VERITAS

VERITAS Volume Manager™
Command Line Interface
Administrator's Guide
Release 3.0.1

Contents

VERITAS Volume Manager Command Line Interface Administrator's Guide

Preface

The *VERITAS® Volume Manager Command Line Interface Administrator's Guide* describes the Command Line Interface (CLI) and the Support Operations interface (vxdiskadm).

Audience

This guide is intended for system administrators responsible for configuring and maintaining systems using the VERITAS Volume Manager.

This guide assumes:

• a basic understanding of system administration

• a working knowledge of the operating system being used

• experience with the window system being used

Scope

This guide describes how to perform a variety of Volume Manager tasks using the Volume Manager command line interface and the Volume Manager Support Operations menu interface (vxdiskadm). The chapters in this guide focus on frequently-used Volume Manager tasks.

You can use one or more of the following interfaces to perform Volume Manager tasks:

• command line interface—a set of Volume Manager commands (including an automated command—vxassist)

- a menu-driven interface (`vxdiskadm`)

- Volume Manager Storage Administrator—a graphical user interface

This guide describes how to use the command line interface and `vxassist` interfaces to perform similar Volume Manager tasks. The Storage Administrator interface is described in the *VERITAS Volume Manager Storage Administrator Administrator's Guide.*

Organization

This guide provides information on how to use the command line interface to the Volume Manager.

This guide is organized as follows:

Chapter 1, "The Command Line Interface," is an overview of the command line interface to Volume Manager.

Chapter 2, "Disk Tasks," describes how to use various commands to perform tasks on disks and disk groups. Topics include placing disks under Volume Manager control and adding disks to disk groups

Chapter 3, "Volume Tasks," describes how to use various commands to perform tasks on volumes. Topics include creating, mirroring, and resizing volumes.

Chapter 4, "Volume Administration Tasks," describes how to use commands to perform administration tasks on volumes.

Chapter 5, "Menu Interface Tasks," describes how to use the `vxdiskadm` (Volume Manager Task Operations) menu interface to perform tasks on disks and volumes.

Related Documents

The following documents provide related information:

- The *VERITAS Volume Manager Getting Started Guide* provides administrators with an overview of the Volume Manager and its features. It also provides general information about Volume Manager setup.

- The *VERITAS Volume Manager Storage Administrator Administrator's Guide* provides administrators with information on how to perform various Volume Manager tasks through the Storage Administrator graphical user interface (SA).

- The *VERITAS Volume Manager Administrator's Reference Guide* provides administrators with detailed information on Volume Manager recovery procedures and disk administration.

Conventions

The following table describes the typographic conventions used in this guide.

Typeface	Usage	Examples

courier	Computer output; user input; names of commands, files, and directories	`$You have mail.`
		The `cat` command displays files.
		`$ls -a`

italics	New terms; document titles; words to be emphasized; variables to be substituted with a real name or value	`$cat` *filename* Refer to the *User's Guide* for details.
bold	Glossary terms	

The Command Line Interface   1

Introduction

This chapter provides information about the VERITAS Volume Manager command line interface (CLI). The Volume Manager command set (for example, `vxassist`) ranges from commands requiring minimal user input to commands requiring detailed user input. Many of the Volume Manager commands require an understanding of Volume Manager concepts. Volume Manager concepts are described in the *VERITAS Volume Manager Getting Started Guide*. Most Volume Manager commands require superuser or access privileges.

Refer to the chapters that follow for information on using Volume Manager commands to perform common tasks. The `vxintro`(1M) manual page also contains introductory information relating to Volume Manager tasks.

Topics covered in this chapter include:

- Using the `vxassist` Command
- Displaying Disk Configuration Information
  - Displaying Disk Listings
  - Displaying Volume Manager Object Listings
  - Displaying Free Space in a Disk Group
- Creating Volumes
- Volume Manager Task Monitor Operations

> **Note:** Your system may use a *device name* that differs from the examples. See Chapter 1, "Understanding Volume Manager," in the *VERITAS Volume Manager Getting Started Guide,* for more information on device names.

Using the vxassist Command

You can use the vxassist command to create and manage volumes from the command line. vxassist requires minimal input from you and performs many underlying tasks for you. You can use vxassist to easily create, mirror, grow, shrink, remove, and back up volumes. vxassist is capable of performing many operations that would otherwise require the use of a series of more complicated Volume Manager commands. vxassist creates and manipulates volumes based on a set of established defaults, but also allows you to supply preferences for each task.

The vxassist command typically takes the following form:

    vxassist *keyword volume_name* [*attributes...*]

Select the specific action to perform by specifying an operation keyword as the first argument on the command line. For example, the keyword for creating a new volume is make. You can create a new volume by entering:

    vxassist make *volume_name length*

The first argument after any vxassist keyword is always a volume name. Follow the volume name with a set of attributes. Use these attributes to specify where to allocate space and whether you want mirroring or striping to be used.

You can select the disks on which the volumes are to be created by specifying the disk names at the end of the command line. For example, to create a 30 megabyte striped volume on three specific disks (disk03, disk04, and disk05), enter:

    vxassist make stripevol 30m layout=stripe disk03 disk04\
    disk05

vxassist defaults are listed in the file /etc/defaults/vxassist. The defaults listed in this file take effect if there is no overriding default specified on the vxassist command line.

Refer to the *VERITAS Volume Manager Administrator's Reference Guide,* or the
`vxassist`(1M) manual page for more information on the `vxassist`
command.

Displaying Disk Configuration Information

You can display disk configuration information from the command line.
Output listings available include: available disks, Volume Manager objects, and
free space in disk groups.

Displaying Disk Listings

To display a list of available disks, use this command:

```
vxdisk list
```

The output of `vxdisk list` lists the device name, the type of disk, the disk
name, the disk group to which the disk belongs, and status of the disk. This is
an example of `vxdisk list` output:

```
DEVICE      TYPE      DISK      GROUP      STATUS
c0t0d0s2    sliced    disk01    rootdg     online
c0t1d0s2    sliced    disk02    rootdg     online
c0t2d0s2    sliced    disk03    rootdg     online
```

Displaying Volume Manager Object Listings

The `vxprint` command displays detailed information on existing Volume
Manager objects. To display detailed output for all currently-existing objects
use this command:

```
vxprint -ht
```

Here is a typical output display:

```
Disk group: rootdg

DG NAME          NCONFIG      NLOG     MINORS    GROUP-ID
DM NAME          DEVICE       TYPE     PRIVLEN   PUBLEN   STATE
V  NAME          USETYPE      KSTATE   STATE     LENGTH   READPOL    PREFPLE
PL NAME          VOLUME       KSTATE   STATE     LENGTH   LAYOUT     NCOL/WI  MODE
SD NAM           PLEX         DISK     DISKOFFS  LENGTH   [COL/]OFF  DEVICE   MODE


dg rootdg        default      default  0                 774226267.1025.tweety

dm disk01        c0t0d0s2     sliced   1055      1054944  -
dm disk02        c0t1d0s2     sliced   1119      1043840  -

v  vol03         fsgen        DISABLED ACTIVE    1310720  SELECT-
pl vol03-01      vol03        DISABLED ACTIVE    1311632  STRIPE     2/128    RW
sd disk01-01     vol03-01     disk01   0         655380   0/0        c0t0d0   ENA
sd disk02-08     vol03-01     disk02   0         655760   1/0        c0t1d0   ENA

v  vm-build      fsgen        DISABLED ACTIVE    409600   SELECT     -
pl vm-build-01   vm-build     DISABLED ACTIVE    410016   STRIPE     2/128    RW
sd disk01-02     vm-build-01  disk01   655380    204831   0/0        c0t0d0   ENA
sd disk02-01     vm-build-01  disk02   655760    204960   1/0        c0t1d0   ENA
```

In this output, dg is a disk group, dm is a disk, v is a volume, pl is a plex (mirror), and sd is a subdisk. The top few lines indicate the headers that match each type of output line that follows. Each volume is listed along with its associated plex(es) and subdisk(s).

Displaying Free Space in a Disk Group

Before you add volumes and file systems to your system, make sure you have enough free disk space to meet your needs. You can use the Volume Manager to request a display of free space.

To display free space for a disk group, enter:

    vxdg -g *diskgroup* free

where -g *diskgroup* optionally specifies a disk group.

To see the free space in the default disk group, rootdg, enter:

    vxdg -g rootdg free

The Volume Manager displays the following:

    GROUP   DISK   DEVICE   TAG   OFFSET   LENGTH   FLAGS

    rootdg   disk04 c0t0d0s2   c0t0d0   726400   102672   -

    rootdg   disk01 c1t0d0s2   c1t0d0   0   102128   -

    rootdg   disk01 c1t0d0s2   c1t0d0   175856   26384   -

    rootdg   disk02 c1t1d0s2   c1t1d0   26624   175616   -

The free space is measured in 512-byte sectors.

Creating Volumes

Volumes are created to take advantage of the Volume Manager concept of virtual disks. Once a volume exists, a file system can be placed on the volume to organize the disk space with files and directories. Also, applications such as databases can be used to organize data on volumes.

You can create volumes using either a basic or advanced approach:

*   Basic—The basic approach takes information about what you want to accomplish and then performs the necessary underlying tasks. This approach requires only minimal input from you, but also permits more detailed specifications. Basic operations are performed primarily through the vxassist command.

- Advanced—The advanced approach consists of a number of commands that typically require you to specify detailed input. These commands use a "building block" approach that requires you to have a detailed knowledge of the underlying structure and components to manually perform the commands necessary to accomplish a certain task. Advanced operations are performed through several Volume Manager commands.

The creation of a volume involves the creation of plex and subdisk components. With the basic approach to volume creation, you indicate the desired volume characteristics and the underlying plexes and subdisks are created automatically.

Volumes can be created with various layout types:

- Concatenated—A volume whose subdisks are arranged both sequentially and contiguously within a plex. 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.

- Striped—A volume with data spread evenly across multiple disks. *Stripes* are equal-sized fragments that are allocated alternately and evenly to the subdisks of a single plex. There must be at least two subdisks in a striped plex, each of which must exist on a different disk. Throughput increases with the number of disks across which a plex is striped. Striping helps to balance I/O load in cases where high traffic areas exist on certain subdisks.

- RAID-5—A volume that uses striping to spread data and parity evenly across multiple disks in an array. Each stripe contains a parity stripe unit and data stripe units. Parity can be used to reconstruct data if one of the disks fails. In comparison to the performance of striped volumes, write throughput of RAID-5 volumes decreases since parity information needs to be updated each time data is accessed. However, in comparison to mirroring, the use of parity reduces the amount of space required.

- Mirrored—A volume with multiple plexes that duplicate the information contained in a volume. Although a volume can have a single plex, at least two are required for true mirroring (redundancy of data). Each of these plexes should contain disk space from different disks, for the redundancy to be useful.

- Striped and Mirrored—A volume with a striped plex and another plex that mirrors the striped one. This requires at least two disks for striping and one or more other disks for mirroring (depending on whether the plex is simple or striped). A striped and mirrored volume is advantageous because it both spreads data across multiple disks and provides redundancy of data.

- Mirrored and Striped—A volume with a plex that is mirrored and another plex that is striped. The mirrored and striped layout offers the benefits of spreading data across multiple disks (striping) while providing redundancy (mirroring) of data. The mirror and its striped plex are allocated from separate disks.

- Layered Volume—A volume built on top of volumes. Layered volumes can be constructed by mapping a subdisk to a VM disk or to a *storage* volume. A storage volume provides a recursive level of layout that is similar to the top-level volumes. Layered volumes allow for more combinations of logical layouts.

Volume Manager Task Monitor Operations

The Volume Manager Task Monitor tracks the progress of system recovery by monitoring task creation, maintenance, and completion. This section describes the -t command line option, lists the utilities that support the -t option, and describes the vxtask utility.

Task Monitor Options

Command line option (-t) allows you to specify a *task tag* for any task.

You can use the Task Monitor option with the following utilities:

vxplex, vxsd, vxvol, vxrecover, vxreattach, vxresize, vxassist, vxevac, and vxrootmir

Task Monitor option -t takes the following form:

utility [-t *tasktag*] ...

where:

utility   Volume Manager utilities that support the -t option.

tasktag   Assigns the given task tag to any tasks created by the utility. All tasks related to this operation are identified as a group.

For example, to execute a `vxrecover` command and track all resulting tasks as a group, you can specify a task tag to the `vxrecover` command as follows:

```
vxrecover -t myrecovery -b disk05
```

The `vxrecover` command creates a task to track all recovery jobs. To establish the task tag grouping, `vxrecover` specifies the tag to all utilities that it calls. Additionally, to establish a parent-child task relationship to any tasks those utilities execute, `vxrecover` passes its own task ID to those utilities.

For more information about the utilities, see their respective manual pages.

The `vxtask` Utility

The `vxtask` utility performs basic administrative operations on Volume Manager tasks that are running on the system. Operations include listing tasks, modifying the state of a task (pausing, resuming, aborting) and modifying the rate of progress of a task. See the `vxtask`(1M) manual page for more information.

Volume Manager *tasks* represent long-term operations in progress on the system. Every task gives information on the time the operation started, the size and progress of the operation, and the state and rate of progress of the operation. The administrator can change the state of a task, giving coarse-grained control over the progress of the operation. For those operations that support it, the rate of progress of the task can be changed, giving more fine-grained control over the task.

Every task is given a unique *task identifier*. This is a numeric identifier for the task that can be specified to the vxtask utility to specifically identify a single task. For most utilities, the tag is specified with the `-t tag` option.

Operations

The vxtask utility supports the following operations:

list   List tasks running on the system in one-line summaries. The `-l` option prints tasks in long format. The `-h` option prints tasks hierarchically, with child tasks following the parent tasks. By default,

all tasks running on the system are printed. If a `taskid` argument is supplied, the output is limited to those tasks whose `taskid` or task tag match `taskid`. The remaining arguments are used to filter tasks and limit the tasks actually listed.

monitor The monitor operation causes information about a task or group of tasks to be printed continuously as task information changes. This allows the administrator to track progress on an ongoing basis. Specifying -l causes a long listing to be printed. By default, short one-line listing are printed. In addition to printing task information when a task's state changes, output is also generated when the task completes. When this occurs, the state of the is printed as EXITED (see the OUTPUT section).

pause

resume

abort These three operations request that the specified task change its state. `pause` puts a running task in the paused state, causing it to suspend operation. `resume` causes a paused task to continue operation. `abort` causes the specified task to cease operation. In most cases, the operations "back out" as if an I/O error occurred, reversing to the extent possible what had been done so far.

set The set operation is used to change modifiable parameters of a task. Currently, there is only one modifiable parameter for tasks: the slow attribute, which represents a throttle on the task progress. The larger the *slow* value, the slower the progress of the task and the fewer system resources it consumes in a given time. (The slow attribute is the same attribute that many commands, such as vxplex(1m), vxvol(1m) and vxrecover(1m) accept on their command lines.)

Output

There are two output formats printed by vxtask: a short, one-line summary format per task, and a long task listing. The short listing provides the most used task information for quick perusal.

The long output format prints all available information for a task, spanning multiple lines. If more than one task is printed, the output for different tasks is separated by a single blank line. Each line in the long output format contains a title for the line, followed by a colon (:), followed by the information.

Examples

To list all tasks currently running on the system:

```
vxtask list
```

To trace all tasks in the diskgroup foodg that are currently paused, as well as any tasks with the tag sysstart:

```
vxtask -G foodg -p -i sysstart list
```

To list all tasks on the system that are currently paused:

```
vxtask -p list
```

To monitor all tasks with the tag myoperation:

```
vxtask monitor myoperation
```

To cause all tasks tagged with recovall to exit:

```
vxtask abort recovall
```

Disk Tasks 2 ≣

Introduction

This chapter provides instructions on performing the following disk and disk group tasks using the command line interface:

- Adding a Disk to Volume Manager
- Designating a Disk as a Hot-Relocation Spare
  - Removing a Disk as a Hot-Relocation Spare
- Renaming a Disk
- Reserving Disks
- Adding a Disk to a Disk Group
- Taking a Disk Offline
- Mirroring a Disk
- Removing a Disk
- Displaying Disk Information
  - Displaying Multipaths Under a VM Disk
  - Displaying Multipathing Information
- Creating a Disk Group
- Upgrading a Disk Group
  - Disk Group Upgrade Command
  - Creating Disk Groups of a Specific Version

- Renaming a Disk Group

- Destroying a Disk Group

- Moving a Disk Group

- Displaying Disk Group Information

Refer to the *VERITAS Volume Manager Administrator's Reference Guide,* for additional information and examples on using the command line interface to perform disk tasks. For more information about disks and disk groups, see Chapter 1, "The Command Line Interface."

Note: Your system may use a *device name* that differs from the examples. See Chapter 1, "Understanding Volume Manager," in the *VERITAS Volume Manager Getting Started Guide,* for more information on device names.

Adding a Disk to Volume Manager

You must place a disk under Volume Manager control, or add it to a disk group, before you can use the disk space for volumes. If the disk was previously in use, but not under Volume Manager control, you can preserve existing data on the disk while still letting the Volume Manager take control of the disk. This can be done by using the encapsulation feature of the Volume Manager. If the disk is new, it must be initialized. If the disk was previously not under Volume Manager control, but no data is required to be preserved, it should be initialized.

Add a disk by using this command:

    vxdiskadd *devname*

where *devname* is the device name of the disk to be added.

To add the device `c1t0d0` to Volume Manager control, do the following:

1. Enter the following to start `vxdiskadd`:

   vxdiskadd c1t0d0

   Notice that the `s2` suffix is not used in this example.

2. To continue with the task, enter `y` (or press Return) at the following prompt:

```
Add or initialize disks
Menu: VolumeManager/Disk/AddDisks

 Here is the disk selected.  Output format: [Device_Name]

   c1t0d0


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

3. At the following prompt, specify the disk group to which the disk should be added or press Return to accept rootdg: (See "Adding a Disk to a Disk Group" for more information.)

   ```
   You can choose to add this disk to an existing disk group, a
   new disk group, or leave the disk available for use by future
   add or replacement operations.  To create a new disk group,
   select a disk group name that does not yet exist.  To leave the
   disk available for future use, specify a disk group name of
   "none".
   Which disk group [<group>,none,list,q,?] (default: rootdg)
   ```

4. At the following prompt, either press Return to accept the default disk name or enter a disk name:

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

5. When prompted as to whether this disk should become a hot-relocation spare, enter n (or press Return): (See "Designating a Disk as a Hot-Relocation Spare" for more information.)

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

6. To continue with the task, enter y (or press Return) at the following prompt:

   ```
   The selected disks will be added to the disk group rootdg with
   default disk names.

   c1t0d0

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

7. If there is data on this disk that needs to be preserved, enter y to select encapsulation:

   ```
   The following disk device has a valid VTOC, but does not appear
   ```

> to have been initialized for the Volume Manager. If there is
> data on the disk that should NOT be destroyed you should
> encapsulate the existing disk partitions as volumes instead of
> adding the disk as a new disk Output format: [Device_Name]
>
> c1t0d0

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

8. To continue the task, press Return at the following prompt:

> The following disk has been selected for encapsulation.
> Output format: [Device_Name c1t0d0]

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

A message similar to the following indicates that the disk is being
encapsulated for Volume Manager use:

> The disk device c1t0d0 will be encapsulated and added to the
> disk group rootdg with the disk name disk01.
>
> The c1t0d0 disk has been configured for encapsulation.
>
> The first stage of encapsulation has completed successfully.
> You should now reboot your system at the earliest possible
> opportunity.
>
> The encapsulation will require two or three reboots which will
> happen automatically after the next reboot. To reboot execute
> the command:

shutdown -g0 -y -i6

> This will update the /etc/vfstab file so that volume devices
> are used to mount the file systems on this disk device. You
> will need to update any other references such as backup scripts,
> databases, or manually created swap devices.
>
> Goodbye.

Remember to perform a shutdown and reboot as soon as convenient.

Designating a Disk as a Hot-Relocation Spare

Hot-relocation allows a system to automatically react to I/O failure by relocating redundant subdisks to other disks and restoring the affected Volume Manager objects and data. If a disk has already been designated as a spare in the disk group, the subdisks from the failed disk are relocated to the spare disk. Otherwise, any suitable free space in the disk group is used. Refer to Chapter 2, in the *VERITAS Volume Manager Getting Started Guide*, for more information.

To designate a disk as a hot-relocation spare, use this command:

    vxedit set spare=on *diskname*

For example, to designate disk disk01 as a spare, use this command:

    vxedit set spare=on disk01

You can use the vxdisk list command to confirm that this disk is now a spare; disk01 should be listed with a spare flag.

Any VM disk in this disk group can now use this disk as a spare in the event of a failure. If a disk fails, hot-relocation automatically occurs (if possible). You are notified of the failure and relocation through electronic mail. After successful relocation, you may want to replace the failed disk.

Removing a Disk as a Hot-Relocation Spare

While a disk is designated as a spare, the space on that disk is not used as free space for the creation of Volume Manager objects within its disk group. If necessary, you can free a spare disk for general use by removing it from the pool of hot-relocation disks.

To determine which disks are currently designated as spares, use this command:

    vxdisk list

The output of this command lists any spare disks with the spare flag.

To remove a spare from the hot-relocation pool, enter:

    vxedit set spare=off *diskname*

For example, to make disk `disk01` available for normal use, enter:

```
vxedit set spare=off disk01
```

Renaming a Disk

If you do not specify a Volume Manager name for a disk, the Volume Manager gives the disk a default name when you add the disk to Volume Manager control. The Volume Manager name is used by the Volume Manager to identify the location of the disk or the disk type. You can change the disk name to reflect a change of ownership or use, enter:

```
vxedit rename old_diskname new_diskname
```

To rename `disk01` to `disk03`, enter:

```
vxedit rename disk01 disk03
```

To confirm that the name change took place, enter:

```
vxdisk list
```

The Volume Manager returns the following:

```
DEVICE     TYPE    DISK    GROUP    STATUS

c0t0d0s2  sliced  disk04  rootdg   online

c1t0d0s2  sliced  disk03  rootdg   online

c1t1d0s2  sliced  -       -        online
```

Note: By default, Volume Manager names subdisk objects after the VM disk on which they are located. Renaming a VM disk does not automatically rename the subdisks on that disk.

Reserving Disks

By default, `vxassist` allocates space from any disk that has free space. You can reserve a set of disks for special purposes, such as to avoid general use of a particularly slow or a particularly fast disk.

Use this command to reserve a disk for special purposes:

```
vxedit set reserve=on diskname
```

After you enter this command, `vxassist` does not allocate space from the selected disk unless that disk is specifically mentioned on the `vxassist` command line. For example, if disk `disk03` is reserved, the command:

```
vxassist make vol03 20m disk03
```

overrides the reservation and creates a 20 megabyte volume on `disk03`. However, the command:

```
vxassist make vol04 20m
```

does not use `disk03`, even if there is no free space on any other disk.

To turn off reservation of a disk, enter:

```
vxedit set reserve=no diskname
```

Adding a Disk to a Disk Group

You can add a new disk to an already established disk group. For example, the current disks have insufficient space for the application or work group requirements, especially if these requirements have changed.

To add an initialized disk to a disk group, use this command:

```
vxdiskadd devname
```

To add device `c1t1d0` to `rootdg`, use the following steps:

1. Enter this command to start `vxdiskadd`:

   ```
   vxdiskadd c1t1d0
   ```

   `vxdiskadd` displays the following message:

   ```
   Add or initialize disks
   Menu: VolumeManager/Disk/AddDisks

   Here is the disk selected.  Output format: [Device_Name]

     c1t1d0

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

2. At the following prompt, specify the disk group to which the disk should be added or press Return to accept `rootdg`:

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

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

3. At the following prompt, either press Return to accept the default disk name or enter a disk name:

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

4. When `vxdiskadd` asks whether this disk should become a hot-relocation spare, enter `n` (or press Return):

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

5. To continue with the task, enter `y` (or press Return) at the following prompt:

   ```
   The selected disks will be added to the disk group rootdg with
   default disk names.
   ```

   ```
   c1t1d0
   ```

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

6. The following prompt indicates that this disk has been previously initialized for future Volume Manager use; enter `y` to confirm that you now want to use this disk:

   ```
   The following disk device appears to have been initialized
   already. The disk is currently available as a replacement disk.
   Output format: [Device_Name]
   ```

   ```
   c1t1d0
   ```

   ```
   Use this device? [y,n,q,?] (default: y) y
   ```

7. To reinitialize the disk, enter `y` (or press Return) at the following prompt:

   ```
   The following disk you selected for use appears to already have
   been initialized for the Volume Manager.  If you are certain
   the disk has already been initialized for the Volume Manager,
   ```

```
    then you do not need to reinitialize the disk device.
    Output format: [Device_Name]

    c1t1d0
```

```
Reinitialize this device? [y,n,q,?] (default: y) y
```

Messages similar to the following now confirm that this disk is being reinitialized for Volume Manager use. You may also be given the option of performing surface analysis on some systems.

```
    Initializing device c1t1d0.
```
```
Perform surface analysis (highly recommended)
    [y,n,q,?] (default: y) n
    Adding disk device c1t1d0 to disk group rootdg with disk name
    disk03.
```

To confirm that the disk has been added to the disk group, enter:

```
    vxdisk list
```

The Volume Manager returns a listing similar to the following:

```
    DEVICE     TYPE    DISK    GROUP    STATUS

    c0t0d0s2  sliced  disk04  rootdg   online

    c1t0d0s2  sliced  disk01  rootdg   online

    c1t1d0s2  sliced  disk03  rootdg   online
```

Taking a Disk Offline

You may need to take a physical disk offline. 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 take a physical disk offline, first remove the disk from its disk group (See "Removing a Disk" for more information). Then place the disk in an "offline" state with this command:

```
    vxdisk offline devname
```

To take the device c1t1d0s2 off line, use this command:

```
vxdisk offline c1t1d0s2
```

Note: The device name is used here because the disk is no longer in a disk group and so does not have an administrative name.

Mirroring a Disk

Mirroring the root disk mirrors the root volume and other areas needed for booting onto another disk.

To mirror your boot (root) disk onto another disk, use either the steps outlined here or `vxdiskadm` (see Chapter 5, "Menu Interface Tasks"). This makes it possible to recover from failure of your boot disk by replacing it with the mirror of the boot disk.

Before mirroring your boot disk, the EEPROM variable `use-nvramrc?` must be set to `true` if you want to take advantage of the Volume Manager boot disk aliases to identify the mirror of the boot disk if a replacement is needed. If this variable is set to `false`, you will need to determine which disks are bootable yourself. You can set this variable to `true` as follows:

eeprom use-nvramrc?=true

To mirror your boot disk, do the following:

1. Select a disk that is at least as large as your boot disk.

2. Use the `vxdiskadd` command to add the selected disk as a new disk (if it is not already added).

3. Execute the following command:

    /etc/vx/bin/vxrootmir *alternate_disk*

    where *alternate_disk* is the disk name assigned to the other disk.

    `vxrootmir` creates a mirror for `rootvol` (the volume for the root file system on an alternate disk).

    The alternate boot disk is configured to enable booting from it if the primary boot disk fails.

There may be other volumes on the boot disk, such as volumes for `/home` or `/tmp` file systems. These can be mirrored separately using the `vxassist` utility. For example, if you have a `/home` file system on a volume `homevol`, you can mirror it to *alternate_disk* using the command:

```
vxassist mirror homevol  alternate_disk
```

If you do not have space for a copy of some of these file systems on your alternate boot disk, you can mirror them to other disks. You can also span or stripe these other volumes across other disks attached to your system.

To list all volumes on your primary boot disk, use the command:

```
vxprint -t -v -e\
'aslist.aslist.sd_disk="boot_disk"'
```

To mirror all of the concatenated volumes on this disk to your alternate boot disk, use the command:

```
/etc/vx/bin/vxmirror  boot_disk  alternate_disk
```

Removing a Disk

You can remove a disk from a system and move it to another system if the disk is failing or has failed. Before removing the disk from the current system, you must:

1. Unmount any file systems on the volumes.

2. Stop the volumes on the disk.

3. Move the volumes to other disks or back up the volumes. To move a volume, mirror the volume on one or more other disks, then remove the original copy of the volume. If the volumes are no longer needed, they can be removed instead of moved.

to remove a disk, use these steps:

1. Remove the disk from its disk group with this command:

```
vxdg [-g groupname] rmdisk  diskname
```

where *groupname* is the name of the group to which the disk belongs and *diskname* is the name of the disk to be removed.

For example, to remove `disk01` from `rootdg`, enter:

```
vxdg rmdisk disk01
```

Since `rootdg` is the default disk group, you do not need to specify it.

2. Remove the disk from Volume Manager control with this command:

```
vxdisk rm devname
```

For example, to remove `c1t0d0` from Volume Manager control, enter:

```
vxdisk rm c1t0d0s2
```

Displaying Disk Information

Before you use a disk, you need to know if it has been initialized and placed under Volume Manager control. You also need to know if the disk is part of a disk group, because you cannot create volumes on a disk that is not part of a disk group. The `vxdisk 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.

You can display information on all disks that are defined to the Volume Manager with this command:

```
vxdisk list
```

The Volume Manager returns the following display:

```
DEVICE    TYPE    DISK    GROUP    STATUS

c0t0d0s2  sliced  disk04  rootdg   online

c1t0d0s2  sliced  disk01  rootdg   online

c1t1d0s2  sliced  -       -        online
```

To display details on a particular disk defined to the Volume Manager, enter:

```
vxdisk list disk01
```

Displaying Multipaths Under a VM Disk

The vxdisk utility is used to display the multipathing information for a particular metadevice. The metadevice is a device representation of a particular physical disk having multiple physical paths from the I/O controller of the system. In the Volume Manager, all the physical disks in the system are represented as metadevices with one or more physical paths.

You can see the multipathing information for a particular metadevice with the command:

```
vxdisk list c1t5d0s2
```

c1t5d0s2 is the metadevice formed by the DMP subsystem of Volume Manager.

Displaying Multipathing Information

You can display details on a particular disk defined to the Volume Manager with this command:

```
vxdisk list disk01
```

The Volume Manager returns the following display:

```
Device     c2t0d0s2
devicetag  c2t0d0
type       sliced
hostid     aparajita
disk       name=disk01 id=861086917.1052.aparajita
group      name=rootdg id=861086912.1025.aparajita
flags      online ready autoconfig autoimport imported
pubpaths   block=/dev/vx/dmp/c2t0d0s4 char=/dev/vx/rdmp/c2t0d0s4
privpaths  block=/dev/vx/dmp/c2t0d0s3 char=/dev/vx/rdmp/c2t0d0s3
version    2.1
iosize     min=512 (bytes) max=2048 (blocks)
public     slice=4 offset=0 len=1043840
private    slice=3 offset=1 len=1119
update     time=861801175 seqno=0.48
headers    0 248
configs    count=1 len=795
```

```
logs        count=1 len=120
Defined regions
config      priv 000017-000247[000231]:copy=01 offset=000000 enabled
config      priv 000249-000812[000564]:copy=01 offset=000231 enabled
log         priv 000813-000932[000120]:copy=01 offset=000000 enabled
Multipathing information:
numpaths:        2
c2t0d0s2         state=enabled        type=primary
c1t0d0s2         state=disabled       type=secondary
```

The output shows two paths to a physical device represented by the metadevice c2t0d0s2. The path c2t0d0s2 is active (state=enabled) and the other path c1t0d0s2 is in a failed state (state=disabled).

Additional information in the form of *type* is shown for disks on active/passive type disk arrays. This information indicates the *primary* and *secondary* paths to the disk. For example, DG Clariion, Hitachi DF350, etc.

This *type* information will not be present for disks on active/active type disk arrays because there is no concept of primary and secondary paths to disks on these disk arrays. For example, StorEdge A5000 and Sparc Storage Array (SSA) disk arrays.

The vxdisk list command returns the following display for disk01:

```
Device     c2t0d0s2
devicetag  c2t0d0
type       sliced
hostid     aparajita
disk       name=disk01 id=861086917.1052.aparajita
group      name=rootdg id=861086912.1025.aparajita
flags      online ready autoconfig autoimport imported
pubpaths   block=/dev/vx/dmp/c2t0d0s4 char=/dev/vx/rdmp/c2t0d0s4
privpaths  block=/dev/vx/dmp/c2t0d0s3 char=/dev/vx/rdmp/c2t0d0s3
version    2.1
iosize     min=512 (bytes) max=2048 (blocks)
public     slice=4 offset=0 len=1043840
private    slice=3 offset=1 len=1119
update     time=861801175 seqno=0.48
headers    0 248
```

```
configs      count=1 len=795
logs         count=1 len=120
Defined regions
config      priv 000017-000247[000231]:copy=01 offset=000000 enabled
config      priv 000249-000812[000564]:copy=01 offset=000231 enabled
log         priv 000813-000932[000120]:copy=01 offset=000000 enabled
Multipathing information:
numpaths:        3
c1t0d0s2         state=enabled
c2t0d0s2         state=enabled
c3t0d0s2         state=disabled
```

Creating a Disk Group

Disks need to be in disk groups before they Volume Manager can use the disks for volumes. Volume Manager always has the root disk group, but you can add more disk groups if needed.

Note: The Volume Manager supports a default disk group, rootdg, in which all volumes are created if no further specification is given. All commands default to rootdg as well.

A disk group must always contain at least one disk. A disk that does not belong to any disk group must be available when you create a disk group.

Data related to a particular set of applications or a particular group of users may need to be made accessible on another system. Examples of this are:

• A system has failed and its data needs to be moved to other systems.

• The work load must be balanced across a number of systems.

It is important that the data related to particular application(s) or users be located on an identifiable set of disk, so that when these disks are moved, all data of the application(s) or group of users, and no other data, is moved.

To create the disk group newdg associated with a disk, follow these steps:

1. Enter the following command to start vxdiskadd:

    vxdiskadd c1t1d0

2. At the following prompt, press Return to continue:

```
Add or initialize disks
Menu: VolumeManager/Disk/AddDisks


Here is the disk selected.  Output format: [Device_Name]


   c1t1d0


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

3. At the following prompt, specify the disk group to which the disk should be added (`newdg`, in this case):

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

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

4. When `vxdiskadd` confirms that no active disk group currently exists with the same name and prompts for confirmation that you want to create this new disk group. Enter `y` to continue:

```
There is no active disk group named newdg.
```

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

5. At the following prompt, either press Return to accept the default disk name or enter a disk name:

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

6. When `vxdiskadd` asks whether this disk should become a hot-relocation spare, enter n (or press Return):

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

7. To continue with the task, enter `y` (or press Return) at the following prompt:

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

```
    c1t1d0
```

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

Messages similar to the following should now confirm that this disk is being initialized for Volume Manager use:

```
    Initializing device c1t1d0.


    Creating a new disk group named newdg containing the disk device
    c1t1d0 with the name newdg01.
```

8. To verify that the disk group was created, use this command:

```
    vxdisk list
```

The Volume Manager returns the following display:

```
    DEVICE    TYPE    DISK      GROUP    STATUS

    c0t0d0s2  sliced  disk04    rootdg   online

    c1t0d0s2  sliced  disk03    rootdg   online

    c1t1d0s2  sliced  newdg01   newdg    online
```

Upgrading a Disk Group

Prior to the release of Volume Manager 3.0, the disk group version was automatically upgraded (if needed) when the disk group was imported. Upgrading the disk group makes it incompatible with previous Volume Manager releases.

The Volume Manager 3.0 disk group upgrade feature separates the two operations of importing a disk group and upgrading its version. You can import a disk group of down-level version and use it without upgrading it.

When you want to use new features, the disk group can be upgraded. The upgrade is an explicit operation (as opposed to earlier VxVM versions, where the upgrade was performed when the disk group was imported). Once the upgrade occurs, the disk group becomes incompatible with earlier releases of VxVM that do not support the new version.

Before the imported disk group is upgraded, no changes are made to the disk group to prevent its use on the release that it was imported from until the administrator explicitly upgrades it to the current release.

Until completion of the upgrade, the disk group can be used "as is" provided there is no attempt to use the features of the current version. Attempts to use a feature of the current version that is not a feature of the version the disk group was imported from results in an error message similar to this:

```
vxvm:vxedit: ERROR:  Disk group version doesn't support feature
```

To use any of the new features, the administrator needs to execute a command to explicitly upgrade the disk group to a version that supports those features.

All disk groups have a version number associated with them. Volume Manager releases support a specific set of disk group versions. Volume Manager can import and perform operations on a disk group of that version. The operations are limited by what features and operations the disk group version supports.

Table 2-1 summarizes the VxVM releases that introduce and support specific disk group versions:

Table 2-1: Disk Group Version Assignments

VxVM Release	Introduces Version	Supports Versions
1.2	10	10
1.3	15	15
2.0	20	20
2.2	30	30
2.3	40	40
2.5	50	50
3.0	60	20-60

Note: With the exception of VxVM 3.0, all VxVM releases perform the upgrade of supported disk group versions *when* they are imported.

Importing the disk group of a previous version on a VxVM 3.0 system prevents the use of features introduced since that version was released. Table 2-2 summarizes features not supported by specific disk group versions.

Table 2-2: Disk Group Features Not Supported

Disk Group Version	Features Not Supported
50	Online Relayout, Safe RAID-5 Subdisk Moves
40	
30	Hot Relocation
20	VxSmartSync Recovery Accelerator
10, 15	RAID-5 Volumes, New-style Stripes, Recovery Checkpointing, Diskgroup Configuration Copy Limiting, Dirty Region Logging, Mirrored Volumes Logging

Disk Group Upgrade Command

This section describes the disk upgrade command `vxdg upgrade`.

Listing the Disk Group Version

You can get a disk group version listing by specifying a disk group name with this command:

    vxdg list *dgname*

You can determine the disk group version by using the `vxprint`(1M) command with the `-l` format option.

Creating Disk Groups of a Specific Version

By default, VxVM creates a disk group of the highest version supported by the release of VxVM. For example, VxVM 3.0 creates disk groups of Version 60.

It may be necessary to create a disk group of a back-level version. A disk group created in default fashion on a system running VxVM Release 3.0 would be of disk group version 60. It would not be importable on a system running VxVM Release 2.5, which only supports up to version 50. Therefore, to create a disk

group on a system running VxVM Release 3.0 that can be imported by a system running VxVM Release 2.5, the disk group must be created with a version of 50 or less.

To create such a disk group, the -T <*version*> option can be specified to the `vxdg init` command. Option <*version*> should be the disk group version desired for the disk group. For example, to create a disk group that can be imported by a system running VxVM 2.5, the command:

```
% vxdg -T 40 init newdg newdg01=c0t3d0s2
```

creates the disk group `newdg` with a disk group version of 40. This disk group can be imported by VxVM Release 2.5. Note that while the disk group can be imported on the VxVM 2.5 system, attempts to use VxVM 3.0 features fail with the error message:

```
vxvm:vxedit: ERROR:  Disk group version doesn't support feature
```

Upgrading a Disk Group

VxVM upgrades the disk group to the highest version supported by the release of VxVM that is currently running. For VxVM Release 3.0, the disk group is upgraded to version 60 with this command:

```
vxdg upgrade dgname
```

Renaming a Disk Group

Because only one disk group of a given name can exist per system, you must rename a disk group if you want to move it (import it) to a system already containing a disk group with the same name. A special case of this is with `rootdg`.

Every system running the Volume Manager must have a single `rootdg` disk group. `rootdg` can only be moved across systems if it is renamed.

You can use the following steps to temporarily move the `rootdg` disk group from one host to another (for repair work on the root volume, for example) and then move it back:

1. On the original host, identify the disk group ID of the root disk group to be imported with this command:

```
vxdisk -s list
```

This command results in output that includes the following.

```
dgname: rootdg
dgid:   774226267.1025.tweety
```

2. On the importing host, import and rename the `rootdg` disk group with this command:

   vxdg -tC -n *newdg_name* import *diskgroup*

   where `-t` indicates a temporary import name; `-C` clears import locks (and should only be used if you are *certain* that the disks are not being used by any other hosts); `-n` specifies a temporary name for the `rootdg` to be imported (so that it does not conflict with the existing `rootdg`); and *diskgroup* is the ID of the `rootdg` disk group being imported.

   If a crash or reboot occurs at this point, the temporarily-imported disk group becomes unimported and requires a reimport.

3. After the necessary work has been done on the imported `rootdg`, deport it back to its original host with this command:

   vxdg -h *hostname* deport *diskgroup*

where *hostname* is the name of the system whose `rootdg` is being returned (the name of the system can be confirmed with the command `uname -n`). This command removes the imported `rootdg` from the importing host and returns locks to its original host. The original host then autoimports its `rootdg` on its next reboot.

Destroying a Disk Group

The `vxdg` command provides a destroy option that removes a disk group from the system and frees the disks in that disk group for use in other disk groups. Disk groups that are not needed should be removed with the `vxdg destroy` command so that the disks can be used by other disk groups.

vxdg destroy *diskgroup*

The `vxdg deport` command can still be used to make disks inaccessible. The Volume Manager prevents disks in a deported disk group from being used in other disk groups.

Moving a Disk Group

A disk group can be moved between systems, along with its Volume Manager objects (except for `rootdg`). This relocates the disk group configuration to a new system.

Move a disk group across systems as follows:

1. Unmount and stop all volumes in the disk group on the first system.

2. Deport (disable local access to) the disk group to be moved with this command:

   ```
   vxdg deport diskgroup
   ```

3. Import (enable local access to) the disk group and its disks from the second system with this command:

   ```
   vxdg import diskgroup
   ```

4. After the disk group is imported, start all volumes in the disk group with this command:

   ```
   vxrecover -g diskgroup -sb
   ```

Displaying Disk Group Information

To use disk groups, you need to know their names and what disks belong to each group.

Display information on existing disk groups with this command:

```
vxdg list
```

The Volume Manager returns the following listing of current disk groups:

```
NAME      STATE      ID
rootdg    enabled    730344554.1025.tweety
newdg     enabled    731118794.1213.tweety
```

You can display more detailed information on a specific disk group (such as `rootdg`) with this command:

```
vxdg list rootdg
```

The Volume Manager displays the following example output for `rootdg`:

```
Group:      rootdg
dgid:       730344554.1025.tweety
import-id: 0.1
flags:
version 60
copies:     nconfig=default nlog=default
config:     seqno=0.94553 permlen=795 free=768 templen=16 loglen=8
config disk c2t2d2s2 copy 1 len=795 disabled
config disk c2t2d3s2 copy 1 len=795 disabled
config disk c2t2d4s2 copy 1 len=795 disabled
config disk c2t3d0s2 copy 1 len=795 state=clean online
log disk c2t2d2s2 copy 1 len=120 disabled
log disk c2t2d3s2 copy 1 len=120 disabled
log disk c2t2d4s2 copy 1 len=120 disabled
log disk c2t3d0s2 copy 1 len=120
```

If you need to verify the disk group ID and name associated with a specific disk (to import the disk group, for example), use this command:

```
vxdisk -s list devicename
```

This command provides output that includes the following information for the specified disk:

```
dgname: rootdg
dgid:   774226267.1025.tweety
```

2

VERITAS Volume Manager Command Line Interface Administrator's Guide

Volume Tasks | 3

Introduction

This chapter describes how to create and maintain a system configuration under Volume Manager control. It includes information about creating, removing, and maintaining Volume Manager objects (volumes, plexes, and subdisks). Online backup information is also included.

Information is presented as overviews of commands and more detailed descriptions of tasks using the commands.

This chapter provides instructions on performing the following volume tasks using the command line interface:

- Creating a Volume
  - Creating a Concatenated Volume
  - Creating a Striped Volume
  - Creating a RAID-5 Volume
  - Creating a Mirrored Volume
- Resizing a Volume
  - Resizing Volumes With `vxassist`
  - Resizing Volumes With `vxvol`
  - Resizing Volumes With `vxresize`
- Removing a Volume
- Renaming and Removing Volume Manager Objects

- Mirroring a Volume
  - Creating a Volume with Dirty Region Logging Enabled
  - Mirroring an Existing Volume
  - Mirroring All Volumes
- Adding a RAID-5 Log
- Adding a DRL Log
- Removing a DRL Log
- Removing a RAID-5 Log
- Stopping a Volume
- Starting a Volume
  - Listing Unstartable Volumes
- Preparing a Volume to Restore From Backup
- Recovering a Volume
- Removing a Mirror
- Moving Subdisks
- Splitting Subdisks
- Joining Subdisks
- Displaying Volume Configuration Information
- Performing Online Backup

Note: Some Volume Manager commands require superuser or other appropriate privileges.

Creating a Volume

Volumes are created to take advantage of the Volume Manager concept of virtual disks. A file system can be placed on the volume to organize the disk space with files and directories. Also, applications such as databases can be used to organize data on volumes.

The Volume Manager allows you to create volumes with the following layout types:

- Concatenated

- Striped

- RAID-5

- Mirroring

The vxassist command provides the simplest way to create new volumes.

You can create a volume by using vxassist. The command takes the following form:

    vxassist make *volume_name* *length* [*attributes*]

where make is the keyword for volume creation, *volume_name* is a name you give to the volume, and *length* specifies the number of sectors (by default) in the volume. The length can be specified in kilobytes, megabytes, or gigabytes by using a suffix character of k, m, or g, respectively. Refer to the vxintro(1M) manual page for more information on specifying units of length when creating volumes. Additional attributes can be specified, as appropriate.

By default, vxassist creates volumes in the rootdg disk group. Another disk group can be specified by including -g *diskgroup* in the vxassist command line.

Creating a volume in the disk group rootdg creates two device node files that can be used to access the volume:

- /dev/vx/dsk/*volume_name* (the block device node for the volume)

- /dev/vx/rdsk/*volume_name* (the raw device node for the volume)

For volumes in rootdg and disk groups other than rootdg, these names include the disk group name, as follows:

- /dev/vx/dsk/*diskgroup_name/volume_name*

- /dev/vx/rdsk/*diskgroup_name/volume_name*

"Creating a Concatenated Volume," describes the simplest way to create a (default) volume. Later sections describe how to create volumes with specific attributes.

Creating a Concatenated Volume

By default, vxassist creates a concatenated volume that uses 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.

If there is not enough space on a single disk, vxassist creates a 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.

Creating a Concatenated Volume on Any Disk

If no disk is specified, the Volume Manager selects a disk on which to create the volume.

Create a concatenated, default volume with this command:

```
vxassist make volume_name length
```

where *volume_name* is the name of the volume and *length* specifies the length of the volume in sectors (unless another unit of size is specified with a suffix character).

When you resize a volume, you can specify the length of a new volume in sectors, kilobytes, megabytes, or gigabytes. The unit of measure is added as a suffix to the length (s, m, k, or g). If no unit is specified, sectors are assumed.

You can use this command to create the volume voldefault with a length of 10 megabytes:

```
vxassist make voldefault 10m
```

Creating a Concatenated Volume on a Specific Disk

The Volume Manager automatically selects the disk(s) each volume resides on, unless you specify otherwise. If you want a volume to reside on a specific disk, you must designate that disk for the Volume Manager. More than one disk can be specified.

Create a volume on a specific disk with this command:

```
vxassist make volume_name length diskname [ ... ]
```

To create the volume `volspecific` on `disk03`, use this command:

```
vxassist make volspecific 3m disk03
```

Creating a Striped Volume

A striped volume contains at least one plex that consists of two or more subdisks located on two or more physical disks. For more information on striping, refer to Chapter 1 of the *VERITAS Volume Manager Getting Started Guide.*

Create a striped volume with this command:

```
vxassist make volume_name length layout=stripe
```

To create the striped volume `volzebra`, use this command:

```
vxassist make volzebra 10m layout=stripe
```

This creates a striped volume with the default stripe unit size on the default number of disks.

You can indicate the disks on which the volumes are to be created by specifying the disk names at the end of the command line. For example, to create a 30 megabyte striped volume on three specific disks (`disk03`, `disk04`, and `disk05`), use this command:

```
vxassist make stripevol 30m layout=stripe disk03 disk04\

disk05
```

Creating a RAID-5 Volume

A RAID-5 volume contains a RAID-5 plex that consists of two or more subdisks located on two or more physical disks. Only one RAID-5 plex can exist per volume. A RAID-5 volume can also contain one or more RAID-5 log plexes, which are used to log information about data and parity being written to the volume. For more information on RAID-5 volumes, refer to the *VERITAS Volume Manager Getting Started Guide.*

Create a RAID-5 volume with this command:

```
vxassist make volume_name length layout=raid5
```

To create the RAID-5 volume `volraid`, use this command:

```
vxassist make volraid 10m layout=raid5
```

This creates a RAID-5 volume with the default stripe unit size on the default number of disks. It also creates a RAID-5 log by default.

Creating a Mirrored Volume

Create a new mirrored volume with this command:

```
vxassist make volume_name length layout=mirror
```

To create the mirrored volume, `volmir`, use this command:

```
vxassist make volmir 5m layout=mirror
```

Resizing a Volume

Resizing a volume changes the volume size. This can be done by using either `vxassist`, `vxvol`, or `vxresize`.

If the volume is not large enough for the amount of data that needs to be stored in it, you need to extend the length of the volume. If a volume is increased in size, `vxassist` automatically finds available disk space.

When you resize a volume, you can specify the length of a new volume in sectors, kilobytes, megabytes, or gigabytes. The unit of measure is added as a suffix to the length (`s`, `m`, `k`, or `g`). If no unit is specified, sectors are assumed.

CAUTION! Do not shrink a volume below the size of the file system. If you have a VxFS file system, you can shrink the file system and then shrink the volume. If you do not shrink the file system first, you risk unrecoverable data loss.

Resizing Volumes With `vxassist`

`vxassist` can resize a volume in 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

Extending to a Given Length

Extend a volume *to* a specific length with this command:

    `vxassist growto` *volume_name length*

To extend `volcat` to 2000 sectors, use this command:

    `vxassist growto volcat 2000`

Extending by a Given Length

Extend a volume *by* a specific length with this command:

    `vxassist growby` *volume_name length*

To extend `volcat` by 100 sectors, use this command:

    `vxassist growby volcat 100`

Shrinking to a Given Length

Shrink a volume *to* a specific length with this command:

    `vxassist shrinkto` *volume_name length*

Make sure you do not shrink the volume below the current size of the file system or database using the volume. This command can be safely used on empty volumes.

To shrink `volcat` to 1300 sectors, use this command:

    `vxassist shrinkto volcat 1300`

Shrinking by a Given Length

Shrink a volume *by* a specific length with this command:

    `vxassist shrinkby` *volume_name length*

To shrink `volcat` by 300 sectors, use this command:

```
vxassist shrinkby volcat 300
```

Resizing Volumes With `vxvol`

To change the length of a volume using `vxvol set`, use this command:

`vxvol set len=`*value* `...` *volume_name* `...`

For example, to change the length to 100000 sectors, use this command:

`vxvol set len=100000 vol01`

Note: The `vxvol set len` command cannot increase the size of a volume unless the needed space is available in the plexes of the volume. When the size of a volume is reduced using the `vxvol set len` command, the freed space is not released into the free space pool.

Changing the Volume Read Policy

Volume Manager offers the choice of the following read policies:

- `round` reads each plex in turn in "round-robin" fashion for each nonsequential I/O detected. Sequential access causes only one plex to be accessed. This takes advantage of the drive or controller read-ahead caching policies.

- `prefer` reads first from a plex that has been named as the preferred plex.

- `select` chooses a default policy based on plex associations to the volume. If the volume has an enabled striped plex, `select` defaults to preferring that plex; otherwise, it defaults to round-robin.

The read policy can be changed from `round` to `prefer` (or the reverse), or to a different preferred plex. The `vxvol rdpol` command sets the read policy for a volume.

To set the read policy to `round`, use this command:

`vxvol rdpol round` *volume_name*

For example, the command to set the read policy for volume `vol01` to a round-robin read is:

```
vxvol rdpol round vol01
```

To set the read policy to `prefer`, use this command:

```
vxvol rdpol prefer volume_name preferred_plex_name
```

For example, the command to set the policy for `vol01` to read preferentially from the plex `vol01-02` is:

```
vxvol rdpol prefer vol01 vol01-02
```

To set the read policy to `select`, use this command:

```
vxvol rdpol select volume_name
```

Resizing Volumes With `vxresize`

You can use the `vxresize` command to resize a volume containing a file system. Although other commands can be used to resize volumes containing file systems, `vxresize` offers the advantage of automatically resizing the file system as well as the volume. For details on how to use `vxresize`, see the `vxresize`(1M) manual page. Note that only `vxfs` and `ufs` file systems can be resized with `vxresize`.

If your volume is larger than you need it to be, you can shrink the size of the volume.

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 uses.

Before removing a volume, make sure you do the following:

1. Remove all references to the volume.

2. If the volume is mounted as a file system, unmount it with the command:

   ```
   umount /dev/vx/dsk/volume_name
   ```

3. If the volume is listed in `/etc/vfstab`, remove its entry.

4. Make sure that the volume is stopped with the command:

   vxvol stop *volume_name*

   The vxvol stop command stops all VM activity to the volume.

After following these steps, remove the volume with this command:

vxedit rm *volume_name*

or

vxedit -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.

Note: The -r option of vxedit removes multiple objects. Exercise caution when using it.

You can also remove an entire volume with the vxassist command. Use the keywords remove and volume and provide the volume name on the command line as shown in this example:

   vxassist remove volume *volume_name*

Renaming and Removing Volume Manager Objects

The vxedit command has two functions:

- vxedit allows you to modify certain records in the volume management databases. Only fields that are not volume usage-type-dependent can be modified.

- vxedit can remove or rename Volume Manager objects.

Volume Manager objects that are associated with other objects are not removable by vxedit. This means that vxedit cannot remove:

- a subdisk that is associated with a plex

- a plex that is associated with a volume

> **Note:** Using the recursive suboption (−r) to the removal option of vxedit removes all objects from the specified object downward. In this way, a plex and its associated subdisks, or a volume and its associated plexes and their associated subdisks, can be removed by a single vxedit command.

For detailed information about how to use vxedit, refer to the vxedit(1M) manual page.

Mirroring a Volume

A mirror is a copy of a volume. The mirror copy is not stored on the same disk(s) as the original copy of the volume. Mirroring a volume ensures that the data in that volume is not lost if one of your disks fails.

Creating a Volume with Dirty Region Logging Enabled

To create a mirrored volume with Dirty Region Logging (DRL) enabled, create a mirrored volume with a log with this command:

    vxassist make *volume_name* *length* layout=mirror,log

vxassist creates one log plex for each log subdisk, by default. See the *VERITAS Volume Manager Getting Started Guide* for more information on DRL.

Mirroring an Existing Volume

A mirror (plex) can be added to an existing volume. This can be done with the vxassist command:

vxassist mirror *volume_name*

For example:

vxassist mirror voltest

creates a mirror of the volume voltest.

Another way to mirror an existing volume is by first creating a plex and then associating it with a volume, using these commands:

```
vxmake plex plex_name sd=subdisk_name ...

vxplex att volume_name plex_name
```

Mirroring All Volumes

To mirror all existing volumes on the system to available disk space, use this command:

```
/etc/vx/bin/vxmirror -g diskgroup -a
```

You can also configure the Volume Manager to create mirrored volumes by default. To do this, enter the command:

```
/etc/vx/bin/vxmirror -d yes
```

If you make this change, you can still make unmirrored volumes by specifying nmirror=1 as an attribute to the vxassist command. For example, to create an unmirrored 20-megabyte volume named nomirror, use the command:

```
vxassist make nomirror 20m nmirror=1
```

Adding a RAID-5 Log

Only one RAID-5 plex can exist per RAID-5 volume. Any additional plexes become RAID-5 log plexes, which are used to log information about data and parity being written to the volume. When a RAID-5 volume is created using vxassist, a log plex is created for that volume by default.

Add a RAID-5 log to an existing volume with this command:

```
vxassist addlog volume_name
```

To create a log for the RAID-5 volume volraid, use this command:

```
vxassist addlog volraid
```

Adding a DRL Log

To put Dirty Region Logging into effect for a volume, a log subdisk must be added to that volume and the volume must be mirrored. Only one log subdisk can exist per plex.

Add a DRL log to an existing volume with this command:

```
vxassist addlog volume_name
```

To create a log for the volume `vol03`, use this command:

```
vxassist addlog vol03
```

When `vxassist` is used to add a log subdisk to a volume, a log plex is also created to contain the log subdisk by default.

Once created, the plex containing a log subdisk can be treated as a regular plex. Data subdisks can be added to the log plex. The log plex and log subdisk can be removed using the same procedures used to remove ordinary plexes and subdisks.

Removing a DRL Log

You can also remove a log with the `vxassist` command as follows.

```
vxassist remove log volume_name
```

Use the attribute `nlog=` to specify the number of logs to be removed. By default, `vxassist` removes one log.

Removing a RAID-5 Log

Removing a RAID-5 log involves first dissociating the log from its volume and then removing the log and any associated subdisks completely.

Dissociate and remove the log from its volume in one operation with this command:

```
vxplex -o rm dis plex_name
```

To identify the log plex, use the command:

```
vxprint -ht volume_name
```

where *volume_name* is the name of the RAID-5 volume. This results in output that lists a plex with the STATE field of LOG.

To disassociate and remove the log plex `volraid-02` from `volraid`, use this command:

```
        vxplex -o rm dis volraid-02
```

The output of `vxprint -h` for `volraid` is now:

```
Disk group: rootdg

TY NAME        ASSOC      KSTATE    LENGTH PLOFFS STATE  TUTIL0 PUTIL0
v  volraid     raid5      ENABLED   32     -      ACTIVE -      -
pl volraid-01 volraid     ENABLED   32     -      ACTIVE -      -
sd disk01-03  volraid-01  -         32     0      -      -      -
sd disk02-02  volraid-01  -         32     0      -      -      -
```

You can also remove a RAID-5 log with the `vxassist` command as follows:

`vxassist remove log` *volume_name*

Use the attribute `nlog=` to specify the number of logs to be removed. By default, `vxassist` removes one log.

See the "Recovery" chapter in the *VERITAS Volume Manager Administrator's Reference Guide* for more information on volume STATES.

Stopping a Volume

Stopping a volume affects its availability to the user. Stopping a volume makes it unavailable.

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 this command:

`vxvol stop` *volume_name* `...`

For example, the command to stop a volume named `vol01` is:

`vxvol stop vol01`

To stop all ENABLED volumes, use this command:

`vxvol stopall`

If all mirrors of the volume become STALE, you can place the volume in maintenance mode. Then you can view the plexes while the volume is DETACHED and determine which plex to use for reviving the others. To place a volume in maintenance mode, use this command:

```
vxvol maint volume_name
```

To assist in choosing the revival source plex, list the unstarted volume and display its plexes.

To take plex vol01-02 offline, use this command:

```
vxmend off vol01-02
```

The vxmend command can change the state of an OFFLINE plex of a DISABLED volume to STALE. Then a vxvol start on the volume revives the plex. To put a plex named vol01-02 in the STALE state, use this command:

```
vxmend on vol01-02
```

Starting a Volume

Starting a volume affects its availability to the user. Starting a volume changes its state and makes it available for use.

Starting a volume changes the volume state from DISABLED or DETACHED to ENABLED. The success of this task 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 this command:

```
vxrecover -s volume_name ...
```

To start all DISABLED volumes, use this command:

```
vxrecover -s
```

If all mirrors of the volume become STALE, you can place the volume in maintenance mode. Then you can view the plexes while the volume is DETACHED and determine which plex to use for reviving the others. To place a volume in maintenance mode, use this command:

```
vxvol maint volume_name
```

To assist in choosing the revival source plex, list the unstarted volume and display its plexes.

To take plex `vol01-02` offline, use this command:

```
vxmend off vol01-02
```

The `vxmend` command can change the state of an `OFFLINE` plex of a `DISABLED` volume to `STALE`. Then a `vxvol start` on the volume revives the plex. To put a plex named `vol01-02` in the `STALE` state, use this command:

```
vxmend on vol01-02
```

Listing Unstartable Volumes

An unstartable volume can be incorrectly configured or have other errors or conditions that prevent it from being started. To display unstartable volumes, use the command `vxinfo`. `vxinfo` displays information on the accessibility and usability of one or more volumes:

```
vxinfo [volume_name]
```

Preparing a Volume to Restore From Backup

It is important to make backup 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. The Volume Manager allows you to back up volumes with minimal interruption to users.

You can back up a volume with `vxassist` by using the following procedure:

1. Create a snapshot mirror of the volume to be backed up.

   The `vxassist snapstart` task creates a write-only backup mirror, which is attached to and synchronized with the volume to be backed up. When synchronized with the volume, the backup mirror is ready to be used as a snapshot mirror. However, it continues being updated until it is detached during the actual snapshot portion of the procedure. This may take some time, depending on the volume size.

Create a snapshot mirror for a volume with this command:

```
vxassist snapstart volume_name
```

To create a snapshot mirror of a volume called `voldef`, use this command:

```
vxassist snapstart voldef
```

2. Choose a suitable time to create a snapshot volume.

   If possible, plan to take the snapshot at a time when users are accessing the volume as little as possible.

3. Create a snapshot volume that reflects the original volume at the time of the snapshot.

   The online backup procedure is completed by running the `vxassist snapshot` command on the volume with the snapshot mirror. This task detaches the finished snapshot mirror, creates a new normal volume, and attaches the snapshot mirror to it. The snapshot then becomes a read-only volume. This step should only take a few minutes.

   Create a snapshot volume with this command:

   vxassist snapshot volume_name new_volume_name

   To create a snapshot volume of `voldef`, use this command:

   vxassist snapshot voldef snapvol

   The snapshot volume can now be used by backup utilities while the original volume continues to be available for applications and users.

You can back up the snapshot volume by whatever means you prefer. To avoid wasting space, you should remove the snapshot volume when your backup is complete, The snapshot volume occupies as much space as the original volume.

Recovering a Volume

A system crash or an I/O error can corrupt one or more plexes of a volume and leave no plex CLEAN or ACTIVE. You can 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 the CLEAN state, use this command:

```
vxmend fix clean plex_name
```

For example, the command to place the plex named `vol01-02` in the CLEAN state is:

```
vxmend fix clean vol01-02
```

For information about how to use `vxmend`, refer to the `vxmend`(1M) manual page. For more information on recovery, refer to the *VERITAS Volume Manager Administrator's Reference Guide*, Chapter 1, "Recovery."

Removing a Mirror

When a mirror is no longer needed, you can remove it. Examples of tasks that require removal of a mirror are:

• to provide free disk space

• reduce the number of mirrors in a volume to increase the length of another mirror and its associated volume. The plexes and subdisks are removed, then the resulting space can be added to other volumes

• remove a temporary mirror created to backup a volume and is no longer required

• change the layout of a mirror from concatenated to striped, or back

---

Note: The last valid plex associated with a volume cannot be removed.

CAUTION! To save the data on a mirror to be removed, the configuration of that mirror must be known. Parameters from that configuration (stripe unit size and subdisk ordering) are critical to the creation of a new mirror to contain the same data. Before this type of mirror is removed, its configuration must be recorded.

A mirror can be dissociated and removed from the associated volume with this command:

```
vxplex -o rm dis plex_name
```

To dissociate and remove a mirror named `vol01-02`, use this command:

```
vxplex -o rm dis vol01-02
```

This removes the mirror `vol01-02` and all associated subdisks.

Note: You can first dissociate the plex and subdisks, then remove them with the commands:
`vxplex dis` *plex_name*
`vxedit –r rm` *plex_name*
Together, these commands accomplish the same as `vxplex -o rm dis`.

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 dissociated from the plex, and the new subdisk is associated with the plex. The association is at the same offset within the plex as the source subdisk. To move a subdisk, use this command:

`vxsd mv` *old_subdisk_name new_subdisk_name*

For the subdisk move task to work correctly, these 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.

Splitting Subdisks

Splitting a subdisk divides an existing subdisk into two subdisks. To split a subdisk, use this command:

`vxsd –s` *size* `split` *subdisk_name newsd1 newsd2*

where:

- *subdisk_name* is the name of the original subdisk

- *newsd1* is the name of the first of the two subdisks to be created

• *newsd2* is the name of the second subdisk to be created

The -s option is required to specify the size of the *first* of the two subdisks to be created. The second subdisk occupies the remaining space used by the original subdisk.

If the original subdisk is associated with a plex before the task, upon completion of the split, both of the resulting subdisks are associated with the same plex.

To split the original subdisk into more than two subdisks, repeat the previous command as many times as necessary on the resulting subdisks.

Joining Subdisks

Joining subdisks 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. To join a subdisk, use this command:

vxsd join *subdisk1 subdisk2 new_subdisk*

Displaying Volume Configuration Information

You can use the vxprint command to display information about how a volume is configured.

You can display the volume, plex, and subdisk record information for all volumes in the system with this command:

```
vxprint -ht
```

This is an example of the vxprint output:

```
Disk group: rootdg

DG NAME       NCONFIG  NLOG    MINORS   GROUP-ID
DM NAME       DEVICE   TYPE    PRIVLEN  PUBLEN   STATE
V  NAME       USETYPE  KSTATE  STATE    LENGTH   READPOL     PREFPLEX
PL NAME       VOLUME   KSTATE  STATE    LENGTH   LAYOUT      NCOL/WID  MODE
SD NAME       PLEX     DISK    DISKOFFS LENGTH   [COL/]OFF   DEVICE    MODE
```

```
dg rootdg    759096729.1025.tweety

dm disk10    c1t0d0s2 sliced  559      1044400   -
dm disk20    c2t0d0s2 sliced  559      1044400   -

v  pubs      fsgen    ENABLED ACTIVE  2288       SELECT    -
pl pubs-01   pubs     ENABLED ACTIVE  2288       CONCAT    -         RW
sd disk10-01 pubs-01  disk10  0       2288       0         c0t0d0    ENA

v  voldef    sgen     ENABLED ACTIVE  20480      SELECT    -
pl voldef-01 voldef   ENABLED ACTIVE  20480      CONCAT    -         RW
sd disk10-02 voldef-0 disk10  2288    20480      0         c0t1d0    ENA
```

where dg is a disk group, dm is a disk, v is a volume, pl is a plex, and sd is a subdisk. The top few lines indicate the headers that match each type of output line that follows. Each volume is listed along with its associated plex(es) and subdisk(s).

You can display volume-related information for a specific volume with this command:

```
vxprint -t  volume_name
```

To display information about voldef, use this command:

```
vxprint -t voldef
```

This is an example of the vxprint output:

```
Disk group: rootdg

V  NAME    USETYPE KSTATE  STATE   LENGTH  READPOL  PREFPLEX

v  voldef  fsgen   ENABLED ACTIVE  20480   SELECT   -
```

Performing Online Backup

Volume Manager provides snapshot backups of volume devices. This is done through vxassist and other commands. There are various procedures for doing backups, depending upon the 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 plex that is too small is used, an incomplete copy results.

The recommended approach to volume backup is by using the `vxassist` command which is easy to use. The `vxassist snapstart`, `snapwait`, and `snapshot` tasks provide a way to do online backup of volumes with minimal disruption to users.

The vxassist `snapshot` procedure consists of two steps:

1. Running `vxassist snapstart` to create a snapshot mirror

2. Running `vxassist snapshot` to create a snapshot volume

Note: You can use the `vxassist` command to create a `snapshot` of a RAID-5 volume by using the recommended approach to volume backup described in this section.

The `vxassist snapstart` step 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` mirror. The end of the update procedure is indicated by the new `snapshot` mirror changing its state to `SNAPDONE`. This change can be tracked by the `vxassist snapwait` task, which waits until at least one of the mirrors changes its state to `SNAPDONE`. If the attach process fails, the `snapshot` mirror is removed and its space is released.

Once the `snapshot` mirror is synchronized, it continues being updated until it is detached. You can then select a convenient time at which to create a `snapshot` volume as an image of the existing volume. You 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` mirror is long in contrast to the brief amount of time that it takes to create the `snapshot` volume.

The online backup procedure is completed by running the `vxassist snapshot` command on a volume with a `SNAPDONE` mirror. This task detaches the finished `snapshot` (which becomes a normal mirror), creates a new normal volume and attaches the `snapshot` mirror to the `snapshot` volume. The `snapshot` then becomes a normal, functioning mirror and the state of the `snapshot` is set to `ACTIVE`.

If the `snapshot` procedure is interrupted, the `snapshot` mirror is automatically removed when the volume is started.

Use the following steps to perform a complete `vxassist` backup:

1. Create a `snapshot` mirror for a volume with this command:

   `vxassist  snapstart  volume_name`

2. When the `snapstart` step is complete and the mirror is in a SNAPDONE state, choose a convenient time to complete the `snapshot` task. Inform users of the upcoming `snapshot` and ask them to save files and refrain from using the system briefly during that time.

3. Create a `snapshot` volume that reflects the original volume with this command:

   `vxassist  snapshot  volume_name  temp_volume_name`

4. Use `fsck` (or some utility appropriate for the application running on the volume) to clean the temporary volume's contents. For example, you can use this command:

   `fsck  -y  /dev/vx/rdsk/temp_volume_name`

5. Copy the temporary volume to tape, or to some other appropriate backup media.

6. Remove the new volume with this command:

   `vxedit  -rf  rm  temp_volume_name`

3

VERITAS Volume Manager Command Line Interface Administrator's Guide

Volume Administration Tasks 4

Introduction

This chapter describes how to maintain a system configuration under Volume Manager control by using concepts and procedures that are normally "hidden" from the casual user. Some procedures are managed automatically by higher level commands such as `vxassist`.

Information is presented as overviews of commands and more detailed descriptions of operations using the commands. Online backup information is also included.

Note: Most Volume Manager commands require superuser or other appropriate privileges.

The following main topics are covered in this chapter:

- Volume Administration Tasks
- Volume Manager Commands
- Subdisk Tasks
- Plex Tasks
- Volume Tasks
- Online Relayout

⩧ 4

Volume Administration Tasks

Volume Manager can be used to perform system and configuration management tasks on its objects: disks, disk groups, subdisks, plexes, and volumes. Volumes contain two types of objects:

- subdisk—a region of a physical disk

- plex—a series of subdisks linked together in an address space

Disks and disk groups must be initialized and defined to the Volume Manager before volumes can be created. Volumes can be created in either of the following ways:

- Automatically—you can create and manipulate volumes by using the `vxassist` command. `vxassist` creates required plexes and subdisks by using only the basic attributes of the desired volume as input. `vxassist` can also modify existing volumes. It automatically modifies any underlying or associated objects. `vxassist` uses default values for many volume attributes, unless you provide specific values.

- Manually—you can create volumes by first creating the volume subdisks, then the plexes, then the volume itself (a bottom-up approach). This approach often requires detailed user input and an understanding of Volume Manager concepts. Creating volumes manually is useful for manipulating volumes with specific, non-default attributes. The manual approach does not use the `vxassist` command.

Note: This section focuses on how to create and modify volumes using the manual "bottom-up" approach. However, it can be more convenient to use the automated `vxassist` approach to volume management.

Many of the concepts and procedures in this section are normally "hidden" from the casual user. Some procedures are managed automatically by higher level commands such as `vxassist`. Some of the commands and procedures described are not normally performed by system administrators.

The steps normally involved in creating volumes manually are:

- create subdisks

- create plexes

- associate subdisks and plexes

- create volumes

- associate volumes and plexes

- initialize volumes

Before you create volumes, you should determine which volume layout best suits your system needs. See the *VERITAS Volume Manager Administrator's Reference Guide* and the *VERITAS Volume Manager Getting Started Guide* for more information about volume layouts.

Volume Manager Commands

This section describes volume Manager commands that perform system administration and maintenance functions.

Using `vxassist`

You can use the `vxassist` command to do volume management. `vxassist` is an automated one-step interface to Volume Manager Tasks. Unlike some Volume Manager commands, `vxassist` does not require a thorough understanding of Volume Manager concepts.

`vxassist` can do tasks that would otherwise require the use of several other Volume Manager commands. `vxassist` automatically performs all underlying and related tasks that would otherwise be done by the user (in the form of other commands).

`vxassist` does not conflict with existing Volume Manager commands or preclude their use. Objects created by `vxassist` are compatible and inter-operable with objects created by other VM commands and interfaces.

`vxassist` performs these tasks:

- finds space for and creates volumes

- finds space for and creates mirrors for existing volumes

- finds space for and extends existing volumes

- shrinks existing volumes and returns unused space

- provides facilities for the online backup of existing volumes

- provides an estimate of the maximum size for a new or existing volume

For detailed information about how to use vxassist, refer to the vxassist(1M) manual page.

Advantages of the vxassist Command

Some of the advantages of using vxassist include:

- The use of vxassist involves only one step (command) on the part of the user.

- You are required to specify only minimal information to vxassist, yet you can optionally specify additional parameters to modify its actions.

- vxassist tasks result in a set of configuration changes that either succeed or fail as a group, rather than individually. Most vxassist tasks work so that system crashes or other interruptions do not leave intermediate states to be cleaned up. If vxassist finds an error or an exceptional condition, it exits without leaving partially-changed configurations. The system is left in the same state as it was prior to the attempted vxassist task.

How vxassist Works

The vxassist command allows you to create and modify volumes. You specify the basic volume creation or modification requirements and vxassist performs the necessary tasks.

vxassist obtains most of the information it needs from sources other than your input. vxassist obtains information about the existing objects and their layouts from the objects themselves.

For tasks requiring new disk space, vxassist seeks out available disk space and allocates it in the configuration that conforms to the layout specifications and that offers the best use of free space.

The vxassist command typically takes this form:

vxassist *keyword volume_name [attributes...]*

where *keyword* selects the task to perform. The first argument after a vxassist keyword is a volume name, which is followed by a set of attributes. For details on available vxassist keywords and attributes, refer to the vxassist(1M) manual page.

`vxassist` creates and manipulates volumes based on a set of established defaults, but also allows you to supply preferences for each task.

`vxassist` **Defaults**

`vxassist` uses a set of tunable parameters that can be specified in the defaults files or at the command line. The tunable parameters default to reasonable values if they are not listed on the command line. Tunables listed on the command line override those specified elsewhere. Here are the tunable parameters:

- 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).

- System-wide defaults file—The system-wide defaults file contains default values that you can alter. These values are used for tunables that are not specified on the command line or in an alternate defaults file.

- Alternate defaults file—A non-standard defaults file, specified with the command `vxassist -d` *alt_defaults_file*.

- Command line—The tunable values specified on the command line override any values specified internally or in defaults files.

Defaults File

The default behavior of `vxassist` is controlled by the tunables specified in the `/etc/default/vxassist` file. 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 `vxassist` command line (refer to the `vxassist`(1M) manual page for details).

Here is a sample `vxassist` defaults file:

```
#  by default:
#  create unmirrored, unstriped volumes
#  allow allocations to span drives
#  with RAID-5 create a log, with mirroring don't create a log
#  align allocations on cylinder boundaries
   layout=nomirror,nostripe,span,nocontig,raid5log,noregionlog,
   diskalign
```

```
#  use the fsgen usage type, except when creating RAID-5 volumes
   usetype=fsgen

# allow only root access to a volume
  mode=u=rw,g=,o=
  user=root
  group=root

# when mirroring, create two mirrors
  nmirror=2

#  for regular striping, by default create between 2 and 8 stripe
#  columns
  max_nstripe=8
  min_nstripe=2

#  for RAID-5, by default create between 3 and 8 stripe columns
  max_nraid5stripe=8
  min_nraid5stripe=3

#  create 1 log copy for both mirroring and RAID-5 volumes, by
  default
  nregionlog=1
  nraid5log=1

#  by default, limit mirroring log lengths to 32Kbytes
  max_regionloglen=32k

#  use 64K as the default stripe unit size for regular volumes
  stripe_stwid=64k

#  use 16K as the default stripe unit size for RAID-5 volumes
  raid5_stwid=16k
```

Volume Configuration Daemon `vxdctl`

The volume configuration daemon (`vxconfigd`) is the interface between the Volume Manager commands and the kernel device drivers (vol, vols, DMP). The `config` device is a special device file created by the Volume Manager that interacts with `vxdctl` to make system configuration changes.

Some `vxdctl` tasks involve modifications to the `volboot` file that indicates the location of some root configuration copies.

The `vxdctl` command is the interface to `vxconfigd` and is used for:

- performing tasks related to the state of the `vxconfigd` daemon

- managing boot information and Volume Manager root configuration initialization

- manipulating the contents of the `volboot` file that contains a list of disks containing root configuration databases (this is not normally necessary, as Volume Manager automatically locates all disks on the system)

- reconfiguring the DMP database (if used on your system) to reflect new disk devices attached to the system, and removal of any disk devices from the system

- creating DMP (if used on your system) device nodes in the device directories `/dev/vx/dmp` and `/dev/vx/rdmp`

- reflecting the change in path type into the DMP database for active/passive disk arrays. You can change the path type from primary to secondary and vice-versa through the utilities provided by disk array vendors

For detailed information about how to use `vxdctl`, refer to the `vxdctl`(1M) manual page.

Creating Volume Manager Objects With `vxmake`

You can use the `vxmake` command to add a new volume, plex, or subdisk to the set of objects managed by Volume Manager. `vxmake` adds a new record for that object to the Volume Manager configuration database. You can create records from parameters specified on the command line or by using a description file.

If you specify operands on the command line as shown in this example:

```
vxmake  -Uusage_type vol  volume_name len=length plex=plex_name,..
```

where:

- the first operand (keyword) determines the kind of object to be created
- the second operand is the name given to that object
- additional operands specify attributes for the object

If no operands are specified on the command line, then a description file is used to specify the records to create.

A *description file* is a file that contains plain text that describes the objects to be created with vxmake. A description file can contain several commands, and can be edited to perform a list of tasks. The description file is read from standard input, unless the -d *description_file* option specifies a filename. The following is a sample description file:

```
#rectyp #name        #options
sd      disk3-01    disk=disk3 offset=0 len=10000
sd      disk3-02    disk=disk3 offset=25000 len=10480
sd      disk4-01    disk=disk4 offset=0 len=8000
sd      disk4-02    disk=disk4 offset=15000 len=8000
sd      disk4-03    disk=disk4 offset=30000 len=4480
plex    db-01       layout=STRIPE ncolumn=2 stwidth=16k
                     sd=disk3-01:0/0,disk3-02:0/10000,disk4-01:1/0,\
                     disk4-02:1/8000, disk4-03:1/16000
sd      ramd1-01    disk=ramd1 len=640
                    comment="Hot spot for dbvol
plex    db-02       sd=ramd1-01:40320
vol     db          usetype=gen plex=db-01,db-02
                    readpol=prefer prefname=db-02
                    comment="Uses mem1 for hot spot in last 5m
```

This description file specifies a volume with two plexes. The first plex has five subdisks on physical disks. The second plex is preferred and has one subdisk on a volatile memory disk.

For detailed information about how to use vxmake, refer to the vxmake(1M) manual page.

Performing Plex Tasks With `vxplex`

The `vxplex` command performs Volume Manager tasks on a plex or on volume-and-plex combinations. The first operand is a keyword that specifies the task to perform. The remaining operands specify the objects to which the task is to be applied.

You can use the `vxplex` command to:

- Attach or detach a plex and a volume. A detached plex does not share in I/O activity to the volume, but remains associated with the volume. A detached plex is reattached when a volume is next started.

- Dissociate a plex from the associated volume. When a plex is dissociated, its relationship to the volume is broken. The plex is then available for other uses and can be associated with a different volume. This is useful as part of a backup procedure.

- Copy the contents of the specified volume to the named plex(es). This task makes a copy of a volume (for backup purposes) without mirroring the volume in advance.

- Move the contents of one plex to a new plex. This is useful for moving a plex on one disk to a different location.

For detailed information about how to use `vxplex`, refer to the `vxplex`(1M) manual page.

Performing Subdisk Tasks With `vxsd`

The `vxsd` command maintains subdisk-mirror associations. You can use `vxsd` to:

- associate or dissociate a subdisk from its associated mirror

- move the contents of a subdisk to another subdisk

- split one subdisk into two subdisks that occupy the same space as the original

- join two contiguous subdisks into one.

Note: Some `vxsd` tasks can take a large amount of time to complete.

For detailed information about how to use `vxsd`, refer to the `vxsd(1M)` manual page.

Correcting Problems With `vxmend`

The `vxmend` command performs Volume Manager usage-type-specific tasks on volumes, plexes, and subdisks. These tasks fix simple problems in configuration records (such as clearing utility fields, changing volume or plex states, and offlining or onlining volumes or plexes).

`vxmend` is used primarily to escape from a state that was accidentally reached. The offline and online functions can be performed with disk-related commands.

For detailed information about how to use `vxmend`, refer to the `vxmend(1M)` manual page.

Printing Configuration Information With `vxprint`

The `vxprint` command displays information from records in a Volume Manager configuration database. You can use this command to display partial or complete information about any or all Volume Manager objects. The format can be hierarchical to clarify relationships between Volume Manager objects. UNIX system utilities such as `awk`, `sed`, or `grep` can also use `vxprint` output.

For detailed information about how to use `vxprint`, refer to the `vxprint(1M)` manual page.

Printing Volume Statistics With `vxstat`

The `vxstat` command prints statistics about Volume Manager objects and block devices under Volume Manager control. `vxstat` reads the summary statistics for Volume Manager objects and formats them to the standard output. These statistics represent Volume Manager activity from the time the system initially booted or from the last time statistics were cleared. If no Volume Manager object name is specified, statistics from all volumes in the configuration database are reported.

For detailed information about how to use `vxstat`, refer to the `vxstat(1M)` manual page and the *VERITAS Volume Manager Administrator's Reference Guide*, Chapter 2, "VxVM Performance Monitoring."

Tracing Volume Tasks With `vxtrace`

The `vxtrace` command prints kernel I/O error or I/O error trace event records on the standard output or writes them to a file in binary format. Binary trace records written to a file can also be read back and formatted by `vxtrace`.

If no operands are given, then either all error trace data or all I/O trace data on all virtual disk devices are reported. With error trace data, you can select all accumulated error trace data, wait for new error trace data, or both (the default). Selection can be limited to a specific disk group, to specific Volume Manager kernel I/O object types, or to particular named objects or devices.

For detailed information about how to use `vxtrace`, refer to the `vxtrace`(1M) manual page and the *VERITAS Volume Manager Administrator's Reference Guide*, Chapter 2, "VxVM Performance Monitoring."

Performing Volume Tasks With `vxvol`

The `vxvol` command performs Volume Manager tasks on volumes. You can use `vxvol` to:

- initialize a volume

- start a volume

- stop a volume

- establish 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` (however, it is seldom useful to stop a volume). For more information about volume states, see the "Recovery" chapter in the *VERITAS Volume Manager Administrator's Reference Guide*.

You can use `vxvol` to specify one of these read policies:

- `round` reads each plex in turn in "round-robin" fashion for each nonsequential I/O detected. Sequential access causes only one plex to be accessed. This takes advantage of the drive or controller read-ahead caching policies.

- `prefer` reads first from a plex that has been named as the preferred plex.

- `select` chooses a default policy based on plex associations to the volume. If the volume has an enabled striped plex, `select` defaults to preferring that plex; otherwise, it defaults to round-robin.

For detailed information about how to use `vxvol`, refer to "Changing the Volume Read Policy" in Chapter 3, and the `vxvol`(1M) manual page.

Subdisk Tasks

Subdisks are low-level building blocks in a Volume Manager configuration. The following sections describe tasks that you can perform on subdisks. Subdisk tasks include:

- creating a subdisk
- removing a subdisk
- displaying subdisk information
- associating a subdisk
- associating log subdisks
- dissociating a subdisk
- changing a subdisk
- moving a subdisk
- splitting a subdisk
- joining subdisks

Creating Subdisks

You can use the `vxmake` command to create Volume Manager objects, such as subdisks. When you create a subdisk, you must specify the:

- name of the subdisk
- length of the subdisk
- starting point (offset) of the subdisk within the disk
- disk media name

To create a subdisk, use this command:

`vxmake sd` *subdisk_name disk***,***offset***,***len*

For example, the command to create a subdisk named `disk02-01` that starts at the beginning of disk `disk02` and has a length of 8000 sectors is shown here:

`vxmake sd disk02-01 disk02,0,8000`

By default, Volume Manager commands take sizes in sectors. Adding a suffix (such as `k`, `m`, or `g`) changes the unit of size.

Note: To preserve (encapsulate) any data that exists on the disk, a plex and volume must be created to cover that data.

If you intend to use the new subdisk to build a volume, you must associate the subdisk with a plex (see "Associating a Subdisk"). Subdisks for all plex layouts (concatenated, striped, RAID-5) are created the same way.

Removing Subdisks

To remove a subdisk, use this command:

`vxedit rm` *subdisk_name*

For example, you can use this command to remove a subdisk named `disk02-01`:

`vxedit rm disk02-01`

Displaying Subdisk Information

The `vxprint` command displays information about Volume Manager objects. To display general information for all subdisks, use this command:

`vxprint -st`

The `-s` option specifies information about subdisks. The `-t` option prints a single-line output record that depends on the type of object being listed.

You can display complete information about a particular subdisk by using this command:

```
vxprint -l subdisk_name
```

For example, here is the command to obtain all information on a subdisk named `disk02-01`:

```
vxprint -l disk02-01
```

This command provides the following output:

```
Disk group: rootdg

Subdisk:  disk02-01
info:     disk=disk02 offset=0 len=205632
assoc:    vol=mvol plex=mvol-02 (offset=0)
flags:    enabled
device:   device=c2t0d1s2 path=/dev/vx/dmp/c2t0d1s4 diskdev=32/68
```

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. The entire area that the subdisk fills must not be occupied by any portion of another subdisk. There are several ways that subdisks can be associated with plexes, depending on the overall state of the configuration.

If you have already created all the subdisks needed for a particular plex, associate subdisks at plex creation by using the command:

```
vxmake plex plex_name sd=subdisk_name,...
```

For example, the following command creates the plex `home-1` and associates subdisks `disk02-01`, `disk02-00`, and `disk02-02` with plex `home-1`. This is done during the plex creation process:

```
vxmake plex home-1 sd=disk02-01,disk02-00,disk02-02
```

Subdisks are associated in order starting at offset 0. If you use this type of command, you do not have to specify the multiple commands needed to create the plex and then associate each of the subdisks with that plex. In this example, the subdisks are associated to the plex in the order they are listed (after `sd=`). The disk space defined as `disk02-01` is first, the disk space `disk02-00` is second, and `disk02-02` 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 with this command:

vxsd assoc *plex_name sd_name* [*sd_name2 sd_name3 ...*]

For example, you can use this command to associate subdisks named `disk02-01`, `disk02-00`, and `disk02-02` with a plex named `home-1`:

vxsd assoc home-1 disk02-01 disk02-00 disk02-01

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 command. The -l option associates subdisks at a specific offset within the plex.

The -l option is needed when you have created a sparse plex (that is, a plex with a gap between its subdisks) for a particular volume, and want to make this plex complete. To make the plex complete, you need to create a subdisk of a size that fits the hole in the sparse plex exactly. Then you associate the subdisk with the plex by specifying the offset of the beginning of the hole in the plex. Use this command:

vxsd -l *offset assoc sparse_plex_name exact_size_subdisk*

Note: The subdisk must be exactly the right size because Volume Manager does not allow for the space defined by two subdisks to overlap within a single plex.

For striped subdisks, you can specify a column number and column offset for the subdisk:

vxsd -l *column_#/offset* assoc *plex_name sd_name* ...

If only one number is specified with the -l option for striped plexes, the number is interpreted as a column number and the subdisk is associated at the end of the column.

Associating Log Subdisks

Log subdisks are defined for and added to a plex that is to become part of a volume using Dirty Region Logging. Dirty Region Logging is enabled for a volume when the volume is mirrored and has at least one log subdisk.

For a description of Dirty Region Logging, refer to "Dirty Region Logging," in Chapter 2 of the *VERITAS Volume Manager Getting Started Guide*, and "Dirty Region Logging (DRL) Guidelines" in Chapter 3 of the *Getting Started Guide*. Log subdisks are ignored as far as the usual plex policies are concerned, and are only used to hold the dirty region log.

Note: Only one log subdisk can be associated with a plex. Because this log subdisk is frequently written, care should be taken to position it on a disk that is not heavily used. Placing a log subdisk on a heavily-used disk can degrade system performance.

To add a log subdisk to an existing plex, use this command:

`vxsd aslog` *plex subdisk*

where *subdisk* is the name to be used as a log subdisk. The plex must be associated with a mirrored volume before DRL takes effect.

For example, you can use this command to associate a subdisk named `disk02-01` with a plex named `vol01-02` (which is already associated with volume `vol01`):

`vxsd aslog vol01-02 disk02-01`

Note: You can also use the following command to add a log subdisk to an existing volume:
  `vxassist addlog` *volume_name disk*
This command automatically creates a log subdisk within a log plex for the specified volume.

Dissociating Subdisks

To break an established connection between a subdisk and the plex to which it belongs, the subdisk is *dissociated* from the plex. A subdisk is dissociated when the subdisk is removed or used in another plex. To dissociate a subdisk, use this command:

`vxsd dis` *subdisk_name*

For example, you can dissociate a subdisk named `disk02-01` from the plex with which it is currently associated, by using this command:

```
vxsd dis disk02-01
```

Note: You can also remove subdisks with the command:
`vxsd -orm dis` *subdisk_name*

Changing Subdisk Attributes

CAUTION! Change subdisk attributes with extreme care, and only if necessary.

The `vxedit` command changes attributes of subdisks to other Volume Manager objects. To change information relating to a subdisk, use this command:

`vxedit set` *field=value ... subdisk_name*

For example, here is the command to change the comment field of a subdisk named `disk02-01`:

`vxedit set comment=`*"new_comment"* `disk02-01`

The subdisk fields that can be changed using `vxedit` are:

- name
- `putil[`*n*`]` fields
- `tutil[`*n*`]` fields
- `len` (only if the subdisk is dissociated)
- comment

Note: Entering data in the `putil0` field prevents the subdisk from being used as part of a plex, if it is not already part of a plex.

Plex Tasks

Plexes are logical groupings of subdisks that create an area of disk space independent of physical disk size or other restrictions. Replication (mirroring) of disk data is done by creating multiple plexes for a single volume. Each plex

contains an identical copy of the volume data. Because each plex must reside on different disks, the replication provided by mirroring prevents data loss in the event of a single-point disk-subsystem failure. Multiple plexes also provide increased data integrity and reliability.

Plex tasks include:

- creating a plex
- backup using mirroring
- associating a plex
- dissociating and removing a plex
- listing all plexes
- displaying plexes
- changing plex attributes
- changing plex status
- moving plexes
- copying plexes

Creating Plexes

The `vxmake` command creates Volume Manager objects, such as plexes. When you create a plex, you identify subdisks and associate them to the plex that you want to create.

To create a plex from existing subdisks, use this command:

`vxmake plex` *plex_name sd=subdisk_name*`,...`

For example, the command to create a concatenated plex named `vol01-02` using two existing subdisks named `disk02-01` and `disk02-02` is:

`vxmake plex vol01-02 sd=disk02-01,disk02-02`

Creating a Striped Plex

To create a striped plex, you need to specify additional attributes. For example, the command to create a striped plex named `pl-01` with a stripe width of 32 sectors and 2 columns is:

```
vxmake plex pl-01 layout=stripe stwidth=32 ncolumn=2\
sd=disk01-01,disk02-01
```

If you intend to use a plex to build a volume, you must associate the plex with the volume (see "Associating Plexes," for more information.

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 this command:

vxplex att *volume_name plex_name*

For example, the command to associate a plex named `vol01-02` with a volume named `vol01` is:

vxplex att vol01 vol01-02

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:

vxmake -U *usetype* vol *volume_name* plex=*plex_name1, plex_name2...*

For example, the command to create a mirrored, `fsgen`-type volume named `home` and associate two existing plexes named `home-1` and `home-2` is:

vxmake -Ufsgen vol home plex=home-1,home-2

Note: You can also use the following command on an existing volume to add and associate a plex: vxassist mirror *volume_name*

Dissociating and Removing Plexes

When a plex is no longer needed, you can remove it. You may need to remove a plex in the following situations:

* to provide free disk space

* to reduce the number of mirrors in a volume so you can increase the length of another mirror and its associated volume. When the plexes and subdisks are removed, the resulting space can be added to other volumes

- to remove a temporary mirror that was created to back up a volume and is no longer needed
- to change the layout of a plex

CAUTION! To save the data on a plex to be removed, the configuration of that plex must be known. Parameters from that configuration (stripe unit size and subdisk ordering) are critical to the creation of a new plex to contain the same data. Before a plex is removed, you must record its configuration. See "Displaying Plex Information" for more information.

A plex can be dissociated and removed from the associated volume with this command:

vxplex -o rm dis *plex_name*

To dissociate and remove a plex named vol01-02, use this command:

vxplex -o rm dis vol01-02

This removes the plex vol01-02 and all associated subdisks.

Note: You can first dissociate the plex and subdisks, then remove them with the commands:
vxplex dis *plex_name*
vxedit -r rm *plex_name*
Together, these commands accomplish the same as vxplex -o rm dis.

Backup Using Mirroring

If a volume is mirrored, backup can be done on that volume by taking one of the volume mirrors offline for a period of time. This removes the need for extra disk space for the purpose of backup only. However, it also removes redundancy of the volume for the duration of the time needed for the backup to take place.

Note: The information in this section does not apply to RAID-5.

You can perform backup of a mirrored volume on an active system with these steps:

1. Optionally stop user activity for a short time to improve the consistency of the backup.

2. Dissociate one of the volume mirrors (`vol01-01`, for this example):

   ```
   vxplex dis vol01-01
   ```

3. Create a new, temporary volume using the dissociated plex:

   ```
   vxmake -U gen vol tempvol plex=vol01-01
   ```

4. Start the temporary volume:

   ```
   vxvol start tempvol
   ```

5. Do appropriate backup procedures, using the temporary volume.

6. Stop the temporary volume:

   ```
   vxvol stop tempvol
   ```

7. Dissociate the backup plex from its temporary volume:

   ```
   vxplex dis vol01-01
   ```

8. Reassociate the backup plex with its original volume to regain redundancy of the volume:

   ```
   vxplex att vol01 vol01-01
   ```

9. Remove the temporary volume:

   ```
   vxedit rm tempvol
   ```

For information on an alternative online backup method using the `vxassist` command, refer to "Performing Online Backup."

Displaying Plex Information

Listing plexes helps identify free plexes for building volumes. Using the `vxprint` command with the plex (–p) option lists information about all plexes.

To display detailed information about all plexes in the system, use this command:

```
vxprint -lp
```

To display detailed information about a specific plex, use this command:

```
vxprint -l plex_name
```

The –t option prints a single line of information about the plex. To list free plexes, use this command:

```
vxprint -pt
```

Changing Plex Attributes

CAUTION! Change plex attributes with extreme care, and only if necessary.

The vxedit command changes the attributes of plexes and other volume Manager objects. To change plex attributes, use this command:

```
vxedit set field=value ... plex_name ...
```

The `comment` field and the `putil` and `tutil` fields are used by Volume Manager commands after plex creation. `putil` attributes are maintained on reboot; `tutil` fields are temporary and are not retained on reboot.

Both `putil` and `tutil` have three functions and are numbered according to those functions. These fields can be modified as needed. Volume Manager uses the utility fields marked `putil0` and `tutil0`. Other VERITAS products use those marked `putil1` and `tutil1`. Fields marked `putil2` and `tutil2` are user fields. Table 1 lists the functions of the `putil` and `tutil` fields.

Table 1 The putil[n] and tutil[n] Fields

Field	Description of Utility Fields
putil0	Reserved for use by Volume Manager commands and is retained on reboot.
putil1	Reserved for use by high-level utilities such as the graphical user interface. This field is retained on reboot.
putil2	Reserved for use by the system administrator or site-specific applications. This field is retained on reboot.
tutil0	Reserved for use by Volume Manager commands and is cleared on reboot.
tutil1	Reserved for use by high-level utilities such as the graphical user interface. This field is cleared on reboot.
tutil2	Reserved for use by the system administrator or site-specific applications. This field is cleared on reboot.

The command:

```
vxedit set comment="my plex" tutil2="u" user="admin" \
vol01-02
```

uses vxedit to set the following attributes:

- set the comment field (identifying what the plex is used for) to my plex

- set tutil2 to u to indicate that the subdisk is in use

- change the user ID to admin

To prevent a particular plex from being associated with a volume, set the putil0 field to a non-null string, as specified in this command:

```
vxedit set putil0="DO-NOT-USE" vol01-02
```

Changing Plex Status: Detaching and Attaching Plexes

Once a volume has been created and placed online (ENABLED), Volume
Manager can temporarily disconnect plexes from the volume. This is useful, for
example, when the hardware on which the 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 offline and
attaching and detaching its plexes. The two commands used to accomplish
disk failure resolution are vxmend and vxplex.

To take a plex OFFLINE so that repair or maintenance can be performed on the
physical disk containing subdisks of that plex, use this command:

vxmend off *plex_name* ..

If a disk has a head crash, you should put all plexes that have associated
subdisks on the affected disk OFFLINE. For example, if plexes vol01-02 and
vol02-02 had subdisks on a drive to be repaired, use this command:

vxmend off vol01-02 vol02-02

This command places vol01-02 and vol02-02 in the OFFLINE state, and
they remain in that state until changed.

Detaching Plexes

To temporarily detach one plex in a mirrored volume, use this command:

vxplex det *plex_name*

For example, the command to temporarily detach a plex named vol01-02 and
place it in maintenance mode is:

vxplex det vol01-02

This command temporarily detaches the plex, but maintains the association
between the plex and its volume. However, the plex is not used for I/O. A plex
detached with the preceding command is recovered at system reboot. The plex
state is set to STALE, so that if a vxvol start command is run on the
appropriate volume (for example, on system reboot), the contents of the plex is
recovered and made ACTIVE.

When the plex is ready to return as an active part of its volume, follow this procedure:

• If the volume is not ENABLED, start it with this command:

  `vxvol start` *volume_name*

  If it is unstartable, set one of the plexes to CLEAN by using the command:

  `vxmend fix clean` *plex_name*

  and then start the volume.

• If the plex does not yet have a *kernel state* of ENABLED, use this command:

  `vxplex att` *volume_name plex_name* `...`

As with returning an OFFLINE plex to ACTIVE, this command recovers the contents of the plex(es), then sets the plex state to ACTIVE.

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).

If the volume is currently ENABLED, use this command:

`vxplex att` *volume_name plex_name ...*

For example, the command for a plex named `vol01-02` on a volume named `vol01` is:

`vxplex att vol01 vol01-02`

This command starts to recover the contents of the plex and, after the revive is complete, sets the plex utility state to ACTIVE.

If the volume is not in use (not ENABLED), use this command:

`vxmend on` *plex_name*

For example, the command for a plex named `vol01-02` is:

`vxmend on vol01-02`

In this case, the state of vol01-02 is set to STALE. When the volume is next started, the data on the plex is revived from the other plex, and incorporated into the volume with its state set to ACTIVE.

If it becomes necessary to manually change the state of a plex, refer to "Recovering a Volume." in "Volume Tasks." See the vxmake(1M) and vxmend(1M) manual pages for more information about these commands.

Moving Plexes

Moving a plex copies the data content from the original plex onto a new plex. The command to move data from one plex to another is:

vxplex mv *original_plex new_plex*

For a move task 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 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, you must use the -o force option.
- If the new plex is longer or less sparse than the original plex, the data that exists on the original plex is copied onto the new plex. Any area that was not on the original plex, but is represented on the new plex, is filled from other complete plex(es) 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 is not initialized and remains unused.

Copying Plexes

This task copies the contents of a volume onto a specified plex. The volume to be copied cannot be enabled. The plex cannot be associated with any other volume. To copy a plex, use this command:

vxplex cp *volume_name new_plex*

After the copy task is complete, *new_plex* is not associated with the specified volume *volume_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 If the plex being copied is larger than the volume, an incomplete copy of the data results. For the same reason, *new_plex* should not be sparse.

Volume Tasks

A *volume* is a collection of up to 32 plexes. A volume appears as a block device in the `/dev/vx/dsk` directory and a character device in the `/dev/vx/rdsk` directory. A volume can be used as a partition device.

The following sections describe how to perform common volume tasks. In some cases, either `vxassist` or other commands can be used to do the same task and both are described. Detailed descriptions about the commands used to perform volume tasks are contained in the Volume Manager manual pages.

Note: This section provides information on general volume tasks. For details on tasks specific to RAID-5 volumes, refer to "Performing Online Backup." in "Volume Tasks."

Creating Volumes

Volumes can be created with the `vxassist` or `vxmake` commands. The easiest way to create volumes is by using the `vxassist` command.

You can specify the length of a new volume in sectors, kilobytes, megabytes, or gigabytes. The unit of size is added as a suffix to the length (`s`, `m`, `k`, or `g`). If no unit is specified, sectors are assumed.

`vxassist` Command

You can use the `vxassist` command to create volumes with default settings or with user-specified attributes. `vxassist` automatically creates and attempts to enable new volumes. If the volume fails to be enabled, `vxassist` attempts to remove it and releases the space used to allocate that volume.

To create a simple volume using `vxassist` and its default settings, use this command:

`vxassist make` *volume_name length*

For example, the following command creates a volume named `voldef` with a length of 10 megabytes on any available disk(s):

`vxassist make voldef 10m`

You can specify additional parameters to `vxassist` to reflect the new volume attributes. Refer to the `vxassist`(1M) manual page for details.

The following example shows how to create a volume named `volzebra` that is striped across `disk03` and `disk04`, has the `fsgen` usage type, and is 10 megabytes long:

```
vxassist -Ufsgen make volzebra 10m layout=stripe disk03 \
 disk04
```

`vxmake` **Command**

For detailed information about how to use `vxmake`, and an example of the `vxmake` description files, refer to "Creating Volume Manager Objects With `vxmake`" in this chapter or the `vxmake`(1M) manual page.

To create a volume with an attached plex using `vxmake`, use this command:

`vxmake -U`*usage_type* `vol` *volume_name* `len=`*length* `plex=`*plex_name,...*

If you do not specify a length, the volume length is equal to the length of the plex to which it is attached. You can select a length (less than or equal to the length of the plex) by specifying a length with the `len=` parameter.

You can also create a volume without attaching a plex to it. You do this by omitting the `plex=` parameter. In this case, you must specify a length. The volume you create is not available for use until you attach a plex to it by using `vxplex att`.

Examples of commands for creating an `fsgen`-type volume called `vol01` are:

```
vxmake -Ufsgen vol vol01 len=100000
```

and

```
vxmake vol vol01 use_type=fsgen plex=vol01-01,vol01-2
```

You can specify the usage type for a volume as either `-Ufsgen` or `use_type=fsgen`. If you do not specify a length, or associated plexes are not identified, the length is 0.

Instead of specifying parameters on the command line, you can use a `vxmake` description file to create a volume, as well as associated subdisks and plexes, by using this command:

```
vxmake -d description_file
```

Initializing Volumes

During normal system operation, volume and plex states are affected by system failures, shutdowns, and possible I/O failures. When a volume is first created, it is necessary to initialize the state of its plex(es) according the state of the data on each plex.

Normally, if you have created the volume using `vxassist` or one of the other higher-level interfaces, the state of the plexes is properly set. However, when `vxmake` has been used to create a volume, the states of its plexes must be set manually before the volume can be made available for use through the `vxvol start` command. The command for setting the state of a volume plex is:

```
vxvol init  state volume_name [plex_name]
```

where the *state* 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 to CLEAN. The following examples show how to do this for mirrored and nonmirrored volumes. In the simplest case, in which a volume has been created containing only one plex (mirror), the state of the plex is set to CLEAN. This is because there is no need for any synchronization of the data on the disk. Since there is only one plex in the volume, it is not necessary to specify the *plex_name* argument.

The command to set the state of this volume to CLEAN is:

```
vxvol init clean volume_name
```

Note: The rest of this initialization section applies only to mirrored volumes (not RAID-5 volumes, which are discussed later).

For a newly-created mirror volume vol01, that has multiple plexes associated with it, one of the plexes must be chosen to be synchronized with the other plexes. For example, plex vol01-02 may have been created over disk space that contained data that needed to be accessed after the volume is made available. The following command synchronizes data with the other plexes when the volume is started:

```
vxvol init clean vol01 vol01-02
```

This command sets the state of vol01-02 to CLEAN and the remainder of the plexes to STALE, so they are properly synchronized at the time the volume is made available. You can avoid the initial synchronization of the volume to save time, but only if you are certain that none of the plexes contain data that is the final contents of the volume. In this case, you can temporarily initialize the state of the volume so that the data can be loaded without having to perform a synchronization first. The command to do this is:

```
vxvol init enable volume_name
```

This enables the volume and all its plexes, but leaves the plex utility states set to EMPTY. After the entire contents of the volume have been restored, both mirrors contain exactly the same data and do not need to be synchronized using the vxvol start task. The volume (for example, home1) can be initialized for use and started at the same time by using this command:

```
vxvol init active home1
```

Note: It is strongly recommended that data on each of the mirrors be exactly the same. Otherwise, the system can corrupt the data on both mirrors and, depending on the use of the volume, can crash the system. If you are not sure that the data is identical, use the vxvol init clean method.

Sometimes it is necessary to remove all existing data from disks before new data is loaded. In this case, you can initialize every byte of the volume to zero with this command:

```
vxvol init zero volume_name
```

Estimating Maximum Volume Size

The `vxassist` command can determine the largest possible size for a volume that can currently be created with a given set of attributes. `vxassist` can also determine how much an existing volume can be extended under the current conditions.

To determine the largest possible size for the volume to be created, use the command:

```
vxassist maxsize attributes...
```

For example, to determine the maximum size for a new RAID-5 volume on available disks, enter:

```
vxassist maxsize layout=raid5
```

This does not actually create the volume, but returns an output similar to this example (in sectors, by default):

```
Maximum volume size: 376832 (184Mb)
```

If, however, a volume with the specified attributes cannot be created, an error message is returned similar to this example:

```
vxvm:vxassist: ERROR: No volume can be created within the given
constraints
```

To determine how much an existing volume can grow, use the command:

```
vxassist maxgrow volume_name
```

For example, the command:

```
vxassist maxgrow raidvol
```

does not actually resize the volume, but results in output similar to this example:

```
Volume raidvol can be extended by 366592 to 1677312 (819Mb)
```

Notice that this output indicates both the amount *by* which the volume can be increased and the total size *to* which the volume can grow (in sectors, by default).

Displaying Volume Information

You can list information related to volumes under Volume Manager 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 this command:

```
vxprint -vt
```

To limit the `vxprint` display to a single object, you can specify the object name after the `vxprint` command.

For example, to display information about a specific volume, use this command:

```
vxprint -l volume_name
```

You can display detailed information for all volumes with this command:

```
vxprint -vl
```

To display the plexes for a volume named `vol01`, use this command:

```
vxprint -ht vol01
```

This command displays the volume, its plexes, and the subdisks in those plexes.

You can also display the plexes for `vol01` with the command:

```
vxprint -e 'assoc= "vol01"'
```

Changing Volume Attributes

You can change volume attributes such as read policy, error policies, ownership, permissions, and the values in the comment and utility fields for existing volumes. These attributes are typically changed when use of the volume or your needs change.

CAUTION! Change volume attributes with extreme care and only if necessary.

There are two Volume Manager commands that set volume attributes:

• The vxedit command sets attributes that are not usage-type dependent.

• The vxvol command sets attributes that are usage-type dependent.

Use the vxedit command as follows:

vxedit set *field=value* . . . *volume_name* . . .

Use the vxvol command as follows:

vxvol set *field=value* . . . *volume_name* . . .

Table 2 details which attributes can be set by each command.

Table 2 Setting Volume Attributes

Command	Attribute	Description
vxedit	comment	the comment field
	tutil0, tutil1, tutil2 putil0, putil1, putil2	descriptive string of volume contents
	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

Table 2 Setting Volume Attributes

Command	Attribute	Description
vxvol	len	numeric length of volume
	logtype	(drl/undef) specifier of Dirty Region Logging mode for volume
	loglen	length of the Dirty Region Logging log
	startopts	options to be executed to the vxvol start task

Note: Setting volume device permissions and ownership using the chgrp, chown, and chmod commands does not work. It is necessary to use vxedit set to modify these values.

For example, to change the owner of the group to susan and the permissions to read/write for owner, group, and other:

```
vxedit set user=susan group=staff mode=0666 vol01
```

Online Relayout

The vxassist command has been enhanced to include a new option named *Online Relayout.* Relayout is used to change the configuration of a volume. Relayout allows you to transform the storage characteristics of a volume without interrupting data access to the volume.

Transforming the storage characteristics of a volume is done by using the existing storage and by using some additional temporary space. If a transformation requires additional storage, the storage is automatically allocated.

In general, Relayout can be used to change the performance and/or reliability of a volume, *online.*

How Online Relayout Works

Online relayout is a process where storage in a volume or a plex is converted from one layout to another while maintaining data availability at all times. You can convert any supported layout in the Volume Manager to another layout, and/or change the stripe width or the number of columns. You can also do just stripe width and/or column changes.

Transformation of a volume or a plex must be initiated using the vxassist command. vxassist creates the necessary infrastructure and storage needed to perform the layout transformation and invokes vxrelayout internally to perform the transformation.

vxrelayout performs the layout transformation by copying a chunk of data to a temporary location. The chunk size is determined by vxassist. vxrelayout then transforms the storage associated with the data to the desired layout, and finally copies the data from the temporary location to destination layout. This is done until all the storage and data is converted to the desired layout.

Note that the data in the volume is always available during the transformation. Since vxrelayout depends on vxassist to do the setup before a layout transformation can be started, the vxassist command must be used to initiate any layout transformation. The vxtask command can be used to monitor the state of an ongoing transformation.

Since these transformations are time consuming, a transformation can be interrupted. The interruption can be caused by a system crash, I/O failure, or the user voluntarily stopping the transformation process. Using kill -9 is not advisable to stop transformations. The vxrelayout command can be used to continue or reverse a transformation process stopped because of interruptions.

Volume Manager reacts to interruptions in the following ways:

- I/O failure—The vxrelayout command exits with an error on stderr. The vxrelocd daemon tries to relocate the failed subdisks. If it succeeds, it invokes vxrecover which in turn invokes vxrelayout to continue the transformation. An I/O failure should be transparent to the user if the relocation succeeds. If the relocation fails, then the transformation process is not restarted again.

- System crash—On a reboot after a system crash, `vxrecover` tries to start all startable volumes, causing `vxrelayout` to be invoked on appropriate volumes and plexes.

If you stop `vxrelayout`, the transformation process can be reversed or continued by invoking `vxrelayout` manually.

When to Use Relayout

There are a number of situations where you may need to use Relayout. Here are a few examples of situations where using Relayout is appropriate.

- You have a RAID-5 volume that does not perform well. You want to convert it to a mirrored volume, and you can sacrifice additional storage.

- You want to change stripe widths as the current `stwidths` are incorrect.

- You want to add more columns to a RAID-5 or striped volume to improve performance.

- You have a mirrored volume and you can sacrifice performance, but you want to use less space with the same redundancy. The mirrored volume can be converted to RAID-5 by using Relayout.

- A plex in a simple mirrored volume needs to be changed to striped.

How to Use Relayout on the Command Line

This section describes how to operate Online Relayout from the command line. Refer to the `vxassist`(1M) manual page for the usage of `vxassist relayout`. This document provides only basic information. You can also perform Online Relayout tasks by using the Storage Administrator.

Using Command `vxassist`

You can use the `vxassist` command to operate Relayout. The command syntax is as follows:

```
vxassist relayout volume_name | plex_name layout=layout \
ncol=[+/-]ncol stwidth=stdwidth tmpsize=tmpsize
```

1. *volume_name | plex_name* specifies the name of the volume/plex that is to be transformed.

2. *layout* specifies the new layout desired.

3. *ncol* specifies the number of columns in the new layout. The `+/-` operator can be used to specify the number of columns to be added to or subtracted from the original volume/plex.

4. *stwidth* specifies the stripe width of the new layout.

5. *tmpsize* specifies the size of the scratch space (this is optional). `vxassist` chooses the volume size as the *tmpsize* for volumes less than 50M. For volumes greater than 50M, `vxassist` uses 50M. For volumes greater than 1G, the *tmpsize* is 1G. You can override the default values by specifying a value for this parameter.

Parameter 1 (*volume_name* | *plex_name)* is mandatory.

Parameters 2, 3, and 4, can be specified in any combination. At least one of these parameters must be specified for a volume to be considered a candidate for Relayout. If only the *layout* parameter is specified and the number of mirrors is increased or decreased as a result of the specification, then a Relayout operation does not occur.

For example, if there is a stripe-mirror volume named `vol`, and you specify:

```
 vxassist relayout vol layout=stripe
```

this specification means to remove a mirror, which is not considered a Relayout operation. For a volume to be considered a candidate for a Relayout operation, one of the following must be part of the reconfiguration:

• a change in number of columns

• a change of stripe-width

• removal or addition for parity

• RAID-5 to mirroring or the reverse operation

Note: `vxassist` also recognizes storage attributes that can be used in the relayout operation. Refer to "Using `vxassist`" in this chapter for more information on `vxassist`. For details on available `vxassist` keywords and attributes, refer to the `vxassist`(1M) manual page.

☰ 4

Examples

This section shows some examples of how to use the Command Line Interface to do Relayout operations.

The basic layouts supported are:

- RAID-5
- stripe
- mirror
- nostripe
- stripe-mirror, mirror-stripe
- concat-mirror, mirror-concat

The basic layouts are described in the *VERITAS Volume Manager Getting Started Guide.*

Changing the Number of Columns

This section describes how to use Relayout to change columns.

To add a column to a volume, use this command:

```
vxassist relayout volume_name ncol=+1
```

To remove a column, use this command:

```
vxassist relayout volume_name ncol=-1
```

To specify a new column number to Relayout, use this command:

```
vxassist relayout volume_name ncol=new_ncol
```

To specify a volume so that the final volume has 5 columns, use this command:

```
 vxassist relayout volume_name ncol=5
```

Changing the Stripe-Width

To change the stripe-width, use this command:

```
vxassist relayout volume_name stwidth=new_stwidth
```

Changing the Stripe-Width and Column

The attributes `ncol` and `stwidth` can also be used in combination. To relayout a volume to result in a stripe-width of 64K and 5 columns, use this command:

```
vxassist relayout volume_name stwidth=64K ncol=5
```

Changing the Layouts

This section describes how to use Relayout to change layout configurations.

RAID-5 to Mirroring

To change a RAID-5 volume to a stripe-mirrored volume, use this command:

```
vxassist relayout volume_name layout=stripe-mirror
```

The stripe-mirror and mirror-stripe layouts are treated the same by Relayout. If you do not specify `nmirror=` *number greater than 1*, then Relayout generates a stripe-mirror. If you specify `nmirror=2`(or greater), Relayout generates the mirror-stripe layout that you specified.

If the stripe-width is not specified during these changes, a default value is chosen that may not be the same as the original volume. This is done for performance reasons. You can use the `vxassist convert` option to convert a stripe-mirror to a mirror-stripe. You can also use the `convert` option for a mirror-concat layout.

Any Layout to RAID-5

To change any layout to RAID-5, use this command:

```
vxassist relayout volume_name layout=raid5
```

In general, you can use the layout field to create any layout desired. This can be used in combination with `ncol` and `stwidth` to exactly specify the final volume characteristics.

For example, to convert a striped volume to a concatenated volume, use this command:

```
vxassist relayout volume_name layout=nostripe
```

▼ 4

Starting, Stopping, and Reversing a Relayout

Relayout is a transactional operation, so it can be stopped, started, and reversed at anytime without loss of data or data availability. The vxtask command can be used to stop a Relayout operation.

A Relayout operation can be restarted on a volume with this command:

    vxrelayout start *volume_name*

A Relayout operation can be reversed by first stopping any ongoing Relayout operation and then doing the following:

    vxrelayout reverse *volume_name*

It is advised that transformation direction changes be done with care.

Online Relayout can be applied only to disk groups with version 50 or above (see the vxdg(1M) manual page) and only one disk group at a time.

The vxrelayout command takes the following form:

    vxrelayout [-f] [-g *diskgroup*] [-U *usagetype*]
    [-o *useopt*] start *volume*

    vxrelayout [-f] [-g *diskgroup*] [-U *usagetype*]
    [-o *useopt*] reverse *volume*

    vxrelayout status volume

The following are the recognized operation keywords:

start   Continue a discontinued layout conversion. This operation determines the current state and direction of a layout transformation and continues from where it was last stopped. Any existing layout transformation process must be stopped before this operation is applied.

reverse   Reverse a discontinued layout conversion. This operation determines the current state and direction of a layout transformation and reverse the transformation from where it was last stopped. Any existing layout transformation process must be stopped before this operation is applied.

status   Display the status of a discontinued or an ongoing layout
          conversion. This operation displays the characteristics of the source
          and the destination layout. The display includes the layout, number
          of columns, and stripe width. It also displays whether the
          transformation is ongoing or stopped and the percentage completed.

The first operand is a keyword that determines whether a layout conversion is
to be continued, reversed, or to display the status of any existing
transformation in the volume. For example, if a RAID-5 volume is being
converted to a striped volume, reversal of such an operation reverts all data
and storage in striped format to the RAID-5 format. A reversal of a reversal is
equivalent to a continue.

The direction of a transformation can be determined by using the status
operation in the `vxrelayout` utility. This command determines if any plex in
the volume is being transformed and displays its characteristics.

The second operand is the name of the topmost volume to which the plex
undergoing relayout is attached. If there are multiple plexes in the volume, the
`vxrelayout` utility selects the appropriate plex using the supplied volume
name. It should be noted that there can only be one relayout operation
happening in a volume at any instant of time. A specific disk group can be
specified with `-g diskgroup`.

Finding the Status of a Relayout Operation

Relayout operations can be time consuming. It is desirable to determine the
progress rate of such operations. It is also useful to know what layout is being
transformed to what layout.

The rate of progress can be found by using the `vxtask` utility. The
characteristics of the storage transformation can be determined by using the
following command:

`vxrelayout status` *volume_name*

This provides the storage characteristics and a rough estimate of the
percentage completed. Note that the second piece of information is not as
accurate as the one provided by the `vxtask` utility. For example, if you are
adding a column to a striped volume and you check the status with this
command:

`vxrelayout status` *volume_name*

---

 you will get:

```
STRIPED, columns=5, stwidth=128 --> STRIPED, columns=6, stwidth=128
Relayout running, 0.00% completed. .
```

Transformation Details

You can use Online Relayout on one data plex in a volume or on the entire volume. You cannot invoke multiple Relayout operations on multiple plexes in a volume. There can only be one Relayout operation at a time in a volume.

Converting a Plex in a Mirrored Volume to a RAID-5 Plex

If there is more than one plex in the mirrored volume to be converted to RAID-5, you must specify which plex is to be converted. The specified plex is then converted to a RAID-5 plex. Because RAID-5 mirroring is not supported in Volume Manager, all other plexes in the volume are removed at the end of the conversion. The space previously occupied by the removed plexes is not reused. A mirrored volume can be converted to a layout other than RAID-5, which does not result in removal of nonconverted plexes.

Logs

A Relayout operation can render existing logs incompatible with the new layout as a result of the transformation. For example, when a mirrored volume with a DRL log is converted to a RAID-5 volume with RAID log, the DRL log is removed and the RAID-5 log is used. If you specified that the RAID-5 volume should not have a log, then the DRL log is also removed.

Menu Interface Tasks

Introduction

This chapter describes the menu-driven interface known as Volume Manager Support Operations (or `vxdiskadm`). `vxdiskadm` is used to perform physical and logical device administration.

This chapter provides the following information on performing disk and volume tasks using `vxdiskadm`:

- Starting `vxdiskadm`
  - The `vxdiskadm` Main Menu
  - The `vxdiskadm` Main Menu Description
- Disk Tasks
  - Bringing Physical Disks Under Volume Manager Control
  - Adding a Disk for Future Use
  - Reinitializing a Disk for Volume Manager Use
  - Adding a VM Disk to the Hot-Relocation Pool
  - Removing a VM Disk From the Hot-Relocation Pool
  - Displaying Information on Physical Disks
  - Removing a Physical Disk
  - Disabling a Physical Disk (Taking a Physical Disk Offline)
  - Enabling a Physical Disk

- • Replacing a Physical Disk
    - • Replacing a Failed or Removed Disk
- • Volume Tasks
    - • Mirroring Volumes on a VM Disk
    - • Moving Volumes from a VM Disk
- • Disk Group Tasks
    - • Adding a Disk to a Disk Group
    - • Creating a Disk Group
    - • Deporting a Disk Group
    - • Importing a Disk Group
- • Exiting the Volume Manager Support Tasks

Starting `vxdiskadm`

To start vxdiskadm, enter:

```
vxdiskadm
```

vxdiskadm displays its main menu showing the tasks available to you.

The `vxdiskadm` **Main Menu**

This is the `vxdiskadm` main menu:

```
Volume Manager Support Operations
Menu: VolumeManager/Disk

1    Add or initialize one or more disks
2    Encapsulate one or more disks
3    Remove a disk
4    Remove a disk for replacement
5    Replace a failed or removed disk
6    Mirror volumes on a disk
7    Move volumes from a disk
8    Enable access to (import) a disk group
```

```
9     Remove access to (deport) a disk group
10    Enable (online) a disk device
11    Disable (offline) a disk device
12    Mark a disk as a spare for a disk group
13    Turn off the spare flag on a disk
list List disk information


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


Select an operation to perform
```

- `?` can be entered at any time to provide help in using the menu. The output of `?` is a list of operations and a definition of each.

- `??` lists inputs that can be used at any prompt.

- `q` returns you to the main menu if you need to restart a process; however, using `q` at the main menu level exits the Volume Manager Support Operations.

The `vxdiskadm` **Main Menu Description**

The `vxdiskadm` menu provides access to the following tasks. The numbers correspond to the items listed in the main menu:

1. Add or initialize one or more disks.

   You can add formatted disks to the system. SCSI disks are already formatted. For other disks, see the manufacturer's documentation for formatting instructions. You are prompted for the disk device(s). You can specify the disk group to which the disk(s) should be added; if `none` is selected, the disk is held as a spare to be used for future operations or disk replacements without needing to be initialized at that time. You can also specify that selected disks be marked as hot-relocation spares for a disk group. If the disk has not been initialized already, the disk is partitioned and initialized for use with the Volume Manager.

2. Encapsulate one or more disks.

You can bring a disk that was added to the system before installing the Volume Manager under Volume Manager control. You are prompted for the disk device(s), disk group, and disk name(s). The disk is added to the specified disk group. The disk is examined to search for partitions that are used for file systems or other purposes. Volumes are created to replace disk partitions as the means of accessing the existing data. If the encapsulation cannot determine the purpose of a partition automatically, you are asked what to do with it. You can choose to replace the partition with a volume, leave the partition alone, or remove the partition.

You must reboot the system if any partitions are being used for mounted file systems or for running applications. You may have to modify application configuration files to use volumes, rather than direct disk devices, to access the disk partitions. File system mount information is adjusted automatically.

3. Remove a disk.

   You can remove a disk from a disk group. You are prompted for the name of a disk to remove. You cannot remove a disk if any volumes use storage on that disk. If any volumes are using storage on the disk, you have the option of asking the Volume Manager to move that storage to other disks in the disk group.

Note: You cannot remove the last disk in a disk group using this task. If you wish to use all the remaining disks in a disk group for some purpose, you should disable (deport) the disk group. You will then be able to reuse the disks. `rootdg` cannot, however, be deported.

---

4. Remove a disk for replacement.

   You can remove a physical disk from a disk group, while retaining the disk name. This changes the state for the named disk to `removed`. If there are any initialized disks that are not part of a disk group, you are given the option of using one of these disks as a replacement.

5. Replace a failed or removed disk.

   You can specify a replacement disk for a disk that you removed with the `Remove a disk for replacement` menu entry, or one that failed during use. You are prompted for a disk name to replace and a disk device to use as a replacement. You can choose an uninitialized disk, in which case

the disk will be initialized, or you can choose a disk that you have already initialized using the `Add or initialize a disk` menu operation.

6. Mirror volumes on a disk

   You can mirror volumes on a disk. These volumes can be mirrored to another disk with available space. Creating mirror copies of volumes in this way protects against data loss in case of disk failure. Volumes that are already mirrored or that are comprised of more than one subdisk will not be mirrored with this task. Mirroring volumes from the boot disk will produce a disk that can be used as an alternate boot disk.

7. Move volumes from a disk.

   You can move any volumes (or parts of a volume) that are using a disk onto other disks. Use this menu task immediately prior to removing a disk, either permanently or for replacement.

Note: Simply moving volumes off a disk, without also removing the disk, does not prevent other volumes from being moved onto the disk by future operations.

---

8. Enable access to (import) a disk group.

   You can enable access by this system to a disk group. If you wish to move a disk group from one system to another, you must first disable (deport) it on the original system. Then, move the disks from the deported disk group to the other system and enable (import) the disk group there. You are prompted for the disk group name.

9. Disable access to (deport) a disk group

   You can 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.

   You are prompted for the name of a disk group. You are asked if the disks should be disabled (offlined). For removable disk devices on some systems, it is important to disable all access to the disk before removing the disk.

10. Enable (online) a disk device.

If you move a disk from one system to another during normal system operation, the Volume Manager will not recognize the disk automatically. Use this menu task to tell the Volume Manager to scan the disk to identify it, and to determine if this disk is part of a disk group. Also, use this task to re-enable access to a disk that was disabled by either the disk group deport task or the disk device disable (offline) operation.

11. Disable (offline) a disk device.

You can disable all access to a disk device through the Volume Manager. This task can be applied only to disks that are not currently in a disk group. Use this task if you intend to remove a disk from a system without rebooting.

Note that some systems do not support disks that can be removed from a system during normal operation. On those systems, the offline operation is seldom useful.

12. Mark a disk as a spare for a disk group.

You can reserve a disk as an automatic replacement disk (for hot-relocation) in case another disk in the disk group should fail.

13. Turn off the spare flag on a disk.

You can free hot-relocation spare disks for use as regular VM disks.

• List disk information

You can display a list of disks attached to your system. This also lists removed or failed disks.

You can also use this task to list detailed information for a particular disk. This information includes the disk group of which the disk is a member, even if that disk group is currently disabled.

Disk Tasks

This section describes the disk tasks available with `vxdiskadm`.

When performing disk administration, it is important that you recognize the difference between a *device name* and a *disk name.*

Note: Your system may use a *device name* that differs from the examples. See Chapter 1, "Understanding Volume Manager," in the *VERITAS Volume Manager Getting Started Guide,* for more information on device names.

The *device name* (sometimes referred to as *devname* or *disk access name*) is the location of the disk. The syntax of a device name is `c#b#t#d#s#`, where:

- `c#` is the number of the controller to which the disk drive is attached.
- `b#` is the corresponding bus number (if used on your system)
- `t#` is the number of the target disk on that controller.
- `d#` is the number of the disk.
- `s#` is the number of the disk slice.

The full pathname of a device is `/dev/vx/dmp/`*devicename.* In this document, only the device name is listed and `/dev/vx/dmp` is assumed. An example of a device name is `c0t0d0s2`.

The *disk name* (sometimes referred to as *disk media name*) is an administrative name for the disk, such as `disk01`. If you do not assign a disk name, the disk name defaults to `disk##` if the disk is being added to `rootdg` (where `##` is a sequence number). Otherwise, the default disk name is *groupname*`##`, where *groupname* is the name of the disk group to which the disk is added.

Bringing Physical Disks Under Volume Manager Control

When you add a disk to a system that is running Volume Manager, you need to put the disk under Volume Manager control so that the Volume Manager can control the space allocation on the disk.

Unless another disk group is specified, Volume Manager places new disks in the default disk group, `rootdg`. Instructions on creating additional disk groups are provided later in this chapter.

The method by which you place a disk under Volume Manager control depends on the circumstances:

- If the disk is new, it needs to be initialized and placed under Volume Manager control (see "Placing a Disk Under Volume Manager Control").

- If the disk is not needed immediately, it can be initialized (but not added to a disk group) and reserved for future use (see "Adding a Disk for Future Use").

- If the disk was previously initialized for future Volume Manager use, it can be reinitialized and placed under Volume Manager control (see "Reinitializing a Disk for Volume Manager Use").

- If the disk was previously in use, but not under Volume Manager control, you may wish to preserve existing data on the disk while still letting Volume Manager take control of the disk. This can be accomplished using encapsulation (see "Encapsulating a Disk for Volume Manager Use").

- Multiple disks on one or more controllers can be placed under Volume Manager control simultaneously. Depending on the circumstances, all of the disks may not be processed the same way. See "Placing Multiple Disks Under Volume Manager Control."

  When initializing or encapsulating multiple disks at once, it is possible to exclude certain disks or certain controllers. To exclude disks, list the names of the disks to be excluded in the file `/etc/vx/disks.exclude` before the initialization or encapsulation. Similarly, you can exclude all disks on specific controllers from initialization or encapsulation by listing those controllers in the file `/etc/vx/cntrls.exclude`.

The sections that follow provide detailed examples of how to use `vxdiskadm` to place disks under Volume Manager control in various ways and circumstances.

Note: A disk must be formatted (using the `format` command, for example) or added to the system (using `diskadd`) before it can be placed under Volume Manager control.

If you attempt to place an unformatted disk under Volume Manager control through `vxdiskadm`, the initialization begins as normal, but quits with a message that informs you that the disk does not appear to be valid and may not be formatted. If this happens, you need to format the disk properly and then attempt to place the disk under Volume Manager control again.

Placing a Disk Under Volume Manager Control

This section describes how to place a formatted disk under Volume Manager control. The disk can be new or previously used outside Volume Manager.

Note: Initialization does not preserve data on disks.

Initialize a single disk for Volume Manager use as follows:

1. Select menu item 1 (`Add or initialize one or more disks`) from the `vxdiskadm` main menu.

2. At the following prompt, enter the disk device name of the disk to be added to Volume Manager control (or enter `list` for a list of disks):

   ```
   Add or initialize disks
   Menu: VolumeManager/Disk/AddDisks


   Use this operation to add one or more disks to a disk group. You
   can add the selected disks to an existing disk group or to a new
   disk group that will be created as a part of the operation. The
   selected disks may also be added to a disk group as spares. The
   selected disks may also be initialized without adding them to a
   disk group leaving the disks available for use as replacement
   disks.

   More than one disk or pattern may be entered at the prompt. Here
   are some disk selection examples:
   ```

```
all:              all disks
c3 c4t2:          all disks on both controller 3 and controller
                  4,target 2
c3t4d0:           a single disk


Select disk devices to add:
[<pattern-list>,all,list,q,?] list
```

<pattern-list> can be a single disk, or a series of disks and/or controllers
(with optional targets). If *<pattern-list>* consists of multiple items, those
items must be separated by white space.

If you enter list at the prompt, vxdiskadm displays a list of the disks
available to the system, followed by a prompt at which you should enter
the device name of the disk to be added:

```
DEVICE          DISK           GROUP          STATUS
c0t0d0          disk01         rootdg         online
c0t1d0          disk02         rootdg         online
c0t2d0          disk03         rootdg         online
c0t3d0          –              –              online
c1t0d0          disk10         rootdg         online
c1t0d1          –              –              error
 .
 .
 .
c3t0d0          –              –              error
c3t1d0          disk33         rootdg         online
c3t2d0          disk34         rootdg         online
c3t3d0          disk35         rootdg         online


Select disk devices to add:
[<pattern-list>,all,list,q,?] c1t0d1
```

All disks attached to the system are recognized by the Volume Manager and
displayed here.

The word `error` in the `STATUS` line tells you that a disk has not yet been added to Volume Manager control. These disks may or may not have been initialized before. The disks that are listed with a disk name and disk group cannot be used for this task, as they are already under Volume Manager control.

3. To continue with the operation, enter `y` (or press Return) at the following prompt:

   ```
   Here is the disk selected.  Output format: [Device_Name]

    c1t2d0

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

4. At the following prompt, specify the disk group to which the disk should be added or press Return to accept `rootdg`:

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

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

5. At the following prompt, either press Return to accept the default disk name or enter a disk name:

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

6. When `vxdiskadm` asks whether this disk should become a hot-relocation spare, enter `n` (or press Return):

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

7. To continue with the operation, enter `y` (or press Return) at the following prompt:

   ```
   The selected disks will be added to the disk group rootdg with
   default disk names.

      c1t2d0

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

8. If you are sure that there is no data on this disk, enter n to avoid encapsulation. When vxdiskadm asks if you want to initialize the disk instead, enter y:

```
The following disk device has a valid VTOC, but does not appear
to have been initialized for the Volume Manager.  If there is
data on the disk that should NOT be destroyed you should
encapsulate the existing disk partitions as volumes instead of
adding the disk as a new disk.
Output format: [Device_Name]

   c1t2d0

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

   c1t2d0

Instead of encapsulating, initialize? [y,n,q,?] (default: n) y
```

Messages similar to the following should now confirm that disk c1t2d0 is being placed under Volume Manager control. On some systems, you are also given the option of performing surface analysis.

```
Initializing device c1t2d0.

Perform surface analysis (highly recommended)
[y,n,q,?] (default: y) n

Adding disk device c1t2d0 to disk group rootdg with disk
name disk39.
```

9. At the following prompt, indicate whether you want to continue to initialize more disks (y) or return to the vxdiskadm main menu (n):

```
Add or initialize other disks? [y,n,q,?] (default: n)
```

Placing Multiple Disks Under Volume Manager Control

This section describes how to place multiple disks under Volume Manager control simultaneously. The set of disks can consist of all disks on the system, all disks on a controller, selected disks, or a combination thereof.

Depending on the circumstances, all of the disks may not be processed the same way. For example, some may be initialized, while others may be encapsulated.

Note: Initialization does not preserve data on disks.

When initializing or encapsulating multiple disks at one time, it is possible to exclude certain disks or certain controllers. To exclude disks, list the names of the disks to be excluded in the file /etc/vx/disks.exclude before the initialization or encapsulation. You can exclude all disks on specific controllers from initialization or encapsulation by listing those controllers in the file /etc/vx/cntrls.exclude.

Place multiple disks under Volume Manager control at one time as follows:

1. Select menu item 1 (Add or initialize one or more disks) from the vxdiskadm main menu.

2. At the following prompt, enter the *pattern-list* for the disks to be added to Volume Manager control. In this case, enter c3 to indicate all disks on controller 3:

```
Add or initialize disks
Menu: VolumeManager/Disk/AddDisks

  Use this operation to add one or more disks to a disk group.
  You can add the selected disks to an existing disk group or to
  a new disk group that will be created as a part of the
  operation. The selected disks may also be added to a disk
  group as spares. The selected disks may also be initialized
  without adding them to a disk group leaving the disks
  available for use as replacement disks.

  More than one disk or pattern may be entered at the prompt.
  Here are some disk selection examples:

  all:              all disks
  c3 c4t2:          all disks on both controller 3 and
                    controller 4, target 2
  c3t4d0:           a single disk

  Select disk devices to add:
  [<pattern-list>,all,list,q,?] c3
```

where *<pattern-list>* can be a single disk, or a series of disks and/or controllers (with optional targets). If *<pattern-list>* consists of multiple items, those items must be separated by white space.

If you do not know the address (device name) of the disk to be added, enter
list at the prompt for a complete listing of available disks.

3. To continue the operation, enter y (or press Return) at the following
prompt:

```
Here are the disks selected.  Output format: [Device_Name]

  c3t0d0 c3t1d0 c3t2d0 c3t3d0
```

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

4. To add these disks to the default disk group, rootdg, enter y (or press
Return) at the following prompt:

```
You can choose to add these disks to an existing disk group, a
new disk group, or you can leave these disks 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 disks available for future use, specify a disk group
name of "none".
```

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

5. To allow vxdiskadm to use default disk names for each of the disks, enter
y (or Press Return) at the following prompt:

```
Use default disk names for these disks? [y,n,q,?] (default: y) y
```

6. At the following prompt, enter n to indicate that these disks should not be
used as hot-relocation spares:

```
Add disks as spare disks for rootdg? [y,n,q,?] (default: n) n
```

7. To continue with the operation, enter y (or press Return) at the following
prompt:

```
The selected disks will be added to the disk group rootdg with
default disk names.


c3t0d0 c3t1d0 c3t2d0 c3t3d0


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

8. The following prompt lists any disks that have already been initialized for Volume Manager use; enter Y to indicate that all of these disks should be used now:

```
The following disk devices appear to have been initialized
already.
The disks are currently available as replacement disks.
Output format: [Device_Name]

c3t1d0 c3t2d0 c3t3d0

Use these devices? [Y,N,S(elect),q,?] (default: Y) Y
```

Note that this prompt allows you to indicate "yes" or "no" for *all* of these disks (Y or N) or to select how to process each of these disks on an individual basis (S).

If you are sure that you want to reinitialize all of these disks, enter Y at the following prompt:

```
The following disks you selected for use appear to already have
been initialized for the Volume Manager.  If you are certain the
disks already have been initialized for the Volume Manager, then
you do not need to reinitialize these disk devices.
Output format: [Device_Name]


c3t1d0 c3t2d0 c3t3d0


Reinitialize these devices? [Y,N,S(elect),q,?] (default: Y) Y
```

9. vxdiskadm now indicates that one of the disks on controller 3 is a candidate for encapsulation; enter y (or press Return) to encapsulate this disk:

```
The following disk device has a valid VTOC, but does not appear
to have been initialized for the Volume Manager. If there is
data on the disk that should NOT be destroyed you should
encapsulate the existing disk partitions as volumes instead of
adding the disk as a new disk. Output format: [Device_Name]
c3t0d0

Encapsulate this device? [y,n,q,?] (default: y) y
```

Encapsulation allows you to add an active disk to Volume Manager control and preserve the data on that disk.

Note: Disk encapsulation requires that you reboot the system and may require a few subsequent reboots. You will be prompted for these operations, as necessary.

`vxdiskadm` now confirms those disks that are being initialized and added to Volume Manager control with messages similar to the following:

```
Initializing device c3t1d0.

Initializing device c3t2d0.

Initializing device c3t3d0.

Adding disk device c3t1d0 to disk group rootdg with disk
name disk33.


Adding disk device c3t2d0 to disk group rootdg with disk
name disk34.


Adding disk device c3t3d0 to disk group rootdg with disk
name disk35.
```

In addition to the output displayed above, you may see prompts that give you the option of performing surface analysis.

10. `vxdiskadm` then confirms any disks that have been selected for encapsulation and prompts you for permission to proceed with the encapsulation; enter `y` (or press Return) to continue encapsulation:

```
The following disk has been selected for encapsulation.
Output format: [Device_Name]

c3t0d0

Continue with encapsulation? [y,n,q,?] (default: y) y
```

`vxdiskadm` now displays an encapsulation status and informs you that you must perform a shutdown and reboot as soon as possible:

```
The disk device c3t0d0 will be encapsulated and added to the
disk group rootdg with the disk name disk38.
```

```
The first stage of encapsulation has completed successfully.
You should now reboot your system at the earliest possible
opportunity.

The encapsulation will require two or three reboots which will
happen automatically after the next reboot.  To reboot execute
the command:

   shutdown -g0 -y -i6

This will update the /etc/vfstab file so that volume devices are
used to mount the file systems on this disk device.  You will
need to update any other references such as backup scripts,
databases, or manually created swap devices.
```

11. At the following prompt, indicate whether you want to continue to initialize more disks (y) or return to the vxdiskadm main menu (n):

```
Add or initialize other disks? [y,n,q,?] (default: n)
```

Encapsulating a Disk for Volume Manager Use

This section describes how to encapsulate a disk for Volume Manager use. Encapsulation preserves any existing data on the disk when the disk is placed under Volume Manager control.

To reduce the chance of encapsulation failure, make sure that the disk:

- has a small amount of free space (at the beginning or end of the disk) that does not belong to any partition
- has two free partitions
- has an s2 (or s0 on some systems) slice that represents the whole disk

On some systems, when encapsulating the boot (root) disk, the swap partition should be tagged as swap so that it's possible to dump to that partition later.

Before encapsulating your boot disk, the EEPROM variable use-nvramrc? must be set to true if you want to take advantage of the Volume Manager boot disk aliases to identify the mirror of the boot disk if a replacement is needed. If this variable is set to false, you need to determine which disks are bootable yourself. You can set this variable to true as follows:

```
eeprom use-nvramrc?=true
```

Encapsulate a disk for Volume Manager use as follows:

1. Select menu item 2 (`Encapsulate one or more disks`) from the
   `vxdiskadm` main menu.

Note: Your system may use a *device name* that differs from the examples. See
Chapter 1, "Understanding Volume Manager," in the *VERITAS Volume Manager
Getting Started Guide,* for more information on device names.

---

2. At the following prompt, enter the disk device name for the disks to be
   encapsulated:

```
Encapsulate one or more disks
Menu: VolumeManager/Disk/Encapsulate

  Use this operation to convert one or more disks to use the
  Volume Manager.This adds the disks to a disk group and
  replaces existing partitions with volumes.  Disk encapsulation
  requires a reboot for the changes to take effect.

  More than one disk or pattern may be entered at the prompt.
  Here are some disk selection examples:

  all:              all disks
  c3 c4t2:          all disks on both controller 3 and controller
                    4,target 2
  c3t4d0:           a single disk

Select disk devices to encapsulate:
[<pattern-list>,all,list,q,?] c2t5d0
```

Where *<pattern-list>* can be a single disk, or a series of disks and/or
controllers (with optional targets). If *<pattern-list>* consists of multiple
items, those items must be separated by white space.

If you do not know the address (device name) of the disk to be encapsulated, enter `l` or `list` at the prompt for a complete listing of available disks.

3. To continue the operation, enter `y` (or press Return) at the following prompt:

```
Here is the disk selected.  Output format: [Device_Name]

c2t5d0
```

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

4. To add the disk to the default disk group, `rootdg`, press Return at the following prompt:

```
You can choose to add this disk to an existing disk group or to
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)

5. At the following prompt, either press Return to accept the default disk name or enter a disk name:

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

6. To continue with the operation, enter `y` (or press Return) at the following prompt:

```
The selected disks will be encapsulated and added to the rootdg
disk group with default disk names.

c2t5d0
```

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

7. To confirm that encapsulation should proceed, enter `y` (or press Return) at the following prompt:

```
The following disk has been selected for encapsulation.  Output
format: [Device_Name]

c2t5d0
```

Continue with encapsulation? [y,n,q,?] (default: y) **y**

A message similar to the following confirms that the disk is being encapsulated for Volume Manager use and tells you that a reboot is needed:

```
The disk device c2t5d0 will be encapsulated and added to the
   disk group rootdg with the disk name disk01.
   The c2t5d0 disk has been configured for encapsulation.


   The first stage of encapsulation has completed successfully.
   You should now reboot your system at the earliest possible
   opportunity.
   The encapsulation will require two or three reboots which
   will happen automatically after the next reboot.  To reboot
   execute the command:

shutdown -g0 -y -i6

   This will update the /etc/vfstab file so that volume devices
   are used to mount the file systems on this disk device.  You
   will need to update any other references such as backup
   scripts, databases, or manually created swap devices.
```

8. At the following prompt, indicate whether you want to encapsulate more disks (y) or return to the vxdiskadm main menu (n):

```
Encapsulate other disks? [y,n,q,?] (default: n) n
```

Under some circumstances, encapsulation of a disk can fail. Encapsulation often fails because there is not enough free space available on the disk to accommodate Volume Manager. If this happens, the procedure outlined above ends abruptly with an error message similar to the following:

```
The disk device c2t5d0 will be encapsulated and added to the
disk group rootdg with the disk name disk01.


The encapsulation operation failed with the following error:


It is not possible to encapsulate c2t5d0, for the following
reason:
   <vxvm:vxslicer: ERROR: Unsupported disk layout.>
Hit RETURN to continue.
```

Adding a Disk for Future Use

If you want to add a disk to Volume Manager control for future use, follow the steps outlined in "Bringing Physical Disks Under Volume Manager Control." However, when you are asked to name a disk group, enter `none` instead of selecting `rootdg` or typing in a disk group name. The disk is then initialized as before, but is reserved for use at a later time. It cannot be used until it is added to a disk group.

Note that this type of "spare disk" should not be confused with a hot-relocation spare disk.

Reinitializing a Disk for Volume Manager Use

This section describes how to reinitialize a disk that has previously been initialized for Volume Manager use.

If the disk you want to add has been used before, but not with Volume Manager, you can do *either* of the following:

- Encapsulate the disk and preserve its information (see "Encapsulating a Disk for Volume Manager Use").

- Reinitialize the disk, allowing the Volume Manager to configure the disk for Volume Manager. Note that reinitialization does not preserve data on the disk. If you wish to have the disk reinitialized, make sure that the disk does not contain data that should be preserved.

Reinitialize a disk for Volume Manager use as follows:

1. Select menu item 1 (`Add or initialize one or more disks`) from the `vxdiskadm` main menu.

2. At the following prompt, enter the disk device name of the disk to be added to Volume Manager control:

```
Add or initialize disks
Menu: VolumeManager/Disk/AddDisks

  Use this operation to add one or more disks to a disk group.
  You can add the selected disks to an existing disk group or to
  a new disk group that will be created as a part of the
  operation. The selected disks may also be added to a disk group
```

```
as spares. The selected disks may also be initialized without
adding them to a disk group leaving the disks available for use
as replacement disks.

More than one disk or pattern may be entered at the prompt. Here
are some disk selection examples:

all:              all disks
c3 c4t2:          all disks on both controller 3 and controller
                  4, target 2
c3t4d0:           a single disk
```

```
Select disk devices to add:
[<pattern-list>,all,list,q,?] c1t3d0
```

Where *<pattern-list>* can be a single disk, or a series of disks and/or
controllers (with optional targets). If *<pattern-list>* consists of multiple
items, those items must be separated by white space.

If you do not know the address (device name) of the disk to be added, enter
l or list at the prompt for a complete listing of available disks.

3. To continue with the operation, enter y (or press Return) at the following
   prompt:

   ```
   Here is the disk selected.  Output format: [Device_Name]

   c1t3d0
   ```

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

4. At the following prompt, specify the disk group to which the disk should be
   added or press Return to accept rootdg:

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

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

5. At the following prompt, either press Return to accept the default disk name or enter a disk name:

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

6. When `vxdiskadm` asks whether this disk should become a hot-relocation spare, enter n (or press Return):

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

7. To continue with the operation, enter y (or press Return) at the following prompt:

    ```
    The selected disks will be added to the disk group rootdg with
    default disk names.

    c1t3d0

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

8. The following prompt indicates that this disk has been previously initialized for future Volume Manager use; enter y to confirm that you now want to use this disk:

    ```
    The following disk device appears to have been initialized
    already. The disk is currently available as a replacement disk.
    Output format: [Device_Name]

    c1t3d0

    Use this device? [y,n,q,?] (default: y) y
    ```

9. To reinitialize the disk, enter y (or press Return) at the following prompt:

    ```
    The following disk you selected for use appears to already have
    been initialized for the Volume Manager.  If you are certain the
    disk has already been initialized for the Volume Manager, then
    you do not need to reinitialize the disk device.
    Output format: [Device_Name]

    c1t3d0

    Reinitialize this device? [y,n,q,?] (default: y) y
    ```

Messages similar to the following now confirm that this disk is being reinitialized for Volume Manager use:

```
Initializing device c1t3d0.

Adding disk device c1t3d0 to disk group rootdg with disk
name disk40.
```

10. At the following prompt, indicate whether you want to continue to initialize more disks (y) or return to the vxdiskadm main menu (n):

```
Add or initialize other disks? [y,n,q,?] (default: n)
```

Adding a VM Disk to the Hot-Relocation Pool

Hot-relocation allows the system to automatically react to I/O failure by relocating redundant subdisks to other disks. Hot relocation then restores the affected Volume Manager objects and data. If a disk has already been designated as a spare in the disk group, the subdisks from the failed disk are relocated to the spare disk. Otherwise, any suitable free space in the disk group is used. Refer to Chapter 1, "The Command Line Interface" for more information.

To designate a disk as a spare, do the following:

1. Select menu item 12 (Mark a disk as a spare for a disk group) from the vxdiskadm main menu.

2. At the following prompt, enter a disk name (such as disk01):

```
Mark a disk as a spare for a disk group
Menu: VolumeManager/Disk/MarkSpareDisk

  Use this operation to mark a disk as a spare for a disk group.
  This operation takes, as input, a disk name. This is the same
  name that you gave to the disk when you added the disk to the
  disk group.


Enter disk name [<disk>,list,q,?] disk01
```

vxdiskadm displays the following confirmation:

```
Marking of disk01 in rootdg as a spare disk is complete.
```

3. At the following prompt, indicate whether you want to add more disks as spares (y) or return to the vxdiskadm main menu (n):

```
Mark another disk as a spare? [y,n,q,?] (default: n)
```

Any VM disk in this disk group can now use this disk as a spare in the event of a failure. If a disk fails, hot-relocation should automatically occur (if possible). You should be notified of the failure and relocation through electronic mail. After successful relocation, you may want to replace the failed disk.

Removing a VM Disk From the Hot-Relocation Pool

While a disk is designated as a spare, the space on that disk is not used as free space for the creation of Volume Manager objects within its disk group. If necessary, you can free a spare disk for general use by removing it from the pool of hot-relocation disks.

To verify which disks are currently designated as spares, select the list menu item from the vxdiskadm main menu. Disks that are spares are listed with the spare flag.

To remove a disk from the hot-relocation pool, do the following:

1. Select menu item 13 (Turn off the spare flag on a disk) from the vxdiskadm main menu.

2. At the following prompt, enter the name of a spare disk (such as disk01):

```
Turn off the spare flag on a disk
Menu: VolumeManager/Disk/UnmarkSpareDisk

  Use this operation to turn off the spare flag on a disk.
  This operation takes, as input, a disk name. This is the same
  name that you gave to the disk when you added the disk to the
  disk group.

Enter disk name [<disk>,list,q,?] disk01
```

vxdiskadm displays the following confirmation:

```
Disk disk01 in rootdg no longer marked as a spare disk.
```

3. At the following prompt, indicate whether you want to disable more spare
    disks (y) or return to the vxdiskadm main menu (n):

    ```
    Turn-off spare flag on another disk? [y,n,q,?] (default: n)
    ```

Displaying Information on Physical Disks

Displaying disk information shows you which disks are initialized, to which
disk groups they belong, and the disk status. 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.

To display disk information, do the following:

1. Select list from the vxdiskadm main menu.

2. At the following display, enter the address of the disk you want to see or
    enter all for a list of all disks:

    ```
    List disk information
    Menu: VolumeManager/Disk/ListDisk


    Use this menu operation to display a list of disks. You can

    also choose to list detailed information about the disk at

    a specific disk device address.


    Enter disk device or "all" [<address>,all,q,?] (default: all)
    ```

    • If you enter all, the Volume Manager displays the following
      information (some systems may display rootdisk in place of
      c0t0d0s2):

      ```
      DEVICE      DISK        GROUP       STATUS
      c0t0d0      c0t0d0s2    rootdg      online
      c1t0d0      disk01      rootdg      online
      c1t1d0      –           –           online


      Device to list in detail [<address>,none,q,?] (default: none)
      ```

- If you enter the address of the device for which you want information, complete disk information (including the device name, the type of disk, and information about the public and private partitions of the disk) is shown.

  Once you have examined this information, press Return to return to the main menu.

Removing a Physical Disk

Before removing a disk, make sure it contains no data, all data no longer needed, or the data can be moved to other disks. Then remove the disk as follows:

1. Select menu item 3 (Remove a disk) from the vxdiskadm main menu.

Note: You must disable the disk group before you can remove the last disk in that group.

---

2. At the following prompt, enter the disk name of the disk to be removed:

   ```
   Remove a disk
   Menu: VolumeManager/Disk/RemoveDisk


   Use this operation to remove a disk from a disk group. This
   operation takes a disk name as input. This is the same name
   that you gave to the disk when you added the disk to the disk
   group.


   Enter disk name [<disk>,list,q,?] disk01
   ```

3. If there are any volumes on the disk, the Volume Manager asks you whether they should be evacuated from the disk. If you wish to keep the volumes, answer y. Otherwise, answer n.

4. At the following verification prompt, press Return to continue:

   ```
   Requested operation is to remove disk disk01 from group rootdg.

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

`vxdiskadm` removes the disk from the disk group and displays the following success message:

```
Removal of disk disk01 is complete.
```

You can now remove the disk or leave it on your system as a replacement.

5. At the following prompt, indicate whether you want to remove other disks (y) or return to the `vxdiskadm` main menu (n):

```
Remove another disk? [y,n,q,?] (default: n)
```

Disabling a Physical Disk (Taking a Physical Disk Offline)

You can take a disk offline. If the disk is corrupted, you need to take it offline and remove it. You may be moving the physical disk device to another location to be connected to another system. To take a disk offline, first remove it from its disk group, then do the following:

1. Select menu item 11 (Disable (offline) a disk device) from the `vxdiskadm` main menu.

2. At the following prompt, enter the address of the disk you want to disable:

```
Disable (offline) a disk device
Menu: VolumeManager/Disk/OfflineDisk

Use this menu operation to disable all access to a disk device
by the Volume Manager. This operation can be applied only to
disks that are not currently in a disk group. Use this operation
if you intend to remove a disk from a system without rebooting.

NOTE: Many systems do not support disks that can be removed from
      a system during normal operation. On such systems, the
    offline operation is seldom useful.


Select a disk device to disable [<address>,list,q,?] c1t1d0
```

`vxdiskadm` disables the specified disk.

3. At the following prompt, indicate whether you want to disable another device (y) or return to the `vxdiskadm` main menu (n):

```
Disable another device? [y,n,q,?] (default: n)
```

Enabling a Physical Disk

If you move a disk from one system to another during normal system operation, the Volume Manager does not recognize the disk automatically. Use the enable disk task to get Volume Manager to scan the disk to identify it, and to determine if this disk is part of a disk group. Also, use this task to re-enable access to a disk that was disabled by either the disk group deport task or the disk device disable (offline) task. To enable a disk, do the following:

1. Select menu item 10 (Enable (online) a disk device) from the vxdiskadm main menu.

2. At the following prompt, enter the device name of the disk to be enabled (or enter list for a list of devices):

```
Enable (online) a disk device
Menu: VolumeManager/Disk/OnlineDisk


Use this operation to enable access to a disk that was disabled
with the "Disable (offline) a disk device" operation.


You can also use this operation to re-scan a disk that may have
been changed outside of the Volume Manager. For example, if a
disk is shared between two systems, the Volume Manager running
on the other system may have changed the disk. If so, you can
use this operation to re-scan the disk.


NOTE: Many vxdiskadm operations re-scan disks without user
      intervention. This will eliminate most needs to online a
      disk directly, except when the disk is directly offlined.
Select a disk device to enable [<address>,list,q,?] c1t1d0
```

vxdiskadm enables the specified device.

3. At the following prompt, indicate whether you want to enable another device (y) or return to the vxdiskadm main menu (n):

```
Enable another device? [y,n,q,?] (default: n)
```

☰ 5

Replacing a Physical Disk

If a disk fails, you need to replace that disk with another. This task requires disabling and removing the failed disk and installing a new disk in its place. To replace a disk, do the following:

1. Select menu item 4 (Remove a disk for replacement) from the vxdiskadm main menu.

2. At the following prompt, enter the name of the disk to be replaced (or enter list for a list of disks):

   ```
   Remove a disk for replacement
   Menu: VolumeManager/Disk/RemoveForReplace


   Use this menu operation to remove a physical disk from a disk
   group, while retaining the disk name. This changes the state
   for the disk name to a removed disk. If there are any

   initialized disks that are not part of a disk group, you will be
   given the option of using one of these disks as a replacement.

    Enter disk name [<disk>,list,q,?] disk02
   ```

   Additional displays show any volumes associated with the disk you wish to remove. You must decide whether to keep the data associated with the volumes or to allow that data to be lost when the disk is replaced. Answer any prompts accordingly.

3. At the following prompt, either select the device name of the replacement disk (from the list provided) or press Return to choose the default disk:

   ```
   The following devices are available as replacements:
     c1t1d0


   You can choose one of these disks now, to replace disk02.
   Select "none" if you do not wish to select a replacement disk.

   Choose a device, or select "none"
   [<device>,none,q,?] (default: c1t1d0)
   ```

4. At the following prompt, press Return to continue:

```
Requested operation is to remove disk disk02 from group rootdg.
The removed disk will be replaced with disk device c1t1d0.

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

`vxdiskadm` displays the following success messages:

```
Removal of disk disk02 completed successfully.
Proceeding to replace disk02 with device c1t1d0.

Disk replacement completed successfully.
```

5. At the following prompt, indicate whether you want to remove another disk (`y`) or return to the `vxdiskadm` main menu (`n`):

```
Remove another disk? [y,n,q,?] (default: n)
```

Replacing a Failed or Removed Disk

Disks can be removed and then replaced later. To do this, use menu item 4 (`Remove a disk for replacement`) to remove a disk, then do the following:

1. Select menu item 5 (`Replace a failed or removed disk`) from the `vxdiskadm` main menu.

2. Select the disk name of the disk to be replaced:

```
Replace a failed or removed disk
Menu: VolumeManager/Disk/ReplaceDisk

Use this menu operation to specify a replacement disk for a disk
that you removed with the "Remove a disk for replacement" menu
operation, or that failed during use. You will be prompted for
a disk name to replace and a disk device to use as a replacement
.   You can choose an uninitialized disk, in which case the disk
will be initialized, or you can choose a disk that you have
already initialized using the Add or initialize a disk menu
operation.
```

```
Select a removed or failed disk [<disk>,list,q,?] disk02
```

3. `vxdiskadm` displays the device names of the disk devices available for use as replacement disks; enter the device name of the device of your choice or press Return to select the default device:

Note: Your system may use a *device name* that differs from the examples. See Chapter 1, "Understanding Volume Manager," in the *VERITAS Volume Manaager Getting Started Guide,* for more information on device names.

```
The following devices are available as replacements:
   c1t0d0s2 c1t1d0s2


You can choose one of these disks to replace disk02.
Choose "none" to initialize another disk to replace disk02.


Choose a device, or select "none"
[<device>,none,q,?] (default: c1t0d0s2)
```

4. At the following prompt, press Return to replace the disk:

```
The requested operation is to use the initialized device
c1t0d0s2 to replace the removed or failed disk disk02 in disk
group rootdg.


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

`vxdiskadm` displays the following success message:

```
Replacement of disk disk02 in group rootdg with disk device
c1t0d0s2 completed successfully.
```

5. At the following prompt, indicate whether you want to replace another disk (y) or return to the `vxdiskadm` main menu (n):

```
Replace another disk? [y,n,q,?] (default: n)
```

Volume Tasks

This section describes the volume tasks that can be performed with the Volume Manager Support Operations. These volume tasks are used to protect data in case of device failures.

Mirroring Volumes on a VM Disk

Mirroring the volumes on a VM disk gives you one or more copies of your volumes in another disk location. By creating mirror copies of your volumes, you protect your system against loss of data in case of a disk failure. You can use this task on your root disk to make a second copy of the boot information available on an alternate disk. This allows you to boot your system even if your root disk is corrupted.

Note: This task only mirrors concatenated volumes. Volumes that are already mirrored or that contain subdisks that reside on multiple disks are ignored.

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 (Mirror volumes on a disk) from the vxdiskadm main menu.

2. At the following prompt, enter the disk name of the disk that you wish to mirror:

   ```
   Mirror volumes on a disk
   Menu: VolumeManager/Disk/Mirror


   This operation can be used to mirror volumes on a disk. These
   volumes can be mirrored onto another disk or onto any
   available disk space. Volumes will not be mirrored if they are
   already mirrored. Also, volumes that are comprised of more than
   one subdisk will not be mirrored.

   Mirroring volumes from the boot disk will produce a disk that
   can be used as an alternate boot disk.
   ```

```
Enter disk name [<disk>,list,q,?] disk02
```

3. At the following prompt, enter the target disk name (this disk must be the same size or larger than the originating disk):

```
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
```

Note: Be sure to always specify the destination disk when you are creating an alternate root disk. Otherwise, the Volume Manager selects a disk to be the alternate root disk. However, your system may not be able to boot from that disk.

---

4. At the following prompt, press Return to make the mirror:

```
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)
```

vxdiskadm displays the status of the mirroring operation:

```
Mirror volume voltest-bk00 ...

Mirroring of disk disk01 is complete.
```

5. At the following prompt, indicate whether you want to mirror volumes on another disk (y) or return to the vxdiskadm main menu (n):

```
Mirror volumes on another disk? [y,n,q,?] (default: n)
```

Moving Volumes from a VM Disk

Before you disable or remove a disk, you may want to move the data from that disk to other disks on the system. To do this, make sure that the target disks have sufficient space, then do the following:

1. Select menu item 7 (Move volumes from a disk) from the vxdiskadm main menu.

2. At the following prompt, enter the disk name of the disk whose volumes you wish to move:

   ```
   Move volumes from a disk
   Menu: VolumeManager/Disk/Evacuate


   Use this menu operation to move any volumes that are using a
   disk onto other disks. Use this menu immediately prior to
   removing a disk, either permanently or for replacement. You can
   specify a list of disks to move volumes onto, or you can move
   the volumes to any available disk space in the same disk group.

   NOTE: Simply moving volumes off of a disk, without also removing
         the disk, does not prevent volumes from being moved onto
         the disk by future operations. For example, using two
         consecutive move operations may move volumes from the
         second disk to the first.
   Enter disk name [<disk>,list,q,?] disk01
   ```

   After the following display, you can optionally specify a list of disks to which the volume(s) should be moved.

   ```
   You can now specify a list of disks to move onto.  Specify a list
   of disk media names (e.g., disk01) all on one line separated by
   blanks.  If you do not enter any disk media names, then the
   volumes will be moved to any available space in the disk group.
   ```

3. At the following prompt, press Return to move the volumes:

   ```
   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)
```

As the volumes are moved from the disk, `vxdiskadm` displays the status of the operation:

```
Move volume voltest ...
Move volume voltest-bk00 ...
```

When the volumes have all been moved, `vxdiskadm` displays the following success message:

```
Evacuation of disk disk01 is complete.
```

4. At the following prompt, indicate whether you want to move volumes from another disk (y) or return to the `vxdiskadm` main menu (n):

```
Move volumes from another disk? [y,n,q,?] (default: n)
```

Disk Group Tasks

The following sections describe disk group tasks available with the Volume Manager Support operations.

Adding a Disk to a Disk Group

You may wish to add a new disk to an already established disk group. For example, the current disks may 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 following the steps required to add a disk. Refer to "Bringing Physical Disks Under Volume Manager Control."

Creating a Disk Group

You may want 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 must be moved to other systems.

- The work load must be balanced across a number of systems.

In these cases, it is important that the data related to particular application(s) or users be located on an identifiable set of disk drives. When these disks are moved, all data belonging to the application(s) or users on these disks, and no other data, is moved.

Note: The Volume Manager supports a default disk group, `rootdg`, in which all volumes are created if no further specification is given. All commands default to `rootdg` as well.

A disk group can only be created along with a disk. A disk group must have at least one disk associated with it.

If you need to create a disk group in addition to `rootdg`, do the following:

1. Select menu item 1 (`Add or initialize one or more disks`) from the `vxdiskadm` main menu.

2. At the following prompt, enter the disk device name of the disk to be added to Volume Manager control:

```
Add or initialize disks
Menu: VolumeManager/Disk/AddDisks

  Use this operation to add one or more disks to a disk group.
  You can add the selected disks to an existing disk group or
  to a new disk group that will be created as a part of the
  operation. The selected disks may also be added to a disk
  group as spares. The selected disks may also be initialized
  without adding them to a disk group leaving the disks
  available for use as replacement disks.

  More than one disk or pattern may be entered at the prompt.
  Here are some disk selection examples:

  all:          all disks
  c3c4t2:       all disks on both controller 3 and controller
                4, target 2
```

```
    c3t4d0:          a single disk

 Select disk devices to add:
 [<pattern-list>,all,list,q,?] c1t2d0
```

Where *<pattern-list>* can be a single disk, or a series of disks and/or controllers (with optional targets). If *<pattern-list>* consists of multiple items, those items must be separated by white space.

If you do not know the address (device name) of the disk to be added, enter l or list at the prompt for a listing of all disks.

3. To continue with the operation, enter y (or press Return) at the following prompt:

```
 Here is the disk selected.  Output format: [Device_Name]

 c1t2d0
```

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

4. At the following prompt, specify the disk group to which the disk should be added (anotherdg, in this case):

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

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

5. vxdiskadm confirms that no active disk group currently exists with the same name and prompts for confirmation that you really want to create this new disk group:

```
 There is no active disk group named anotherdg.

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

6. At the following prompt, either press Return to accept the default disk name or enter a disk name:

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

7. When vxdiskadm asks whether this disk should become a hot-relocation spare, enter n (or press Return):

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

8. To continue with the operation, enter y (or press Return) at the following prompt:

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

c1t2d0

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

9. The following prompt appears if this disk was previously initialized for future Volume Manager use; enter y to confirm that you now want to use this disk:

```
The following disk device appears to have been initialized
already. The disk is currently available as a replacement disk.
Output format: [Device_Name]

c1t2d0

Use this device? [y,n,q,?] (default: y) y
```

If you are sure you want to reinitialize the disk, enter y (or press Return) at the following prompt:

```
The following disk you selected for use appears to already have
been initialized for the Volume Manager.  If you are certain the
disk has already been initialized for the Volume Manager, then
you do not need to reinitialize the disk device.
Output format: [Device_Name]

c1t2d0

Reinitialize this device? [y,n,q,?] (default: y) y
```

Messages similar to the following confirm that this disk is being reinitialized for Volume Manager use:

```
Initializing device c1t2d0.

Creating a new disk group named anotherdg containing the disk
device c1t2d0 with the name another01.
```

10. At the following prompt, indicate whether you want to continue to initialize more disks (`y`) or return to the `vxdiskadm` main menu (`n`):

```
Add or initialize other disks? [y,n,q,?] (default: n)
```

Deporting a Disk Group

Use the deport disk group task 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.

To deport a disk group, do the following:

1. Select menu item 9 (`Remove access to (deport) a disk group`) from the `vxdiskadm` main menu.

2. At the following prompt, enter the name of the disk group to be deported:

```
Remove access to (deport) a disk group
Menu: VolumeManager/Disk/DeportDiskGroup


Use this menu operation to remove access to a disk group that is
currently enabled (imported) by this system. Deport a disk
group if you intend to move the disks in a disk group to another
system. Also, deport a disk group if you want to use all of the
disks remaining in a disk group for some new purpose.


You will be prompted for the name of a disk group. You will also
be asked if the disks should be disabled (offlined). For
removable disk devices on some systems, it is important to
disable all access to the disk before removing the disk.
```

```
Enter name of disk group [<group>,list,q,?] (default: list)
newdg
```

3. At the following prompt, enter y if you intend to remove the disks in this disk group:

```
The requested operation is to disable access to the removable
disk group named newdg. This disk group is stored on the
following disks:
  newdg01 on device c1t1d0s2


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)
```

4. At the following prompt, press Return to continue with the operation:

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

Once the disk group is deported, vxdiskadm displays the following message:

```
Removal of disk group newdg was successful.
```

5. At the following prompt, indicate whether you want to disable another disk group (y) or return to the vxdiskadm main menu (n):

```
Disable another disk group? [y,n,q,?] (default: n)
```

Importing a Disk Group

Use this menu task to enable access by this system to a disk group. If you wish to move a disk group from one system to another, you must first disable (deport) it on the original system, then move the disk between systems and enable (import) the disk group.

To import a disk group, do the following:

1. Select menu item 8 (Enable access to (import) a disk group) from the vxdiskadm main menu.

2. At the following prompt, enter the name of the disk group to import:

```
Enable access to (import) a disk group
Menu: VolumeManager/Disk/EnableDiskGroup

Use this operation to enable access to a disk group. This can be
used as the final part of moving a disk group from one system to
another. The first part of moving a disk group is to use the
"Remove access to (deport) a disk group" operation on the
original host.

A disk group can be imported from another host that failed
without first deporting the disk group. Be sure that all disks
in the disk group are moved between hosts.

If two hosts share a SCSI bus, be very careful to ensure that
the other host really has failed or has deported the disk group.
If two active hosts import a disk group at the same time, the
disk group will be corrupted and will become unusable.

Select disk group to import [<group>,list,q,?] (default: list)
newdg
```

Once the import is complete, vxdiskadm displays the following success
message:

```
The import of newdg was successful.
```

3. At the following prompt, indicate whether you want to import another disk
group (y) or return to the vxdiskadm main menu (n):

```
Select another disk group? [y,n,q,?] (default: n)
```

Exiting the Volume Manager Support Tasks

When you have completed all of your disk administration activities, exit the
Volume Manager Support Operations by selecting q from the main menu.

Index

vxplex, 67
vxprint, 68
vxsd, 67
vxstat, 68
vxvol, 69

V

vm disk operations, 22
Volume Manager, 1
volumes, 85, 91
    adding a mirror, 45
    backing up, 50
    changing information, 91
    concatenated, creating, 38
    creating, 2, 36, 85
        concatenated, 38
    decreasing size, 43
    displaying, 90
    displaying configuration, 54
    error policies, 91
    estimating size, 89
    increasing size, 40
    initializing, 87, 88
    kernel state, 69
    managing, 2
    mirroring, 45
    mirroring all existing volumes, 46
    mirroring on disk, 133
    moving from disk, 135
    operations, 69
    permissions, 91
    RAID-5
        creating, 39
    read policy, 42
    recovery, 51
    removing, 43
    removing a RAID-5 log from, 47
    spanned
        creating, 38
    starting, 48, 49
    stopping, 48, 49
    striped

        creating, 39
vxassist, 2, 40, 45, 55, 56, 57, 61, 62, 85, 86
    backup, 56
    creating volumes, 60
    defaults, 63
    description of, 61
    growby, 40
    growto, 40
    shrinkby, 40
    shrinkto, 40
vxassist addlog, 46
vxassist growby, 41
vxassist growto, 41
vxassist make, 17, 38
vxassist snapshot, 51
vxassist snapstart, 51
vxconfigd, 65
vxdctl, 65
    description of, 65
vxdg, 32
    moving disk groups, 32
vxdg free, 5
vxdg list, 32
vxdg rmdisk, 22
vxdisk list, 22
vxdisk rm, 22
vxdiskadd, 12, 23, 25
vxdiskadm
    starting, 102
vxedit, 44, 53, 71, 75, 78, 80
    description of, 44
    removing subdisks, 71
vxedit rename, 16
vxedit set, 17
vxmake, 46, 65, 66, 70, 72, 76, 77, 86, 87
    associating mirrors, 77
    associating subdisks, 72
    creating mirrors, 76
    creating subdisks, 70
    description of, 65