visual Administrator forms contain Help buttons that provide information relevant to the form and its fields. Help is accessed by clicking MB1 on a form's Help button.

C.1.6.3 Navigating Help Windows

Regardless of where you try to access Help, LSM displays a Help window that includes text relevant to the topic.

From a given Help window, the user can access different Help topics in any of the following ways:

- Click on any Help topic listed in the current Help screen's see ALSO section (located at the bottom of the Help window). The existing Help text is immediately replaced with that of the selected new topic.
- Click on Previous in the Help window's menu bar. The existing Help text is immediately replaced with that of the Help topic that was previously being viewed.
- Click on Next in the Help window's menu bar. The existing Help text is immediately replaced with that of the Help topic that was being viewed before skipping backwards to a previous topic.

C.1.6.4 Help Index

A Help index can be accessed via the Help menu located in the menu bar of the Help window itself. The Help index lists all available help topics for the Visual Administrator. Once identified, any of the listed Help topics can be accessed by clicking MB1 on the matching topic listed in the SEE ALSO section of the Help window (directly below the Help index listing).

C.1.7 Exiting the Visual Administrator and its Windows

Most Visual Administrator windows contain a File menu, which is used to exit that particular Visual Administrator window or to exit the session completely.

The following list describes the menu selections you can access via the File menu:

- Exit

  File → Exit

  `Description` This menu selection both closes the current window and exits the Visual Administrator completely. Although the session is closed down, user preferences set during the session are retained. Because the Visual Administrator operations are applied to the Visual Administrator configuration as they are issued, quitting a session has no effect on the configuration and does not undo any changes made through the Visual Administrator.

- Close

  File → Close

  `Description:` This menu selection closes the current window only.

C.2 File System Operations

This section provides information on menus and forms relating to UFS operations. You access UFS operations via the Basic-Ops menu. This menu is located in view windows, such as View of rootdg. This menu provides access to UFS operations involving general file system maintenance, and is accessed as shown:

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.2.1 File System Menus

The following list describes the file system operations menu items:

- Create

  Basic-Ops → UFS Operations → Create

  `Description:` 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 (described in Section C.2.2).

- Make File System

  Basic-Ops → UFS Operations → Make File System

  `Description:` 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.2.2).

- Mount

  Basic-Ops → UFS Operations → Mount

  `Description:` 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.2.2).

- Unmount

  Basic-Ops → UFS Operations → Unmount

  `Description:` 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)

  `Description:` 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.2.2).

- Display Properties

  Basic-Ops → UFS Operations → Display Properties

  `Description:` Display information for file systems mounted on the system. The user may 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.2.2 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.2.2.1 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

  `Description:` 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.

Note

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 14 characters.

Field	Description
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

  `Description:` 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.

Note

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 14 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, Logical Storage Manager 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 drive configuration, as best striping performance is achieved when the stripe width corresponds to the track width of the drive. The default value for this field is 128 sectors, chosen 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.

Field	Description
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.

- Make File System Form

  Basic-Ops → UFS Operations → Make

  `Description:` This form is used to make a file system (using `newfs`) according to the user's specifications. The following table describes the fields for this form.

Note

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:	This field 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:	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.

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`). Yes is the default.

- Mount File System Form

  Basic-Ops → UFS Operations → Mount

  `Description:` 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.

Note

> 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:	This field 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)

  `Description:` 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.

Note

The fields in this form are required. Fields in this form are read/write fields, unless listed as read only.

Field	Description
Volume:	This fields 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:	This fields indicates the type of the file system to be checked.

C.2.2.2 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

  `Description:` 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 the user can select the file system whose properties are to be displayed. The following table describes the fields for this form.

Note

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.

Field	Description
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 is available for use.
% disk space available:	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:	This is the maximum number of characters that a file name may be on this file system. This restriction is imposed by the file system.
FS attributes:	This field indicates attributes associated with this file system: `Read Only` indicates a file system that cannot be written to.

C.3 Volume Operations

This section provides information on menus and forms relating to volume operations.

C.3.1 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 relevant to the items and operations listed in that particular menu.

C.3.1.1 Basic-Ops Menu

You access the Basic-Ops menu as shown:

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 Help selection accesses a Help window, which displays information relevant to the basic volume operations.

The following list describes these menu selections:

- Create

  Basic-Ops → Volume Operations → Create

  `Description:` This operation creates a volume from one or more disks. The user may 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, LSM automatically determines which disks are to be used 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.

  If a mirrored volume is desired, a simple or striped volume must be created and then mirrored 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 (described in Section C.3.2).

- Remove Volumes Recursively

  Basic-Ops → Volume Operations → Remove Volumes Recursively

  `Description:` 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 (i.e. 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

  `Description:` 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 true mirroring (redundancy of data) to be in effect.

  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.

Type	Description
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.

Disks can be selected 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 LSM.

`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

  `Description:` 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

  `Description:` 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. A disk cannot be selected for this operation.

  `Requirements:`

  - A volume icon must be selected.
  - A striped volume cannot be resized and therefore cannot be selected.
  - A volume containing a mounted file system cannot be shrunk.

  `Form:` Volume Resize Form (described in Section C.3.2).

- Snapshot

  Basic-Ops → Volume Operations → Snapshot

  `Description:` 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, the snapshot volume can be removed without losing any data.

  Note

  For UFS volumes, Digital recommends 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 greyed out.
Snapshot	At a convenient time (preferably after warning users to reduce 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 (described in Section C.3.2).

C.3.1.2 Advanced-Ops Menu

You access the Advanced-Ops menu selections as shown:

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
- Help

The Help selection accesses a Help window, which displays information relevant to the advanced volume operations.

The following list describes these menu selections:

- Create

  Advanced-Ops → Volume → Create

  `Description:` This operation creates a volume. The user may select one or more plexes to be associated with the new volume after creation.

  `Form:` Volume Create Form (described in Section C.3.2).

- Remove Volumes

  Advanced-Ops → Volume → Remove Volumes

  `Description:` 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

  `Description:` 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

  `Description:` 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

  Description: 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

  Description: This operation brings all plexes within the selected volumes up to date. Any plexes that are inconsistent are resynchronized to contain consistent data.

  Depending on how currrent the plexes are, this operation may take some time.

  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

  Description: 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.

C.3.2 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.3.2.1 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

  Description: 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 greyed out because the default is not to add a file system to the volume. The following tables describes the fields for this form.

Note

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 14 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.

Field	Description
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

  `Description:` 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 greyed out because the default is not to add a file system to the volume. The following table describes the fields for this form.

Note

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 14 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 drive configuration, as best striping performance is achieved when the stripe width corresponds to the track width of the drive. The default value for this field is 128 sectors, chosen as 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

  Basic-Ops → Volume Operations → Resize

  `Description:` 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.

  Note

  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.
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.

Field	Description
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

  `Description:` This form creates a snapshot of the selected volume for backup purposes. The following table describes the fields for this form.

  Note

  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:	This field displays 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 14 characters. The snapshot name must be unique.

`Requirement:` There must be sufficient free space to accommodate the snapshot volume.

C.3.2.2 Advanced-Ops Forms

The following forms are accessed via volume-related selections from the Advanced-Ops menu:

- Volume Create Form

  Advanced-Ops → Volume → Create

  `Description:` This form creates a volume according to the user's specifications. The following table describes the fields for this form.

Note

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 14 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 14 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 14 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.

Field	Description
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.
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 No.
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. This field is optional.
Putil2:	Permanent utility field 2. This field is reserved, but may be changed. This field is optional.

C.3.3 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 mouse button on the desired volume icon. (If volume icon is undergoing analysis, press Shift-MB3 instead.)

  `Description:` This form provides detailed information on the attributes of a particular volume. The following table describes the fields in this form.

Note

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 14 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 14 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 14 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.

Field	Description
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.
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=valuepairs.*
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 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.

C.4 Plex Operations

This chapter provides information on menus and forms relating to plex operations.

C.4.1 Plex Menus

You access plex operations via the Advanced-Ops menu, as shown here:

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

  `Description:` 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 (described in Section C.4.2).

- Remove Plexes

  Advanced-Ops → Plex → Remove plexes

  `Description:` 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

  `Description:` 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 a fair amount of 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

`Description:` 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

`Description:` 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

`Description:` 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.4.2 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

  `Description:` The following table describes the fields in this form.

Note

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 14 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 14 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 scatters 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 Striped plex layout has been specified, this field is required. This field must be blank if Concatenated plex layout has been specified.

Field	Description
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.
Putil1:	Permanent utility field 1. This field is reserved, but may be changed. This field is optional.
Putil2:	Permanent utility field 2. This field is reserved, but may be changed. This field is optional.

C.4.3 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.

  `Description:` This form provides detailed information on the attributes of a particular plex. The following table describes the fields in this form.

Note

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 14 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.

Field	Description
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 scatters 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.
Log Subdisk:	This field shows the name of the subdisk that is being used for logging on this plex. If there is no associated Block Change 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 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 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.
Tutil2:	Temporary utility field 2. This field is reserved, but may 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 I/O operations that have failed on this plex. This field is read only and cannot be changed.

C.5 Subdisk Operations

This chapter provides information on menus and forms relating to subdisk operations.

C.5.1 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

  `Description:` This operation creates a subdisk on the selected LSM disk.

  `Requirement:` An LSM disk must be selected.

  `Form:` Subdisk Create Form (described in Section C.5.2).

- Remove Subdisks

  Advanced-Ops → Subdisk → Remove Subdisks

  `Description:` 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

Description: 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

  Description: This operation associates the selected subdisk as a log subdisk with the selected plex. Block Change Logging is in effect. 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.

- Disassociate Subdisks

  Advanced-Ops → Subdisk → Disassociate Subdisks

  Description: 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

  Description: 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

`Description:` 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 and subdisks associated with striped plexes cannot be split.

`Forms:` Subdisk Split Into Two

Subdisk Split Into Many (described in Section C.5.2).

C.5.2 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

`Description:` This form creates a subdisk according to the user's specifications. The following table describes the fields in this form.

Note

> 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 14 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.
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 14 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.

Field	Description
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

  `Description:` This form is used to split the selected subdisk into exactly 2 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.

  Note

  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

  `Description:` 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.

Note

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.5.3 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 the MB3 mouse button on desired subdisk icon.

  (If the subdisk is undergoing analysis, press Shift-MB3 instead.)
  `Description:` This form provides detailed information on the attributes of a particular subdisk. The following table describes the fields in this form.

Note

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 14 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 Block Change 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.
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 I/O operations that have failed on this subdisk. This field is read only and cannot be changed.

C.6 Disk Operations

This chapter provides information on menus and forms relating to disk operations.

C.6.1 Disk Menus

You access disk operations using the Advanced-Ops menu, as shown here:

Advanced-Ops → Disk

The Advanced-Ops menu provides access to the following disk-related menus:

- 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

  `Description:` 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

  `Description:` 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

  `Description:` 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

  `Description:` 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

  `Description:` 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.6.2 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.

The following forms are accessed via disk-related selections from the Advanced-Ops menu:

- Disk Init Form

  Advanced-Ops → Disk → Initialize

  `Description:` This form is used to initialize a disk for LSM use. The following table describes the fields in this form.

  Note

  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`. A name in the form `rz`*nn* is used to assign the full disk under LSM control. The disk `rz`*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 `rz`*nnp* is used to assign partition *p* on disk `rz`*nn* under LSM control. The disk partition `rz`*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. Before initializing the disk, change the disklabel to have LSM disklabel tags. 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, 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.
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 2 copies.

Field	Description
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

`Description:` This form is used to define a disk. The following table describes the fields in this form.

Note

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`. A name in the form `rz`*nn* is used to assign the full disk under LSM control. The disk `rz`*nn* would be added as a sliced LSM disk. A name in the form `rz`*nnp* is used to assign partition *p* on disk `rz`*nn* under LSM control. The disk partition `rz`*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

  `Description:` This form is used to online a disk. The following table describes the fields in this form.

Field	Description
Device name:	The disk access name of the disk to be onlined. This must be a valid disk access name. This field is required.

- Free Space Form

  To access the free space form, click the MB3 mouse button on a gap between subdisk icons in a LSM disk icon.

  `Description:` 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.

Note

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.6.3 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 the MB3 mouse button on desired LSM disk icon. (If the LSM disk icon is undergoing analysis, press Shift-MB3 instead.)

  `Description:` 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.

Note

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.

Field	Description
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.
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 the MB3 mouse button on desired physical disk icon.

  `Description:` This form provides detailed information on the attributes of a particular physical disk. The following table describes the fields in this form.

Note

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.
Disk Heads:	The number of read/write heads on this disk.
Cylinders:	The number of cylinders on this disk.
Sectors:	The total number of sectors on this disk.
Sector Size:	The size, in bytes, of each sector on this disk.

Field	Description
Total Size:	The total size of the disk, in sectors.

- Partition Properties Form

  To access the partition properties form, click the MB3 mouse button on desired partition icon.

  `Description:` This form provides detailed information on the attributes of a particular partition. The following table describes the fields in this form.

  Note

  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.
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.7 Disk Group Operations

This chapter provides information on menus and forms relating to disk group operations.

Note

With the Visual Administrator, partition icons represent partitions containing disk access records.

C.7.1 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

  `Description:` This operation defines a new disk group with a name specified by the user. The new disk group contains one or more LSM disks corresponding to the partitions selected by the user.

  `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

  `Description:` 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

  `Description:` 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

  `Description:` This operation adds a 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

  `Description:` 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

  `Description:` 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

  `Description:` 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.7.2 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 fields 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

  `Description:` This form is used to define a new disk group consisting of selected disks.

The following table describes the fields 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

  `Description:` This form is used to make the specified disk group available to the system. The following table describes the fields 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

  `Description:` 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

  `Description:` This form is used to add a LSM disk to a disk group. The following table describes the fields in this form.

Note

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.8 Projection Analysis

This section provides information on menus and forms relating to projection and analysis. In addition, tables are provided to summarize various aspects of projection and analysis behaviour. 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 rootdg. The Projection menu start or stop projection, as well as to highlight 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 rootdg. The Analyze menu can be used to start or stop analysis, as well as to set analysis-related preferences.

C.8.1 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.8.2 Projection Menus

The following list describes the menus, submenus, and menu selections you can access via the Projection menu:

- Icon Projection

  Projection → Icon Projection

  `Description:` This menu provides access to projection options used to start or stop projection for icons.

- Start

  Projection → Icon Projection → Start

  `Description:` 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

  `Description:` 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

  `Description:` This options stops projection for all icons that are currently undergoing selection.

- Show Free Subdisks

  Projection → Show Free Subdisks

  `Description:` 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.8.3 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.9 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.9.1 Analysis Menus

The following menu selections are accessed via the Analyze menu:

- Start

  Analyze → Start

  `Description:` 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

  `Description:` 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

  `Description:` 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

  `Description:` 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.9.2 Analysis Forms

The following forms are accessed via the Analyze menu:

- Analysis Parameters Form

  Analyze → Parameters

  `Description:` This form is used to set user preferences for conducting analysis. The following table describes the fields in this form.

  Note

  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 to be used for the statistics log. If the file does not already exist, it will be created. The filename is taken to be relative unless a pathname is given. To stop logging to the file, the filename text in this field must be erased. This field is optional. This log file is a binary file. In order to view the log file, `/usr/bin/lsmlog2text` *filename* must be run on this 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 the MB3 mouse button on desired icon that is being analyzed.
`Description:` 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.

Note

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.
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.9.3 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 Section 5.4.2 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

Index

E

F

G

H

I

J

K

L

M

N

O

Q

R

T

U

V

W

How to Order Additional Documentation

Technical Support

If you need help deciding which documentation best meets your needs, call 800-DIGITAL (800-344-4825) before placing your electronic, telephone, or direct mail order.

Electronic Orders

To place an order at the Electronic Store, dial 800-234-1998 using a 1200- or 2400-bps modem from anywhere in the USA, Canada, or Puerto Rico. If you need assistance using the Electronic Store, call 800-DIGITAL (800-344-4825).

Telephone and Direct Mail Orders

Your Location	Call	Contact
Continental USA, Alaska, or Hawaii	800-DIGITAL	Digital Equipment Corporation P.O. Box CS2008 Nashua, New Hampshire 03061
Puerto Rico	809-754-7575	Local Digital subsidiary
Canada	800-267-6215	Digital Equipment of Canada Attn: DECdirect Operations KAO2/2 P.O. Box 13000 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6
International	————	Local Digital subsidiary or approved distributor
Internal[a]	————	SSB Order Processing – NQO/V19 *or* U. S. Software Supply Business Digital Equipment Corporation 10 Cotton Road Nashua, NH 03063-1260

[a] For internal orders, you must submit an Internal Software Order Form (EN-01740-07).

Reader's Comments

DEC OSF/1
Logical Storage Manager
AA-Q3NCB-TE

Digital 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) 881-0120, Attn: UEG 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)	☐	☐	☐	☐
Completeness (enough information)	☐	☐	☐	☐
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 ______________________ Dept. __________
Company ______________________ Date __________
Mailing Address ______________________
______________ Email ______________ Phone __________

---- Do Not Tear - Fold Here and Tape ----

digital™

No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
UEG PUBLICATIONS MANAGER
ZKO3-3/Y32
110 SPIT BROOK ROAD
NASHUA, NH 03062-9987

--- Do Not Tear - Fold Here ---

Cut Along Dotted Line