

VERITAS Volume Manager™ 3.2

Troubleshooting Guide

Solaris

August 2001
30-000394-011


VERITAS

Disclaimer

The information contained in this publication is subject to change without notice. VERITAS Software Corporation makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. VERITAS Software Corporation shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this manual.

Copyright

Copyright © 2000-2001 VERITAS Software Corporation. All Rights Reserved. VERITAS, VERITAS SOFTWARE, the VERITAS logo, *Business Without Interruption*, VERITAS The Data Availability Company, and VERITAS Volume Manager are trademarks or registered trademarks of VERITAS Software Corporation in the USA and/or other countries. Other product names mentioned herein may be trademarks or registered trademarks of their respective companies.

Printed in the USA and/or Ireland, July 2001.

VERITAS Software Corporation
350 Ellis Street
Mountain View, CA 94043
Phone 650-527-8000
Fax 650-527-2901
www.veritas.com



Contents

Preface	vii
Introduction	vii
Audience and Scope	vii
Organization	vii
Related Documents	viii
Conventions	ix
Getting Help	x
Downloading and Running VRTSexplorer	x
.....	xii
 Chapter 1. Recovery from Hardware Failure	1
Introduction	1
Understanding the Plex State Cycle	2
Listing Unstartable Volumes	4
Restarting a Disabled Volume	4
Recovering a Mirrored Volume	4
Reattaching Disks	5
Failures on RAID-5 Volumes	6
System Failures	6
Disk Failures	7
Default Startup Recovery Process for RAID-5	8
Recovering a RAID-5 Volume	9
Recovery After Moving RAID-5 Subdisks	12
Starting RAID-5 Volumes	12



Recovering from Incomplete Disk Group Moves	15
Recovery from DCO Log Volume Failure	16
Chapter 2. Recovery from Boot Disk Failure	19
Introduction	19
Possible root, swap, and usr Configurations	19
Booting from Alternate Boot Disks	20
The Boot Process on SPARC Systems	21
Hot-Relocation and Boot Disk Failure	22
Unrelocating Subdisks to a Replacement Boot Disk	22
Recovery from Boot Failure	23
Boot Device Cannot be Opened	23
Cannot Boot From Unusable or Stale Plexes	24
Invalid UNIX Partition	26
Incorrect Entries in /etc/vfstab	26
Missing or Damaged /etc/system	28
Repairing root or /usr File Systems on Mirrored Volumes	30
Recovering a root Disk and root Mirror from Backup Tape	30
Re-Adding and Replacing Boot Disks	33
Re-Adding a Failed Boot Disk	34
Replacing a Failed Boot Disk	35
Recovery by Reinstallation	36
General Reinstallation Information	36
Reinstalling the System and Recovering VxVM	37
Chapter 3. Error Messages	47
Introduction	47
Logging Error Messages	47
Configuring Logging in the Startup Script	48
Understanding Error Messages	49
Kernel Panic Messages	49

Kernel Warning Messages	49
Kernel Notice Messages	54
vxassist Error Messages	55
vxassist Warning Messages	55
vxconfigd Fatal Error Messages	56
vxconfigd Error Messages	57
vxconfigd Warning Messages	70
vxconfigd Notice Messages	75
vxdbg Error Messages	77
vxdump Notice Messages	79
vxdumpadm Error Messages	81
Cluster Error Messages	82
Index	89





Preface

Introduction

The *VERITAS Volume Manager™ Troubleshooting Guide* provides information about how to recover from hardware failure, and how to understand and deal with VxVM error messages. For detailed information about the Volume Manager and how to use the Volume Manager, refer to the *VERITAS Volume Manager Administrator's Guide*.

Audience and Scope

This guide is intended for system administrators responsible for installing, configuring, and maintaining systems under the control of the VERITAS Volume Manager.

This guide assumes that the user has a:

- ◆ working knowledge of the UNIX operating system
- ◆ basic understanding of UNIX system administration
- ◆ basic understanding of volume management

The purpose of this guide is to help the system administrator recover from the failure of disks and other hardware upon which virtual software objects such as subdisks, plexes and volumes are constructed in Volume Manager. Guidelines are also included on how to understand and react to the various VxVM error messages that you may see.

Organization

This guide is organized as follows:

- ◆ [Recovery from Hardware Failure](#)
- ◆ [Recovery from Boot Disk Failure](#)
- ◆ [Error Messages](#)



Related Documents

The following documents provide information related to the Volume Manager:

- ◆ *VERITAS Volume Manager Installation Guide*
- ◆ *VERITAS Volume Manager Release Notes*
- ◆ *VERITAS Volume Manager Hardware Application Notes*
- ◆ *VERITAS Volume Manager Administrator's Guide*
- ◆ *VERITAS Volume Manager Storage Administrator Administrator's Guide*
- ◆ *VERITAS Volume Manager manual pages*

Conventions

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

Typeface	Usage	Examples
<code>monospace</code>	Computer output, file contents, files, directories, software elements such as command options, function names, and parameters	Read tunables from the <code>/etc/vx/tunefstab</code> file. See the <code>ls(1)</code> manual page for more information.
<i>italic</i>	New terms, book titles, emphasis, variables to be replaced by a name or value	See the <i>User's Guide</i> for details. The variable <code>ncsize</code> determines the value of...
<code>monospace</code> (bold)	User input; the “#” symbol indicates a command prompt	# mount -F vxfs /h/filesys
<i>monospace</i> (bold and italic)	Variables to be replaced by a name or value in user input	# mount -F <i>fstype</i> <i>mount_point</i>

Symbol	Usage	Examples
%	C shell prompt	
\$	Bourne/Korn/Bash shell prompt	
#	Superuser prompt (all shells)	
\	Continued input on the following line	# mount -F vxfs \ /h/filesys
[]	In a command synopsis, brackets indicates an optional argument	<code>ls [-a]</code>
	In a command synopsis, a vertical bar separates mutually exclusive arguments	<code>mount [suid nosuid]</code>



Getting Help

If you have any comments or problems with the VERITAS products, contact the VERITAS Technical Support:

- ◆ U.S. and Canadian Customers: 1-800-342-0652
- ◆ International Customers: +1 (650) 527-8555
- ◆ Email: support@veritas.com

For license information (U.S. and Canadian Customers):

- ◆ Phone: 1-925-931-2464
- ◆ Email: license@veritas.com
- ◆ Fax: 1-925-931-2487

For software updates:

- ◆ Email: swupdate@veritas.com

For additional technical support information, such as TechNotes, product alerts, and hardware compatibility lists, visit the VERITAS Technical Support Web site at:

- ◆ <http://support.veritas.com> (U.S. and Canada Customers)

For additional information about VERITAS and VERITAS products, visit the Web site at:

- ◆ <http://www.veritas.com>

Downloading and Running VRTSexplorer

If you have access to the Internet, you can use the VRTSexplorer program to assist Technical Support in diagnosing the cause of your problem as follows:

1. Use a web browser or the ftp program to download the VRTSexplorer program at the following URL:

`ftp://ftp.veritas.com/pub/support/vxexplore.tar.Z`

Save the file to a temporary directory such as `/tmp` as shown in these instructions. If you download the file to a different directory, substitute its pathname for `/tmp` throughout.

2. Log in as `root` on the affected system, and use the following commands to extract the contents of the downloaded file to the directory `/tmp/VRTSexplorer`:

```
# cd /tmp
# zcat vxexplore.tar.Z | tar xvf -
```

3. Run the `VRTSexplorer` program located in the `VRTSexplorer` directory by entering the following command:

```
# /tmp/VRTSexplorer/VRTSexplorer
```

4. When `VRTSexplorer` prompts you for a destination directory for the information that it collects, press Return to accept the default directory `/tmp`, or enter a pathname of your own choice. `VRTSexplorer` writes the results of its investigations to a compressed tar file named `VRTSexplorer.case_number.tar.Z` in the specified directory.
5. Use the file upload facility of your web browser or the `ftp` program to transfer the file output by `VRTSexplorer` to the VERITAS Customer Support anonymous FTP site:

```
ftp://ftp.veritas.com/incoming
```

6. Call VERITAS Customer Support on 1-800-342-0652, inform them that you have run `VRTSexplorer` and tell them the name of the file that you transferred to the FTP site.

Alternatively, if you have already been assigned a call ID number by Customer Support, e-mail `support@veritas.com` including your case ID number in the subject line.

For more information about the `VRTSexplorer` program, consult the `README` file located in the `VRTSexplorer` directory.





Introduction

VERITAS Volume Manager (VxVM) protects systems from disk and other hardware failures and helps you to recover from such events. This chapter describes recovery procedures and information to help you prevent loss of data or system access due to disk and other hardware failures.

If a volume has a disk I/O failure (for example, because the disk has an uncorrectable error), VxVM can detach the plex involved in the failure. I/O stops on that plex but continues on the remaining plexes of the volume.

If a disk fails completely, VxVM can detach the disk from its disk group. All plexes on the disk are disabled. If there are any unmirrored volumes on a disk when it is detached, those volumes are also disabled.

Note Apparent disk failure may not be due to a fault in the physical disk media or the disk controller, but may instead be caused by a fault in an intermediate or ancillary component such as a cable, host bus adapter, or power supply.

The hot-relocation feature in VxVM automatically detects disk failures, and notifies the system administrator and other nominated users of the failures by electronic mail. Hot-relocation also attempts to use spare disks and free disk space to restore redundancy and to preserve access to mirrored and RAID-5 volumes. For more information, see the “Administering Hot-Relocation” chapter in the *VERITAS Volume Manager Administrator's Guide*.

Recovery from failures of the boot (`root`) disk requires the use of the special procedures described in “[Recovery from Boot Disk Failure](#)” on page 19. The chapter also includes procedures for repairing the `root (/)` and `usr` file systems.

For more information about protecting your system, see the *VERITAS Volume Manager Installation Guide*.

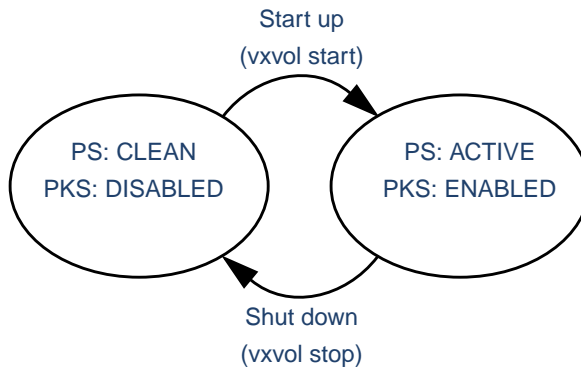


Understanding the Plex State Cycle

Changing plex states are part of normal operations, and do not necessarily indicate abnormalities that must be corrected. A firm understanding of the various plex states and their interrelationship is necessary if you want to be able to perform the recovery procedure described in this chapter.

The figure “[Main Plex State Cycle](#)” shows the main transitions that take place between plex states in VxVM. (For more information about plex states, see the chapter “Creating and Administering Plexes” in the *VERITAS Volume Manager Administrator’s Guide*.)

Main Plex State Cycle



PS = Plex State

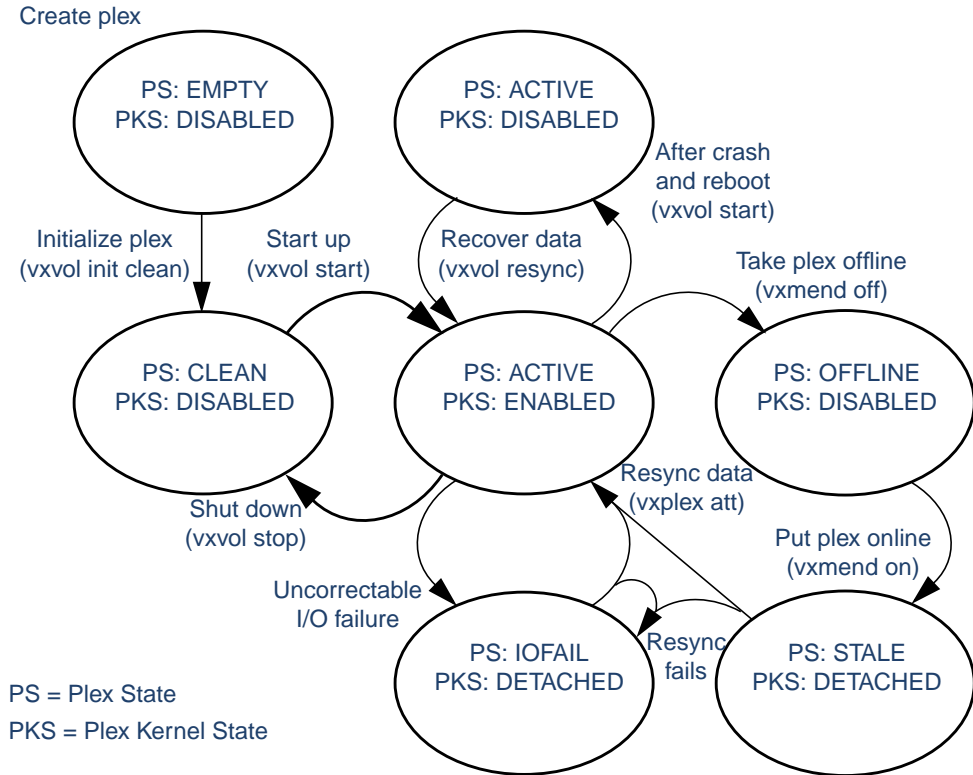
PKS = Plex Kernel State

At system startup, volumes are started automatically and the `vxvol start` task makes all CLEAN plexes ACTIVE. At shutdown, the `vxvol stop` task marks all ACTIVE plexes CLEAN. If all plexes are initially CLEAN at startup, this indicates that a controlled shutdown occurred and optimizes the time taken to start up the volumes.

The next figure “[Additional Plex State Transitions](#)” shows additional transitions that are possible between plex states as a result of hardware problems, abnormal system shutdown, and intervention by the system administrator.

When first created, a plex has state EMPTY until the volume to which it is attached is initialized. Its state is then set to CLEAN. Its plex kernel state remains set to DISABLED and is not set to ENABLED until the volume is started.

Additional Plex State Transitions



After a system crash and reboot, all plexes of a volume are ACTIVE but marked with plex kernel state DISABLED until their data is recovered by the `vxvol resync` task.

A plex may be taken offline with the `vxmend off` command, made available again using `vxmend on`, and its data resynchronized with the other plexes when it is reattached using `vxplex att`. A failed resynchronization or uncorrectable I/O failure places the plex in the IOFAIL state.

The following section, “[Listing Unstartable Volumes](#),” describes the actions that you can take if a system crash or I/O error leaves no plexes of a mirrored volume in a CLEAN or ACTIVE state.

For information on the recovery of RAID-5 volumes, see “[Failures on RAID-5 Volumes](#)” on page 6 and subsequent sections.



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 `vxinfo` command. This displays information about the accessibility and usability of volumes:

```
# vxinfo [-g diskgroup] [volume ...]
```

The following example output shows one volume, `mkting`, as being unstartable:

home	fsgen	Started
mkting	fsgen	Unstartable
src	fsgen	Started
rootvol	root	Started
swapvol	swap	Started

Restarting a Disabled Volume

If a disk failure caused a volume to be disabled, you must restore the volume from a backup after replacing the failed disk. Any volumes that are listed as `Unstartable` must be restarted using the `vxvol` command before restoring their contents from a backup. For example, to restart the volume `mkting` so that it can be restored from backup, use the following command:

```
# vxvol -o bg -f start mkting
```

The `-f` option forcibly restarts the volume, and the `-o bg` option resynchronizes plexes as a background task.

Recovering a Mirrored Volume

A system crash or an I/O error can corrupt one or more plexes of a mirrored 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 as follows:

1. Place the desired plex in the `CLEAN` state using the following command:

```
# vxmend fix clean plex
```

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

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


2. To recover the other plexes in a volume from the `CLEAN` plex, the volume must be disabled, and the other plexes must be `STALE`. If necessary, make any other `CLEAN` or `ACTIVE` plexes `STALE` by running the following command on each of these plexes in turn:

```
# vxmend fix stale plex
```

3. To enable the `CLEAN` plex and to recover the `STALE` plexes from it, use the following command:

```
# vxvol start volume
```

For example, to recover volume `vol01`:

```
# vxvol start vol01
```

For more information about the `vxmend` and `vxvol` command, see the `vxmend(1M)` and `vxvol(1M)` manual pages.

Note Following severe hardware failure of several disks or other related subsystems underlying all the mirrored plexes of a volume, it may be impossible to recover the volume using `vxmend`. In this case, remove the volume, recreate it on hardware that is functioning correctly, and restore the contents of the volume from a backup or from a snapshot image.

Reattaching Disks

You can perform a reattach operation if a disk fails completely and hot-relocation is not possible, or if VxVM is started with some disk drivers unloaded and unloadable (causing disks to enter the failed state). If the underlying problem has been fixed, you can use the `vxreattach` command to reattach the disks without plexes being flagged as `STALE`. However, the reattach must occur before any volumes on the disk are started.

The `vxreattach` command is called as part of disk recovery from the `vxdiskadm` menus and during the boot process. If possible, `vxreattach` reattaches the failed disk media record to the disk with the same device name. Reattachment places a disk in the same disk group as it was located in before and retains its original disk media name.

After reattachment takes place, recovery may not be necessary. Reattachment can fail if the original (or another) cause for the disk failure still exists.

You can use the command `vxreattach -c` to check whether reattachment is possible, without performing the operation. Instead, it displays the disk group and disk media name where the disk can be reattached.

See the `vxreattach(1M)` manual page for more information on the `vxreattach` command.



Failures on RAID-5 Volumes

Failures are seen in two varieties: *system failures* and *disk failures*. A system failure means that the system has abruptly ceased to operate due to an operating system panic or power failure. Disk failures imply that the data on some number of disks has become unavailable due to a system failure (such as a head crash, electronics failure on disk, or disk controller failure).

System Failures

RAID-5 volumes are designed to remain available with a minimum of disk space overhead, if there are disk failures. However, many forms of RAID-5 can have data loss after a system failure. Data loss occurs because a system failure causes the data and parity in the RAID-5 volume to become unsynchronized. Loss of synchronization occurs because the status of writes that were outstanding at the time of the failure cannot be determined.

If a loss of sync occurs while a RAID-5 volume is being accessed, the volume is described as having *stale parity*. The parity must then be reconstructed by reading all the non-parity columns within each stripe, recalculating the parity, and writing out the parity stripe unit in the stripe. This must be done for every stripe in the volume, so it can take a long time to complete.

Caution While the resynchronization of a RAID-5 volume *without* log plexes is being performed, any failure of a disk within the volume causes its data to be lost.

Besides the vulnerability to failure, the resynchronization process can tax the system resources and slow down system operation.

RAID-5 logs reduce the damage that can be caused by system failures, because they maintain a copy of the data being written at the time of the failure. The process of resynchronization consists of reading that data and parity from the logs and writing it to the appropriate areas of the RAID-5 volume. This greatly reduces the amount of time needed for a resynchronization of data and parity. It also means that the volume never becomes truly stale. The data and parity for all stripes in the volume are known at all times, so the failure of a single disk cannot result in the loss of the data within the volume.

Disk Failures

Disk failures can cause the data on a disk to become unavailable. In terms of a RAID-5 volume, this means that a subdisk becomes unavailable.

This can occur due to an uncorrectable I/O error during a write to the disk. The I/O error can cause the subdisk to be detached from the array or a disk being unavailable when the system is booted (for example, from a cabling problem or by having a drive powered down).

When this occurs, the subdisk cannot be used to hold data and is considered *stale* and *detached*. If the underlying disk becomes available or is replaced, the subdisk is still considered stale and is not used.

If an attempt is made to read data contained on a stale subdisk, the data is reconstructed from data on all other stripe units in the stripe. This operation is called a *reconstructing-read*. This is a more expensive operation than simply reading the data and can result in degraded read performance. When a RAID-5 volume has stale subdisks, it is considered to be in *degraded mode*.

A RAID-5 volume in degraded mode can be recognized from the output of the `vxprint -ht` command as shown in the following display:

V	NAME	RVG	KSTATE	STATE	LENGTH	READPOL	PREFPLEX	UTYPE
PL	NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WID	MODE
SD	NAME	PLEX	DISK	DISKOFFS	LENGTH	[COL/]OFF	DEVICE	MODE
SV	NAME	PLEX	VOLNAME	NVOLLAY	RLENGTH	[COL/]OFF	AM/NM	MODE
...								
v	r5vol	-	ENABLED	DEGRADED	204800	RAID	-	raid5
pl	r5vol-01	r5vol	ENABLED	ACTIVE	204800	RAID	3/16	RW
sd	disk01-01	r5vol-01	disk01	0	102400	0/0	c2t9d0	ENA
sd	disk02-01	r5vol-01	disk02	0	102400	1/0	c2t10d0	dS
sd	disk03-01	r5vol-01	disk03	0	102400	2/0	c2t11d0	ENA
pl	r5vol-02	r5vol	ENABLED	LOG	1440	CONCAT	-	RW
sd	disk04-01	r5vol-02	disk04	0	1440	0	c2t12d0	ENA
pl	r5vol-03	r5vol	ENABLED	LOG	1440	CONCAT	-	RW
sd	disk05-01	r5vol-03	disk05	0	1440	0	c2t14d0	ENA

The volume `r5vol` is in degraded mode, as shown by the volume state, which is listed as `DEGRADED`. The failed subdisk is `disk02-01`, as shown by the `MODE` flags; `d` indicates that the subdisk is detached, and `S` indicates that the subdisk's contents are stale.

Note Do not run the `vxr5check` command on a RAID-5 volume that is in degraded mode.

A disk containing a RAID-5 log plex can also fail. The failure of a single RAID-5 log plex has no direct effect on the operation of a volume provided that the RAID-5 log is mirrored. However, loss of all RAID-5 log plexes in a volume makes it vulnerable to a complete



failure. In the output of the `vxprint -ht` command, failure within a RAID-5 log plex is indicated by the plex state being shown as `BADLOG` rather than `LOG`. This is shown in the following display, where the RAID-5 log plex `r5vol-11` has failed:

V	NAME	RVG	KSTATE	STATE	LENGTH	READPOL	PREFPLEX	UTYPE
PL	NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WID	MODE
SD	NAME	PLEX	DISK	DISKOFFSLENGTH	[COL/]OFF	DEVICE		MODE
SV	NAME	PLEX	VOLNAME	NVOLLAYRLLENGTH	[COL/]OFF	AM/NM		MODE
...								
v	r5vol	RAID-5	ENABLED	ACTIVE	204800	RAID	-	raid5
pl	r5vol-01	r5vol	ENABLED	ACTIVE	204800	RAID	3/16	RW
sd	disk01-01	r5vol-01	disk01	0	102400	0/0	c2t9d0	ENA
sd	disk02-01	r5vol-01	disk02	0	102400	1/0	c2t10d0	ENA
sd	disk03-01	r5vol-01	disk03	0	102400	2/0	c2t11d0	ENA
pl	r5vol-02	r5vol	DISABLED	BADLOG	1440	CONCAT	-	RW
sd	disk04-01	r5vol-11	disk04	0	1440	0	c2t12d0	ENA
pl	r5vol-03	r5vol	ENABLED	LOG	1440	CONCAT	-	RW
sd	disk05-01	r5vol-12	disk05	0	1440	0	c2t14d0	ENA

Default Startup Recovery Process for RAID-5

VxVM may need to perform several operations to restore fully the contents of a RAID-5 volume and make it usable. Whenever a volume is started, any RAID-5 log plexes are zeroed before the volume is started. This prevents random data from being interpreted as a log entry and corrupting the volume contents. Also, some subdisks may need to be recovered, or the parity may need to be resynchronized (if RAID-5 logs have failed).

VxVM takes the following steps when a RAID-5 volume is started:

1. If the RAID-5 volume was not cleanly shut down, it is checked for valid RAID-5 log plexes.
 - If valid log plexes exist, they are replayed. This is done by placing the volume in the `DETACHED` volume kernel state and setting the volume state to `REPLAY`, and enabling the RAID-5 log plexes. If the logs can be successfully read and the replay is successful, move on to Step 2.
 - If no valid logs exist, the parity must be resynchronized. Resynchronization is done by placing the volume in the `DETACHED` volume kernel state and setting the volume state to `SYNC`. Any log plexes are left in the `DISABLED` plex kernel state.

The volume is not made available while the parity is resynchronized because any subdisk failures during this period makes the volume unusable. This can be overridden by using the `-o unsafe start` option with the `vxvol` command. If any stale subdisks exist, the RAID-5 volume is unusable.

Caution The `-o unsafe` start option is considered dangerous, as it can make the contents of the volume unusable. Using it is not recommended.

2. Any existing log plexes are zeroed and enabled. If all logs fail during this process, the start process is aborted.
3. If no stale subdisks exist or those that exist are recoverable, the volume is put in the `ENABLED` volume kernel state and the volume state is set to `ACTIVE`. The volume is now started.

Recovering a RAID-5 Volume

The types of recovery that may typically be required for RAID-5 volumes are the following:

- ◆ [Parity Resynchronization](#); see page 10.
- ◆ [Log Plex Recovery](#); see page 11.
- ◆ [Stale Subdisk Recovery](#); see page 11.

Parity resynchronization and stale subdisk recovery are typically performed when the RAID-5 volume is started, or shortly after the system boots. They can also be performed by running the `vxrecover` command.

For more information on starting RAID-5 volumes, see “[Starting RAID-5 Volumes](#)” on page 12.

If hot-relocation is enabled at the time of a disk failure, system administrator intervention is not required unless no suitable disk space is available for relocation. Hot-relocation is triggered by the failure and the system administrator is notified of the failure by electronic mail.

Hot relocation automatically attempts to relocate the subdisks of a failing RAID-5 plex. After any relocation takes place, the hot-relocation daemon (`vxrelocd`) also initiates a parity resynchronization.

In the case of a failing RAID-5 log plex, relocation occurs only if the log plex is mirrored; the `vxrelocd` daemon then initiates a mirror resynchronization to recreate the RAID-5 log plex. If hot-relocation is disabled at the time of a failure, the system administrator may need to initiate a resynchronization or recovery.



Note Following severe hardware failure of several disks or other related subsystems underlying a RAID-5 plex, it may be impossible to recover the volume using the methods described in this chapter. In this case, remove the volume, recreate it on hardware that is functioning correctly, and restore the contents of the volume from a backup.

Parity Resynchronization

In most cases, a RAID-5 array does not have stale parity. Stale parity only occurs after all RAID-5 log plexes for the RAID-5 volume have failed, and then only if there is a system failure. Even if a RAID-5 volume has stale parity, it is usually repaired as part of the volume start process.

If a volume without valid RAID-5 logs is started and the process is killed before the volume is resynchronized, the result is an active volume with stale parity. For an example of the output of the `vxprint -ht` command, see the following example for a stale RAID-5 volume:

```
V  NAME      RVG      KSTATE  STATE   LENGTH  READPOL  PREFPLEX  UTYPE
PL NAME      VOLUME   KSTATE  STATE   LENGTH  LAYOUT   NCOL/WID  MODE
SD NAME      PLEX     DISK    DISKOFFSLENGTH  [COL/]OFF DEVICE  MODE
SV NAME      PLEX     VOLNAME NVOLLAYRLLENGTH  [COL/]OFF AM/NM   MODE
...
v  r5vol      -        ENABLED NEEDSYNC 204800  RAID     -         raid5
pl r5vol-01  r5vol    ENABLED ACTIVE  204800  RAID     3/16      RW
sd disk01-01 r5vol-01disk01  0        102400  0/0      c2t9d0    ENA
sd disk02-01 r5vol-01disk02  0        102400  1/0      c2t10d0   dS
sd disk03-01 r5vol-01disk03  0        102400  2/0      c2t11d0   ENA
...
```

This output lists the volume state as `NEEDSYNC`, indicating that the parity needs to be resynchronized. The state could also have been `SYNC`, indicating that a synchronization was attempted at start time and that a synchronization process should be doing the synchronization. If no such process exists or if the volume is in the `NEEDSYNC` state, a synchronization can be manually started by using the `resync` keyword for the `vxvol` command. For example, to resynchronize the RAID-5 volume in the figure “Invalid RAID-5 Volume” on page 13, use the following command:

```
# vxvol resync r5vol
```

Parity is regenerated by issuing `VOL_R5_RESYNC` `ioctl`s to the RAID-5 volume. The resynchronization process starts at the beginning of the RAID-5 volume and resynchronizes a region equal to the number of sectors specified by the `-o iosize` option. If the `-o iosize` option is not specified, the default maximum I/O size is used. The `resync` operation then moves onto the next region until the entire length of the RAID-5 volume has been resynchronized.

For larger volumes, parity regeneration can take a long time. It is possible that the system could be shut down or crash before the operation is completed. In case of a system shutdown, the progress of parity regeneration must be kept across reboots. Otherwise, the process has to start all over again.

To avoid the restart process, parity regeneration is *checkpointed*. This means that the offset up to which the parity has been regenerated is saved in the configuration database. The `-o checkpoint=size` option controls how often the checkpoint is saved. If the option is not specified, the default checkpoint size is used.

Because saving the checkpoint offset requires a transaction, making the checkpoint size too small can extend the time required to regenerate parity. After a system reboot, a RAID-5 volume that has a checkpoint offset smaller than the volume length starts a parity resynchronization at the checkpoint offset.

Log Plex Recovery

RAID-5 log plexes can become detached due to disk failures. These RAID-5 logs can be reattached by using the `att` keyword for the `vxplex` command. To reattach the failed RAID-5 log plex, use the following command:

```
# vxplex att r5vol r5vol-11
```

Stale Subdisk Recovery

Stale subdisk recovery is usually done at volume start time. However, the process doing the recovery can crash, or the volume may be started with an option such as `-o delayrecover` that prevents subdisk recovery. In addition, the disk on which the subdisk resides can be replaced without recovery operations being performed. In such cases, you can perform subdisk recovery using the `vxvol recover` command. For example, to recover the stale subdisk in the RAID-5 volume shown in the figure “[Invalid RAID-5 Volume](#)” on page 13, use the following command:

```
# vxvol recover r5vol disk01-00
```

A RAID-5 volume that has multiple stale subdisks can be recovered in one operation. To recover multiple stale subdisks, use the `vxvol recover` command on the volume, as follows:

```
# vxvol recover r5vol
```



Recovery After Moving RAID-5 Subdisks

When RAID-5 subdisks are moved and replaced, the new subdisks are marked as `STALE` in anticipation of recovery. If the volume is active, the `vxsd` command may be used to recover the volume. If the volume is not active, it is recovered when it is next started. *The RAID-5 volume is degraded for the duration of the recovery operation.*

Any failure in the stripes involved in the move makes the volume unusable. The RAID-5 volume can also become invalid if its parity becomes stale. To avoid this occurring, `vxsd` does not allow a subdisk move in the following situations:

- ◆ a stale subdisk occupies any of the same stripes as the subdisk being moved
- ◆ the RAID-5 volume is stopped but was not shut down cleanly; that is, the parity is considered stale
- ◆ the RAID-5 volume is active and has no valid log areas

Only the third case can be overridden by using the `-o force` option.

Subdisks of RAID-5 volumes can also be split and joined by using the `vxsd split` command and the `vxsd join` command. These operations work the same way as those for mirrored volumes.

Note RAID-5 subdisk moves are performed in the same way as subdisk moves for other volume types, but without the penalty of degraded redundancy.

Starting RAID-5 Volumes

When a RAID-5 volume is started, it can be in one of many states. After a normal system shutdown, the volume should be clean and require no recovery. However, if the volume was not closed, or was not unmounted before a crash, it can require recovery when it is started, before it can be made available. This section describes actions that can be taken under certain conditions.

Under normal conditions, volumes are started automatically after a reboot and any recovery takes place automatically or is done through the `vxrecover` command.

Unstartable RAID-5 Volumes

A RAID-5 volume is unusable if some part of the RAID-5 plex does not map the volume length:

- ◆ the RAID-5 plex cannot be sparse in relation to the RAID-5 volume length
- ◆ the RAID-5 plex does not map a region where two subdisks have failed within a stripe, either because they are stale or because they are built on a failed disk

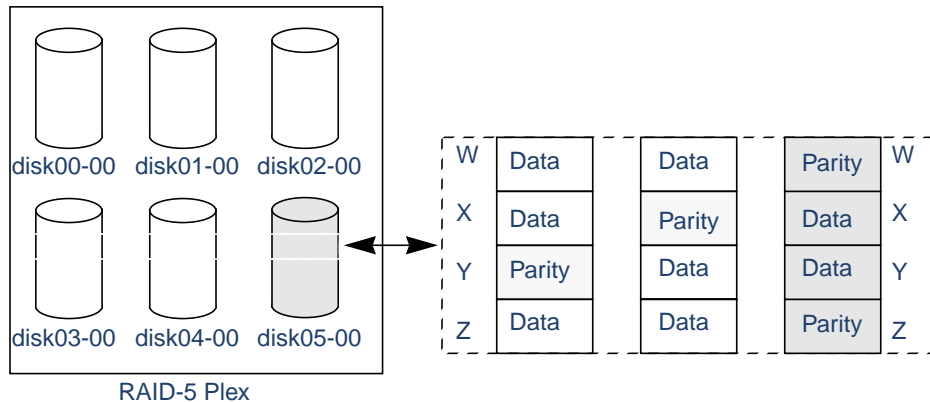
When this occurs, the `vxvol start` command returns the following error message:

```
vxvm:vxvol: ERROR: Volume r5vol is not startable; RAID-5 plex does
not map entire volume length.
```

At this point, the contents of the RAID-5 volume are unusable.

Another possible way that a RAID-5 volume can become unstartable is if the parity is stale and a subdisk becomes detached or stale. This occurs because within the stripes that contain the failed subdisk, the parity stripe unit is invalid (because the parity is stale) *and* the stripe unit on the bad subdisk is also invalid. The situation shown in “[Invalid RAID-5 Volume](#)” illustrates a RAID-5 volume that has become invalid due to stale parity and a failed subdisk.

Invalid RAID-5 Volume



This example shows four stripes in the RAID-5 array. All parity is stale and subdisk `disk05-00` has failed. This makes stripes X and Y unusable because two failures have occurred within those stripes.

This qualifies as two failures within a stripe and prevents the use of the volume. In this case, the output display from the `vxvol start` command is as follows:

```
vxvm:vxvol: ERROR: Volume r5vol is not startable; some subdisks are
unusable and the parity is stale.
```

This situation can be avoided by *always* using two or more RAID-5 log plexes in RAID-5 volumes. RAID-5 log plexes prevent the parity within the volume from becoming stale which prevents this situation (see “[System Failures](#)” on page 6 for details).



Forcibly Starting RAID-5 Volumes

You can start a volume even if subdisks are marked as stale. For example, if a stopped volume has stale parity and no RAID-5 logs and a disk becomes detached and then reattached.

The subdisk is considered stale even though the data is not out of date (because the volume was in use when the subdisk was unavailable) and the RAID-5 volume is considered invalid. To prevent this case, always have multiple valid RAID-5 logs associated with the array. However, this is not always possible.

To start a RAID-5 volume with stale subdisks, you can use the `-f` option with the `vxvol start` command. This causes all stale subdisks to be marked as non-stale. Marking takes place before the `start` operation evaluates the validity of the RAID-5 volume and what is needed to start it. Also, you can mark individual subdisks as non-stale by using the following command:

```
# vxmend fix unstale subdisk
```

- ◆ If some subdisks are stale and need recovery, and if valid logs exist, the volume is enabled by placing it in the `ENABLED` kernel state and the volume is available for use during the subdisk recovery. Otherwise, the volume kernel state is set to `DETACHED` and it is not available during subdisk recovery.

This is done because if the system were to crash or the volume was ungracefully stopped while it was active, the parity becomes stale, making the volume unusable. If this is undesirable, the volume can be started with the `-o unsafe start` option.

Caution The `-o unsafe start` option is considered dangerous, as it can make the contents of the volume unusable. It is therefore not recommended.

- ◆ The volume state is set to `RECOVER` and stale subdisks are restored. As the data on each subdisk becomes valid, the subdisk is marked as no longer stale.

If any subdisk recovery fails and there are no valid logs, the volume start is aborted because the subdisk remains stale and a system crash makes the RAID-5 volume unusable. This can also be overridden by using the `-o unsafe start` option.

Caution The `-o unsafe start` option is considered dangerous, as it can make the contents of the volume unusable. It is therefore not recommended.

If the volume has valid logs, subdisk recovery failures are noted but they do not stop the start procedure.

- ◆ When all subdisks have been recovered, the volume is placed in the `ENABLED` kernel state and marked as `ACTIVE`. It is now started.

Recovering from Incomplete Disk Group Moves

If the system crashes or a subsystem fails while a disk group move, split or join operation is being performed, VxVM attempts either to reverse or to complete the operation when the system is restarted or the subsystem is repaired. Whether the operation is reversed or completed depends on how far it had progressed.

Automatic recovery depends on being able to import both the source and target disk groups. If this is not possible (for example, if one of the disk groups has been imported on another host), perform the following steps to recover the disk group:

1. Use the `vxprint` command to examine the configuration of both disk groups. Objects in disk groups whose move is incomplete have their TUTIL0 fields set to MOVE.
2. Enter the following command to attempt completion of the move:

```
# vxdg recover sourcedg
```

This operation fails if one of the disk groups cannot be imported because it has been imported on another host or because it does not exist:

```
vxvm: vxdg: ERROR: diskgroup: Disk group does not exist
```

If the recovery fails, perform one of the following steps as appropriate.

- ❖ If the disk group has been imported on another host, export it from that host, and import it on the current host. If all the required objects already exist in either the source or target disk group, use the following command to reset the MOVE flags in that disk group:


```
# vxdg -o clean recover diskgroup1
```

Use the following command on the other disk group to remove the objects that have TUTIL0 fields marked as MOVE:

```
# vxdg -o remove recover diskgroup2
```
- ❖ If only one disk group is available to be imported, use the following command to reset the MOVE flags on this disk group:


```
# vxdg -o clean recover diskgroup
```



Recovery from DCO Log Volume Failure

Persistent FastResync uses a data change object (DCO) log volume to perform tracking of changed regions in a volume. If an error occurs while reading or writing a DCO log volume, it is detached and the `badlog` flag is set on the DCO. (You can use one of the options `-a`, `-F` or `-m` to `vxprint` to check if the `badlog` flag is set on a DCO.) All further writes to the volume are not tracked by the DCO.

To recover the DCO log volume, perform the following steps:

1. Correct the problem that caused the I/O failure.
2. Use the following command to remove the `badlog` flag from the DCO:

```
# vxddco -g diskgroup -o force enable dco
```

3. Restart the DCO log volume using the following command:

```
# vxvol -g diskgroup start dco_log_vol
```

4. Use the `vxassist snapclear` command to clear the FastResync maps for the original volume and for all its snapshots. This ensures that potentially stale FastResync maps are not used when the snapshots are snapped back (a full resynchronization is performed). FastResync tracking is re-enabled for any subsequent snapshots of the volume.

Caution You must use the `vxassist snapclear` command on all the snapshots of the volume after removing the `badlog` flag from the DCO. Otherwise, data may be lost or corrupted when the snapshots are snapped back.

If a volume and its snapshot volume are in the same disk group, the following command clears the FastResync maps for both volumes:

```
# vxassist -g diskgroup snapclear volume snap_obj_to_snapshot
```

Here *snap_obj_to_snapshot* is the name of the snap object associated with *volume* that points to the snapshot volume.

If a snapshot volume and the original volume are in different disk groups, you must perform a separate `snapclear` operation on each volume:

```
# vxassist -g diskgroup1 snapclear volume snap_obj_to_snapshot  
# vxassist -g diskgroup2 snapclear snapvol snap_obj_to_volume
```

Here *snap_obj_to_volume* is the name of the snap object associated with the snapshot volume, *snapvol*, that points to the original volume.

5. To snap back the snapshot volume on which you performed a `snapclear` in the previous step, use the following command (after using the `vxchg move` command to move the snapshot volume back to the original disk group, if necessary):

```
# vxplex -f -g diskgroup snapback volume snapvol_plex
```

Note You cannot use `vxassist snapback` because the `snapclear` operation removes the snapshot association information.

The following command sequence demonstrates how to recover the DCO log volume that tracks the top-level volume `vol1` in the disk group `egd`, and also how to snap back the snapshot volume, `SNAP-vol1`, with `vol1`:

```
# vxdco -g egdg -o force enable vol1_dco
# vxvol -g egdg start vol1_dco
# vxassist -g egdg snapclear vol1 SNAP-vol1_snp
# vxplex -g egdg snapback vol1 SNAP-vol1-01
```

Here `vol1_dco` is the DCO associated with `vol1`, `SNAP-vol1_snp` is the snap object associated with `vol1` that points to the snapshot `SNAP-vol1`, and `SNAP-vol1-01` is the snapshot plex that is snapped back with `vol1`.

For more information, see the `vxassist(1M)` and `vxdco(1M)` manual pages.





Recovery from Boot Disk Failure

2

Introduction

VERITAS Volume Manager (VxVM) protects systems from disk and other hardware failures and helps you to recover from such events. This chapter describes recovery procedures and information to help you prevent loss of data or system access due to the failure of the boot (`root`) disk. It also includes procedures for repairing the `root (/)` and `usr` file systems.

For information about recovering volumes and their data on non-boot disks, see [“Recovery from Hardware Failure”](#) on page 1.

For more information about protecting your system, see the *VERITAS Volume Manager Installation Guide*.

Possible root, swap, and usr Configurations

During installation, it is possible to set up a variety of configurations for the `root (/)` and `usr` file systems, and for `swap`. The following cases are possible:

- ◆ `usr` is a directory under `/` and no separate partition is allocated for it. In this case, `usr` becomes part of the `rootvol` volume when the root disk is encapsulated and put under VERITAS Volume Manager control.
- ◆ `usr` is on a separate partition from the root partition on the root disk. In this case, a separate volume is created for the `usr` partition. `vxmirror` mirrors the `usr` volume on the destination disk.
- ◆ `usr` is on a disk other than the root disk. In this case, a volume is created for the `usr` partition only if you use VxVM to encapsulate the disk. Note that encapsulating the root disk and having mirrors of the root volume is ineffective in maintaining the availability of your system if the separate `usr` partition becomes inaccessible for any reason. For maximum availability of the system, it is recommended that you encapsulate both the `root` disk and the disk containing the `usr` partition, and have mirrors for the `usr`, `rootvol`, and `swapvol` volumes.



The `rootvol` volume must exist in the `rootdg` disk group. See “Boot-time Volume Restrictions” in the “Administering Disks” chapter of the *VERITAS Volume Manager Administrator’s Guide* for information on `rootvol` and `usr` volume restrictions.

VxVM allows you to put `swap` partitions on any disk; it does not need an initial `swap` area during early phases of the boot process. By default, the VERITAS Volume Manager installation chooses partition 0 on the selected root disk as the `root` partition, and partition 1 as the `swap` partition. However, it is possible to have the `swap` partition on a partition not located on the root disk. In such cases, you are advised to encapsulate that disk and create mirrors for the `swap` volume. If you do not do this, damage to the `swap` partition eventually causes the system to crash. It may be possible to boot the system, but having mirrors for the `swapvol` volume prevents system failures.

Booting from Alternate Boot Disks

If the root disk is encapsulated and mirrored, you can use one of its mirrors to boot the system if the primary boot disk fails. To boot the system after failure of the primary boot disk on a SPARC system, follow these steps:

1. Check that the EEPROM variable `use-nvramrc?` is set to `true` by entering the following command at the boot prompt:

```
ok printenv use-nvramrc?
```

If set to `true`, this variable allows the use of alternate boot disks. To set the value of `use-nvramrc?` to `true`, enter the following command at the boot prompt:

```
ok setenv use-nvramrc? true
```

If `use-nvramrc?` is set to `false`, the system fails to boot from the `devalias` and displays an error message such as the following:

```
Rebooting with command: boot vx-mirdisk
Boot device: /pci@1f,4000/scsi@3/disk@0,0 File and args:vx-mirdisk
boot: cannot open vx-mirdisk
Enter filename [vx-mirdisk]:
```

2. Check for available boot disk aliases using the following command at the boot prompt:

```
ok devalias
```

Suitable mirrors of the root disk are listed with names of the form `vx-diskname`.

3. Enter this command:

```
ok boot alias
```

where *alias* is the name of an alternate root mirror found from the previous step.

If a selected disk contains a root mirror that is stale, `vxconfigd` displays an error stating that the mirror is unusable and lists any non-stale alternate bootable disks.

More information about the boot process may be found in “[The Boot Process on SPARC Systems](#)” on page 21

The Boot Process on SPARC Systems

A Sun SPARC system prompts for a boot command unless the `autoboot` flag has been set in the nonvolatile storage area used by the firmware. Machines with older PROMs have different prompts than that for the newer V2 and V3 versions. These newer versions of PROM are also known as OpenBoot PROMs (OBP). The `boot` command syntax for the newer types of PROMs is:

```
ok boot [OBP names] [filename] [boot-flags]
```

OBP names specify the OpenBoot PROM designations. For example, on Desktop SPARC systems, the designation `sbus/esp@0,800000/sd@3,0:a` indicates a SCSI disk (`sd`) at target 3, lun 0 on the SCSI bus, with the `esp` host bus adapter plugged into slot 0.

Note You can use VERITAS Volume Manager boot disk alias names instead of OBP names. Example aliases are `vx-rootdisk` or `vx-disk01`. To list the available boot devices, use the `devalias` command at the OpenBoot prompt.

filename is the name of a file that contains the kernel. The default is `/kernel/unix` in the root partition. If necessary, you can specify another program (such as `/stand/diag`) by specifying the `-a` flag. (Some versions of the firmware allow the default filename to be saved in the nonvolatile storage area of the system.)

Note Do not boot a system running VxVM with rootability enabled using all the defaults presented by the `-a` flag. See “[Restore a Copy of /etc/system on the root Disk](#)” on page 28 for the correct responses.

Boot flags are not interpreted by the `boot` program. The `boot` program passes all *boot-flags* to the file identified by *filename*. See the `kernel (1)` and `kadb (1M)` manual pages for information on the options available with the default standalone program, `/kernel/unix`.



Hot-Relocation and Boot Disk Failure

If the boot (`root`) disk fails and it is mirrored, hot-relocation automatically attempts to replace the failed `root` disk mirror with a new mirror. To achieve this, hot-relocation uses a surviving mirror of the root disk to create a new mirror, either on a spare disk, or on a disk with sufficient free space. This ensures that there are always at least two mirrors of the `root` disk that can be used for booting. The hot-relocation daemon also calls the `vxbootsetup` utility to configure the disk with the new mirror as a bootable disk.

Hot-relocation can fail for a `root` disk if the `rootdg` disk group does not contain sufficient spare or free space to fit the volumes from the failed `root` disk. The `rootvol` and `swapvol` volumes require contiguous disk space. If the `root` volume and other volumes on the failed `root` disk cannot be relocated to the same new disk, each of these volumes may be relocated to different disks.

Mirrors of `rootvol` and `swapvol` volumes must be cylinder-aligned. This means that they can only be created on disks that have enough space to allow their subdisks to begin and end on cylinder boundaries. Hot-relocation fails to create the mirrors if these disks are not available.

Unrelocating Subdisks to a Replacement Boot Disk

When a boot disk is encapsulated, the `root` file system and other system areas, such as the swap partition, are made into volumes. VxVM creates a private region using part of the existing swap area, which is usually located in the middle of the disk. However, when a disk is initialized as a VM disk, VxVM creates the private region at the beginning of the disk.

If a mirrored encapsulated boot disk fails, hot-relocation creates new copies of its subdisks on a spare disk. The name of the disk that failed and the offsets of its component subdisks are stored in the subdisk records as part of this process. After the failed boot disk is replaced with one that has the same storage capacity, it is “initialized” and added back to the disk group. `vxunreloc` can be run to move all the subdisks back to the disk. However, the difference of the disk layout between an initialized disk and an encapsulated disk affects the way the offset into a disk is calculated for each unrelocated subdisk. Use the `-f` option to `vxunreloc` to move the subdisks to the disk, but not to the same offsets. For this to be successful, the replacement disk should be at least 2 megabytes larger than the original boot disk.

`vxunreloc` makes the new disk bootable after it moves all the subdisks to the disk.

Note The system dump device is usually configured to be the swap partition of the `root` disk. Whenever a swap subdisk is moved (by hot-relocation, or using `vxunreloc`) from one disk to another, the dump device must be re-configured on the new disk.

In Solaris 2.6 and earlier releases, the name of the dump device is stored in the `dumpfile` structure. Use the following command to discover its setting:

```
# echo dumpfile+0x10/s | adb -k /dev/ksyms /dev/mem
```

This displays output similar to the following:

```
physmem 3d24
dumpfile+0x10: /dev/dsk/c0t0d0s1
```

In this example, the dump device is configured to be `/dev/dsk/c0t0d0s1`. To change this setting, shut down and reboot the system. This configures the first swap partition as the dump device.

In Solaris 7, and later releases, use the `dumppadm` command to view and set the dump device. For details, see the `dumppadm(1M)` manual page.

Recovery from Boot Failure

While there are many types of failures that can prevent a system from booting, the same basic procedure can be taken to bring the system up. When a system fails to boot, you should first try to identify the failure by the evidence left behind on the screen and then attempt to repair the problem (for example, by turning on a drive that was accidentally powered off). If the problem is one that cannot be repaired (such as data errors on the boot disk), boot the system from an alternate boot disk that contains a mirror of the `root` volume, so that the damage can be repaired or the failing disk can be replaced.

The following sections outline some possible failures and provides instructions on the corrective actions:

- ◆ “[Boot Device Cannot be Opened](#)”
- ◆ “[Cannot Boot From Unusable or Stale Plexes](#)” on page 24
- ◆ “[Invalid UNIX Partition](#)” on page 26
- ◆ “[Incorrect Entries in /etc/vfstab](#)” on page 26
- ◆ “[Missing or Damaged /etc/system](#)” on page 28

Boot Device Cannot be Opened

Early in the boot process, immediately following system initialization, there may be messages similar to the following:

```
SCSI device 0,0 is not responding
Can't open boot device
```



This means that the system PROM was unable to read the boot program from the boot drive. Common causes for this problem are:

- ◆ The boot disk is not powered on.
- ◆ The SCSI bus is not terminated.
- ◆ There is a controller failure of some sort.
- ◆ A disk is failing and locking the bus, preventing any disks from identifying themselves to the controller, and making the controller assume that there are no disks attached.

The first step in diagnosing this problem is to check carefully that everything on the SCSI bus is in order. If disks are powered off or the bus is unterminated, correct the problem and reboot the system. If one of the disks has failed, remove the disk from the bus and replace it.

If no hardware problems are found, the error is probably due to data errors on the boot disk. In order to repair this problem, attempt to boot the system from an alternate boot disk (containing a mirror of the root volume). If you are unable to boot from an alternate boot disk, there is still some type of hardware problem. Similarly, if switching the failed boot disk with an alternate boot disk fails to allow the system to boot, this also indicates hardware problems.

Cannot Boot From Unusable or Stale Plexes

If a disk is unavailable when the system is running, any mirrors of volumes that reside on that disk become stale. This means that the data on that disk is inconsistent relative to the other mirrors of that volume. During the boot process, the system accesses only one copy of the `root` volume (the copy on the boot disk) until a complete configuration for this volume can be obtained.

If it turns out that the plex of this volume that was used for booting is stale, the system must be rebooted from an alternate boot disk that contains non-stale plexes. This problem can occur, for example, if the system was booted from one of the disks made bootable by VxVM with the original boot disk turned off. The system boots normally, but the plexes that reside on the unpowered disk are stale. If the system reboots from the original boot disk with the disk turned back on, the system boots using that stale plex.

Another possible problem can occur if errors in the VERITAS Volume Manager headers on the boot disk prevent VxVM from properly identifying the disk. In this case, VxVM does not know the name of that disk. This is a problem because plexes are associated with disk names, so any plexes on the unidentified disk are unusable.

A problem can also occur if the root disk has a failure that affects the root volume plex. At the next boot attempt, the system still expects to use the failed root plex for booting. If the root disk was mirrored at the time of the failure, an alternate root disk (with a valid root plex) can be specified for booting.

If any of these situations occur, the configuration daemon, `vxconfigd`, notes it when it is configuring the system as part of the `init` processing of the boot sequence. `vxconfigd` displays a message describing the error and what can be done about it, and then halts the system. For example, if the plex `rootvol-01` of the root volume `rootvol` on disk `rootdisk` is stale, `vxconfigd` may display this message:

```
vxvm:vxconfigd: Warning Plex rootvol-01 for root volume is stale or
unusable.
vxvm:vxconfigd: Error: System boot disk does not have a valid root
plex
Please boot from one of the following disks:
Disk: disk01           Device: c0t1d0s2
vxvm:vxconfigd:      Error: System startup failed
The system is down.
```

This informs the administrator that the alternate boot disk named `disk01` contains a usable copy of the root plex and should be used for booting. When this message is displayed, reboot the system from the alternate boot disk as described in [“Booting from Alternate Boot Disks”](#) on page 20.

Once the system has booted, the exact problem needs to be determined. If the plexes on the boot disk were simply stale, they are caught up automatically as the system comes up. If, on the other hand, there was a problem with the private area on the disk or the disk failed, you need to re-add or replace the disk.

If the plexes on the boot disk are unavailable, you should receive mail from VERITAS Volume Manager utilities describing the problem. Another way to determine the problem is by listing the disks with the `vxdisk` utility. In the above example, if the problem is a failure in the private area of root disk (such as due to media failures or accidentally overwriting the VERITAS Volume Manager private region on the disk, `vxdisk list` shows this display:

DEVICE	TYPE	DISK	GROUP	STATUS
-	-	rootdisk	rootdg	failed was: c0t3d0s2
c0t1d0s2	sliced	disk01	rootdg	ONLINE



Invalid UNIX Partition

Once the boot program has loaded, it attempts to access the boot disk through the normal UNIX partition information. If this information is damaged, the boot program fails with an error such as:

```
File just loaded does not appear to be executable
```

If this message appears during the boot attempt, the system should be booted from an alternate boot disk. While booting, most disk drivers display errors on the console about the invalid UNIX partition information on the failing disk. The messages are similar to this:

```
WARNING: unable to read label
WARNING: corrupt label_sdo
```

This indicates that the failure was due to an invalid disk partition. You can attempt to re-add the disk as described in [“Re-Adding a Failed Boot Disk”](#) on page 34. However, if the reattach fails, then the disk needs to be replaced as described in [“Replacing a Failed Boot Disk”](#) on page 35.

Incorrect Entries in /etc/vfstab

When the `root` disk is encapsulated and put under VERITAS Volume Manager control, as part of the normal encapsulation process, volumes are created for all of the partitions on the disk. VxVM modifies the `/etc/vfstab` to use the corresponding volumes instead of the disk partitions. Care should be taken while editing the `/etc/vfstab` file manually, and you should always make a backup copy before committing any changes to it. The most important entries are those corresponding to `/` and `/usr`. The `vfstab` that existed prior to VERITAS Volume Manager installation is saved in `/etc/vfstab.prevvm`.

Damaged root (/) Entry in /etc/vfstab

If the entry in `/etc/vfstab` for the `root` file system (`/`) is lost or is incorrect, the system boots in single-user mode. Messages similar to the following are displayed:

```
File just loaded does not appear to be executable
```

It is recommended that you run `fsck` at this point:

```
# fsck /dev/vx/rdisk/rootvol
```

At this point in the boot process, / is mounted read-only, not read/write. Since the entry in /etc/vfstab was either incorrect or deleted, mount / as read/write manually, using this command:

```
# mount -o remount /dev/vx/dsk/rootvol
```

After mounting / as read/write, exit the shell. The system prompts for a new run level. For multi-user mode, enter run level 3:

```
ENTER RUN LEVEL (0-6,s or S): 3
```

Restore the entry in /etc/vfstab for / after the system boots.

Damaged /usr Entry in /etc/vfstab

The /etc/vfstab file has an entry for /usr only if /usr is located on a separate disk partition. After encapsulation of the disk containing the /usr partition, VxVM changes the entry in /etc/vfstab to use the corresponding volume.

In the event of loss of the entry for /usr from /etc/vfstab, the system cannot be booted (even if you have mirrors of the /usr volume). In this case, boot the system from the CD-ROM and restore /etc/vfstab using the following procedure:

1. Boot the operating system into single-user mode from its installation CD-ROM using the following command at the boot prompt:

```
ok boot cdrom -s
```

2. Mount /dev/dsk/c0t0d0s0 on a suitable mount point such as /a or /mnt:

```
# mount /dev/dsk/c0t0d0s0 /a
```

3. Edit /a/etc/vfstab, and ensure that there is an entry for the /usr file system, such as the following:

```
/dev/vx/dsk/usr /dev/vx/rdisk/usr /usr ufs 1 yes -
```

4. Shut down and reboot the system from the same root partition on which the vfstab file was restored.



Missing or Damaged /etc/system

Do not edit any entries in `/etc/system` that have been added by VxVM. All VERITAS Volume Manager entries are enclosed between `vxvm_START` and `vxvm_END` comments.

Make a copy of `/etc/system` in the `root` file system before making any changes to it. The saved copy can be specified to the `boot` program if changes to the new `/etc/system` file are incorrect. To specify the saved system file to the `boot` program, follow the procedure in the next section.

Restore a Copy of /etc/system on the root Disk

If the `/etc/system` file is damaged and a saved copy of the `/etc/system` file is available, the system can be booted with VERITAS Volume Manager rootability as follows:

1. Boot the system with the following command:

```
ok boot -a
```

2. Press Return to accept the default for all prompts *except* the following:

- a. The default pathname for the kernel program, `/kernel/unix`, may not be appropriate for your system's architecture. If this is so, enter the correct pathname, such as `/platform/sun4u/kernel/unix`, at the following prompt:

```
Enter filename [/kernel/unix]: /platform/sun4u/kernel/unix
```

- b. Enter the name of the saved system file, such as `/etc/system.save` at the following prompt:

```
Name of system file [/etc/system]: /etc/system.save
```

- c. Enter `/pseudo/vxio@0:0` as the physical name of the root device at the following prompt:

```
Enter physical name of root device  
[...]: /pseudo/vxio@0:0
```

Copy of /etc/system is not Available on the root Disk

If the `/etc/system` file is damaged or missing, and a saved copy of the system file is not available on the `root` disk, the system cannot be booted with the VERITAS Volume Manager rootability feature turned on.

The following procedure assumes the device name of the root disk to be `c0t0d0s2`, and that the `root (/)` file system is on partition `s0`.

To boot the system without VERITAS Volume Manager rootability and restore the `/etc/system` file:

1. Boot the operating system into single-user mode from its installation CD-ROM using the following command at the boot prompt:

```
ok boot cdrom -s
```

2. Mount `/dev/dsk/c0t0d0s0` on a suitable mount point such as `/a` or `/mnt`:

```
# mount /dev/dsk/c0t0d0s0 /a
```

3. If a backup copy of `/etc/system` is available, restore this as the file `/a/etc/system`. If a backup copy is not available, create a new `/a/etc/system` file. Ensure that `/a/etc/system` contains the following entries that are required by VxVM:

```
* vxvm_START
forceload: drv/driver
...
forceload: drv/vxio
forceload: drv/vxspec
forceload: drv/vxdmp
rootdev:/pseudo/vxio@0:0
set vxio:vol_rootdev_is_volume=1
* vxvm_END
```

Lines of the form `forceload: drv/driver` are used to load the drivers forcibly that are required for the root mirror disks. Example *driver* names are `pci`, `sd`, `ssd`, `dad` and `ide`. To find out the names of the drivers, use the `ls` command to obtain a long listing of the special files that correspond to the devices used for the root disk, for example:

```
# ls -al /dev/dsk/c0t0d0s2
```

This produces output similar to the following (with irrelevant detail removed):

```
lrwxrwxrwx ... /dev/dsk/c0t0d0s2 ->
.../devices/pci@1f,0/pci@1/pci@1/SUNW,isp2wo@4/sd@0,0:c
```

This example would require lines to force load both the `pci` and the `sd` drivers:

```
forceload: drv/pci
forceload: drv/sd
```

4. Shut down and reboot the system from the same root partition on which the `system` file was restored.



Repairing root or /usr File Systems on Mirrored Volumes

If the `root` or `/usr` file system is defined on a mirrored volume, errors in the partition that underlies one of the mirrors can result in data corruption or system errors at boot time (when VxVM is started and assumes that the mirrors are synchronized).

Two alternate workarounds exist for this situation:

- ◆ Mount one plex of the `root` or `/usr` file system, repair it, unmount it, and use `dd` to copy the fixed plex to all other plexes. This procedure is not recommended as it can be error prone.
- ◆ Restore the system from a valid backup tape. This procedure is described in the following section. It does not require the operating system to be re-installed from the base CD-ROM. It provides a simple, efficient, and reliable means of recovery when both the root disk and its mirror are damaged.

Recovering a root Disk and root Mirror from Backup Tape

This procedure assumes that you have:

- ◆ A listing of the partition table for the original `root` disk before you encapsulated it.
- ◆ A current full backup of all the file systems on the original `root` disk that was under VERITAS Volume Manager control. If the `root` file system is of type `ufs`, you can back it up using the `ufsdump` command. See the `ufsdump(1M)` manual page for more information.
- ◆ A new boot disk installed to replace the original failed boot disk if the original boot disk was physically damaged.

This procedure requires the reinstallation of the `root` disk. To prevent the loss of data on disks not involved in the reinstallation, only involve the `root` disk in the reinstallation procedure.

Several of the automatic options for installation access disks other than the root disk without requiring confirmation from the administrator. Therefore, disconnect all other disks containing volumes from the system prior to starting this procedure. This will ensure that these disks are unaffected by the reinstallation. Reconnect the disks after completing the procedure.

The following procedure assumes the device name of the new `root` disk to be `c0t0d0s2`, and that you need to recover both the `root` (`/`) file system on partition `s0`, and the `/usr` file system on partition `s6`. If your system does not have a separate `/usr` file system, omit steps 7 and 8.

1. Boot the operating system into single-user mode from its installation CD-ROM using the following command at the boot prompt:

```
ok boot cdrom -s
```

2. Use the `format` command to create partitions on the new root disk (`c0t0d0s2`). These should be identical in size to those on the original root disk before encapsulation unless you are using this procedure to change their sizes. If you change the size of the partitions, ensure that they are large enough to store the data that is restored to them. See the `format(1M)` manual page for more information.

Note A maximum of five partitions may be created for file systems or swap areas as encapsulation reserves two partitions for VERITAS Volume Manager `private` and `public` regions.

3. Use the `mkfs` command to make new file systems on the `root` and `usr` partitions that you created in the previous step. For example, to make a `ufs` file system on the `root` partition, enter:

```
# mkfs -F ufs /dev/dsk/c0t0d0s0
```

See the `mkfs(1M)` and `mkfs_ufs(1M)` manual pages for more information.

4. Mount `/dev/dsk/c0t0d0s0` on a suitable mount point such as `/a` or `/mnt`:

```
# mount /dev/dsk/c0t0d0s0 /a
```

5. Restore the `root` file system from tape into the `/a` directory hierarchy. For example, if you used `ufsdump` to back up the file system, use the `ufsrestore` command to restore it. See the `ufsrestore(1M)` manual page for more information.

6. Use the `installboot` command to install a bootblock device on `/a`.

7. Use the `mkdir` command to create a suitable mount point, such as `/a/usr/`, and mount `/dev/dsk/c0t0d0s6` on it:

```
# mkdir -p /a/usr
```

```
# mount /dev/dsk/c0t0d0s6 /a/usr
```

8. Restore the `/usr` file system from tape into the `/a/usr` directory hierarchy.



9. Disable startup of VxVM by modifying files in the restored root file system as follows:

- a. Create the file `/a/etc/vx/reconfig.d/state.d/install-db` to prevent the configuration daemon, `vxconfigd`, from starting:

```
# touch /a/etc/vx/reconfig.d/state.d/install-db
```

- b. Copy `/a/etc/system` to a backup file such as `/a/etc/system.old`.

- c. Comment out the following lines from `/a/etc/system` by putting a `*` character in front of them:

```
rootdev:/pseudo/vxio@0:0
set vxio:vol_rootdev_is_volume=1
```

These lines should then read:

```
* rootdev:/pseudo/vxio@0:0
* set vxio:vol_rootdev_is_volume=1
```

- d. Copy `/a/etc/vfstab` to a backup file such as `/a/etc/vfstab.old`.

- e. Edit `/a/etc/vfstab`, and replace the volume device names (beginning with `/dev/vx/dsk`) for the `/` and `/usr` file system entries with their standard disk devices, `/dev/dsk/c0t0d0s0` and `/dev/dsk/c0t0d0s6`. For example, replace the following lines:

```
/dev/vx/dsk/rootvol /dev/vx/rdisk/rootvol / ufs 1 no -
/dev/vx/dsk/usrvol /dev/vx/rdisk/usrvol /usr ufs 1 yes -
```

with this line:

```
/dev/dsk/c0t0d0s0 /dev/rdsk/c0t0d0s0 / ufs 1 no -
/dev/dsk/c0t0d0t6 /dev/rdsk/c0t0d0s6 /usr ufs 1 yes -
```

10. Shut down the system cleanly using the `init 0` command, and reboot from the new root disk. The system comes up thinking that VxVM is *not* installed.

The next step in the procedure depends on whether there are root disk mirrors in the old rootdg:

- ◆ If there are other disks in the old rootdg that are *not* used as root disk mirrors, perform only step 11.
- ◆ If there are only root disk mirrors in the old rootdg, perform only step 12.

11. If there are other disks in the old rootdg that are *not* used as root disk mirrors, follow these steps to bring in the old rootdg (minus the boot disk which VxVM will think has failed) and set up the new boot disk.

- a. Remove files involved with the installation that are no longer needed:

```
# rm -r /etc/vx/reconfig.d/state.d/installdb
```

- b. Start the VERITAS Volume Manager I/O daemons:

```
# vxiod set 10
```

- c. Start the VERITAS Volume Manager configuration daemon in disabled mode:

```
# vxconfigd -m disable
```

- d. Initialize the volboot file:

```
# vxdctl init
```

- e. Enable vxconfigd:

```
# vxdctl enable
```

Steps a through e enable the old rootdg excluding the root disk which VxVM interprets as failed.

- f. Use the vxedit command (or the VERITAS Volume Manager Storage Administrator (VMSA)) to remove the old root disk volumes and the root disk itself from VERITAS Volume Manager control.
- g. Use the vxdiskadm command to encapsulate the new root disk and initialize any disks that are to serve as root disk mirrors. After the required reboot, mirror the root disk onto the root disk mirrors.

12. If there are only root disk mirrors in the old rootdg:

- a. Run the vxinstall command to encapsulate the new boot disk, and initialize the root disk mirrors.
- b. After the required reboot, mirror the root disk onto the root disk mirrors.

Re-Adding and Replacing Boot Disks

Data that is not critical for booting the system is only accessed by VxVM after the system is fully operational, so it does not have to be located in specific areas. VxVM can find it. However, boot-critical data must be placed in specific areas on the bootable disks for the boot process to find it.

On some systems, the controller-specific actions performed by the disk controller in the process and the system BIOS constrain the location of this critical data.



If a boot disk fails, one of the following procedures can be used to correct the problem:

- ◆ If the errors are transient or correctable, re-use the same disk. This is known as *re-adding* a disk. In some cases, reformatting a failed disk or performing a surface analysis to rebuild the alternate-sector mappings are sufficient to make a disk usable for re-addition.
- ◆ If the disk has failed completely, replace it.

The following sections describe how to re-add or replace a failed boot disk.

Re-Adding a Failed Boot Disk

Re-adding a disk is the same procedure as replacing the disk, except that the same physical disk is used. Normally, a disk that needs to be re-added has been *detached*. This means that VxVM has detected the disk failure and has ceased to access the disk.

Note Your system may use a *device name* or *path* that differs from the examples. See “Disk Devices” in the “Administering Disks” chapter of the *VERITAS Volume Manager Administrator’s Guide* for more information on device names.

For example, consider a system that has two disks, `disk01` and `disk02`, which are normally mapped into the system configuration during boot as disks `c0t0d0s2` and `c0t1d0s2`, respectively. A failure has caused `disk01` to become detached. This can be confirmed by listing the disks with the `vxdisk list` utility with this command:

```
# vxdisk list
```

`vxdisk` displays this (example) list:

DEVICE	TYPE	DISK	GROUP	STATUS
<code>c0t0d0s2</code>	sliced	-	-	error
<code>c0t1d0s2</code>	sliced	disk02	rootdg	online
-	-	disk01	rootdg	failed was: <code>c0t0d0s2</code>

Note that the disk `disk01` has no device associated with it, and has a status of `failed` with an indication of the device that it was detached from. It is also possible for the device (such as `c0t0d0s2` in the example) not to be listed at all should the disk fail completely.

In some cases, the `vxdisk list` output can differ. For example, if the boot disk has uncorrectable failures associated with the UNIX partition table, a missing root partition cannot be corrected but there are no errors in the VERITAS Volume Manager private area. The `vxdisk list` command displays a listing such as this:

DEVICE	TYPE	DISK	GROUP	STATUS
<code>c0t0d0s2</code>	sliced	disk01	rootdg	online
<code>c0t1d0s2</code>	sliced	disk02	rootdg	online

However, because the error was not correctable, the disk is viewed as failed. In such a case, remove the association between the failing device and its disk name using the `vxdiskadm` “Remove a disk for replacement” menu item. (See the `vxdiskadm` (1M) manual page for more information.) You can then perform any special procedures to correct the problem, such as reformatting the device.

To re-add the disk, select the `vxdiskadm` “Replace a failed or removed disk” menu item to replace the disk, and specify the *same* device as the replacement. For the example above, you would replace `disk01` with the device `c0t0d0s2`.

If hot-relocation is enabled when a mirrored boot disk fails, an attempt is made to create a new mirror and remove the failed subdisks from the failing boot disk. If a re-add succeeds after a successful hot-relocation, the `root` and other volumes affected by the disk failure no longer exist on the re-added disk. Run `vxunreloc` to move the hot-relocated subdisks back to the newly replaced disk.

Replacing a Failed Boot Disk

The replacement disk must have at least as much storage capacity as was in use on the disk being replaced. It must be large enough to accommodate all subdisks of the original disk at their current disk offsets.

To estimate the size of the replacement disk, use this command:

```
# vxprint -st -e 'sd_disk="diskname"'
```

where *diskname* is the name of the disk that failed or of one of its mirrors. From the resulting output, add the `DISKOFFS` and `LENGTH` values for the last subdisk listed for the disk. This size is in 512-byte sectors. Divide this number by 2 for the size in kilobytes.

Note Disk sizes reported by manufacturers do not usually represent usable capacity. Also, some disk manufacturers use the term “megabyte” to mean a million bytes rather than the usual meaning of 1,048,576 bytes.

To replace a boot disk:

1. Boot the system from an alternate boot disk (see “[Booting from Alternate Boot Disks](#)” on page 20).
2. Remove the association between the failing device and its disk name using the “Remove a disk for replacement” function of `vxdiskadm`. (See the `vxdiskadm` (1M) manual page for more information.)
3. Shut down the system and replace the failed hardware.



4. After rebooting, use the `vxdiskadm` “Replace a failed or removed disk” menu item to notify VxVM that you have replaced the failed disk.

Recovery by Reinstallation

Reinstallation is necessary if all copies of your boot (`root`) disk are damaged, or if certain critical files are lost due to file system damage. If either of these failure occurs and you do not have a backup of the `root` file system, you must reinstall the entire system.

If these types of failures occur, attempt to preserve as much of the original VxVM configuration as possible. Any volumes that are not directly involved in the failure do not need to be reconfigured. You do not have to reconfigure any volumes that are preserved.

General Reinstallation Information

This section describes procedures used to reinstall VxVM and preserve as much of the original configuration as possible after a failure.

Note System reinstallation destroys the contents of any disks that are used for reinstallation.

All VxVM-related information is removed during reinstallation. Data removed includes data in private areas on removed disks that contain the disk identifier and copies of the VxVM configuration. The removal of this information makes the disk unusable as a VM disk.

The system `root` disk is always involved in reinstallation. Other disks can also be involved. If the root disk was placed under VxVM control, either during VERITAS Volume Manager installation or by later encapsulation, that disk and any volumes or mirrors on it are lost during reinstallation. Any other disks that are involved in the reinstallation, or that are removed and replaced, can lose VxVM configuration data (including volumes and mirrors).

If a disk, including the root disk, is not under VxVM control prior to the failure, no VxVM configuration data is lost at reinstallation. For information on replacing disks, see “Removing and Replacing Disks” in the “Administering Disks” chapter of the *VERITAS Volume Manager Administrator’s Guide*.

Although it simplifies the recovery process after reinstallation, not having the root disk under VERITAS Volume Manager control increases the possibility of a reinstallation being necessary. By having the root disk under VxVM control and creating mirrors of the root disk contents, you can eliminate many of the problems that require system reinstallation.

When reinstallation is necessary, the only volumes saved are those that reside on, or have copies on, disks that are not directly involved with the failure and reinstallation. Any volumes on the root disk and other disks involved with the failure or reinstallation are lost during reinstallation. If backup copies of these volumes are available, the volumes can be restored after reinstallation.

Reinstalling the System and Recovering VxVM

To reinstall the system and recover the VERITAS Volume Manager configuration, use the following procedure. These steps are described in detail in the sections that follow:

1. [“Prepare the System for Reinstallation”](#) on page 37.

Replace any failed disks or other hardware, and detach any disks not involved in the reinstallation.

2. [“Reinstall the Operating System”](#) on page 38.

Reinstall the base system and any other unrelated Volume Manager packages.

3. [“Reinstall VxVM”](#) on page 38.

Add the Volume Manager package, but do *not* execute the `vxinstall` command.

4. [“Recover the VERITAS Volume Manager Configuration”](#) on page 38.

5. [“Clean up the System Configuration”](#) on page 40.

Restore any information in volumes affected by the failure or reinstallation, and recreate system volumes (`rootvol`, `swapvol`, `usr`, and other system volumes).

6. [“Start up Hot-Relocation”](#) on page 45.

Prepare the System for Reinstallation

To prevent the loss of data on disks not involved in the reinstallation, involve only the root disk and any other disks that contain portions of the operating system in the reinstallation procedure. For example, if the `/usr` file system is configured on a separate disk, leave that disk connected. Several of the automatic options for installation access disks other than the `root` disk without requiring confirmation from the administrator.

Disconnect all other disks containing volumes (or other data that should be preserved) prior to reinstalling the operating system. For example, if you originally installed the operating system with the `home` file system on a separate disk, disconnect that disk to ensure that the `home` file system remains intact.



Reinstall the Operating System

Once any failed or failing disks have been replaced and disks not involved with the reinstallation have been detached, reinstall the operating system as described in your operating system documentation. Install the operating system prior to installing VxVM.

Ensure that no disks other than the root disk are accessed in any way while the operating system installation is in progress. If anything is written on a disk other than the root disk, the VERITAS Volume Manager configuration on that disk may be destroyed.

Note During reinstallation, you can change the system's host name (or host ID). It is recommended that you keep the existing host name, as this is assumed by the procedures in the following sections.

Reinstall VxVM

To reinstall VERITAS Volume Manager, follow these steps:

1. Load VERITAS Volume Manager from CD-ROM. Follow the instructions in the *VERITAS Volume Manager Installation Guide*.

Caution To reconstruct the Volume Manager configuration that remains on the non-root disks, do *not* use `vxinstall` to initialize VxVM after loading the software from CD-ROM.

2. Use the `vxlicense` command to install the VERITAS Volume Manager license key (see the `vxlicense(1M)` manual page for more information):

```
# vxlicense -c
```

Recover the VERITAS Volume Manager Configuration

Once the VERITAS Volume Manager packages have been loaded, and you have installed the license for VxVM, recover the VERITAS Volume Manager configuration using the following procedure:

1. Touch `/etc/vx/reconfig.d/state.d/install-db`.
2. Shut down the system.
3. Reattach the disks that were removed from the system.
4. Reboot the system.

5. When the system comes up, bring the system to single-user mode using the following command:

```
# exec init s
```

6. When prompted, enter the password and press Return to continue.
7. Remove files involved with installation that were created when you loaded VxVM but are no longer needed using the following command:

```
# rm -rf /etc/vx/reconfig.d/state.d/install-db
```

8. Start some VERITAS Volume Manager I/O daemons using the following command:

```
# vxiod set 10
```

9. Start the VERITAS Volume Manager configuration daemon, `vxconfigd`, in disabled mode using the following command:

```
# vxconfigd -m disable
```

10. Initialize the `vxconfigd` daemon using the following command:

```
# vxdctl init
```

11. Initialize the DMP subsystem using the following command:

```
# vxdctl initdmp
```

12. Enable `vxconfigd` using the following command:

```
# vxdctl enable
```

The configuration preserved on the disks not involved with the reinstallation has now been recovered. However, because the root disk has been reinstalled, it does not appear to VxVM as a VM disk. The configuration of the preserved disks does not include the root disk as part of the VxVM configuration.

If the root disk of your system and any other disks involved in the reinstallation were not under VxVM control at the time of failure and reinstallation, then the reconfiguration is complete at this point. For information on replacing disks, see “Removing and Replacing Disks” in the “Administering Disks” chapter of the *VERITAS Volume Manager Administrator’s Guide*. There are several methods available to replace a disk; choose the method that you prefer.



If the root disk (or another disk) was involved with the reinstallation, any volumes or mirrors on that disk (or other disks no longer attached to the system) are now inaccessible. If a volume had only one plex contained on a disk that was reinstalled, removed, or replaced, then the data in that volume is lost and must be restored from backup. In addition, the system `root` file system, `swap` area, (and on some systems `stand` area), and `/usr` file system are *no longer* located on volumes. To correct these problems, follow the instructions in “[Clean up the System Configuration](#).”

Clean up the System Configuration

To clean up the configuration of your system after reinstallation of VxVM, you must address the following issues:

- ◆ [Clean up Rootability](#)
- ◆ [Clean up Volumes](#)
- ◆ [Clean up Disk Configuration](#)
- ◆ [Reconfigure Rootability](#)
- ◆ [Final Volume Reconfiguration](#)

Clean up Rootability

To begin the cleanup of the VERITAS Volume Manager configuration, remove any volumes associated with rootability. This must be done if the root disk (and any other disk involved in the system boot process) was under VERITAS Volume Manager control. The volumes to remove are:

- ◆ `rootvol`, that contains the `root` file system
- ◆ `swapvol`, that contains the `swap` area
- ◆ (on some systems) `standvol`, that contains the `stand` file system
- ◆ `usr`, that contains the `/usr` file system

To remove the root volume, use the `vxedit` command:

```
# vxedit -fr rm rootvol
```

Repeat this command, using `swapvol` and `usr` (`standvol`) in place of `rootvol`, to remove the `swap`, `stand`, and `usr` volumes.

Clean up Volumes

After completing the rootability cleanup, you must determine which volumes need to be restored from backup. The volumes to be restored include those with all mirrors (all copies of the volume) residing on disks that have been reinstalled or removed. These

volumes are invalid and must be removed, recreated, and restored from backup. If only some mirrors of a volume exist on reinitialized or removed disks, these mirrors must be removed. The mirrors can be re-added later.

To restore the volumes, perform these steps:

1. Establish which VM disks have been removed or reinstalled using the following command:

```
# vxdisk list
```

This displays a list of system disk devices and the status of these devices. For example, for a reinstalled system with three disks and a reinstalled root disk, the output of the `vxdisk list` command is similar to this:

DEVICE	TYPE	DISK	GROUP	STATUS
c0t0d0s2	sliced	-	-	error
c0t1d0s2	sliced	disk02	rootdg	online
c0t2d0s2	sliced	disk03	rootdg	online
-	-	disk01	rootdg	failed was:c0t0d0s2

The display shows that the reinstalled root device, `c0t0d0s2`, is not associated with a VM disk and is marked with a status of `error`. The disks `disk02` and `disk03` were not involved in the reinstallation and are recognized by VxVM and associated with their devices (`c0t1d0s2` and `c0t2d0s2`). The former `disk01`, which was the VM disk associated with the replaced disk device, is no longer associated with the device (`c0t0d0s2`).

If other disks (with volumes or mirrors on them) had been removed or replaced during reinstallation, those disks would also have a disk device listed in `error` state and a VM disk listed as not associated with a device.

2. Once you know which disks have been removed or replaced, locate all the mirrors on failed disks using the following command:

```
# vxprint -sF "%vname" -e'sd_disk = "disk"'
```

where `disk` is the name of a disk with a `failed` status. Be sure to enclose the disk name in quotes in the command. Otherwise, the command returns an error message. The `vxprint` command returns a list of volumes that have mirrors on the failed disk. Repeat this command for every disk with a `failed` status.

3. Check the status of each volume and print volume information using the following command:

```
# vxprint -th volume
```



where *volume* is the name of the volume to be examined. The `vxprint` command displays the status of the volume, its plexes, and the portions of disks that make up those plexes. For example, a volume named `v01` with only one plex resides on the reinstalled disk named `disk01`. The `vxprint -th v01` command produces the following output:

V	NAME	USETYPE	KSTATE	STATE	LENGTH	READPOL	PREFPLEX
PL	NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WIDMODE
SD	NAME	PLEX	DISK	DISKOFFSLENGTH	COL/]OFF	DEVICE	MODE
v	v01	fsgen	DISABLED	ACTIVE	24000	SELECT	-
pl	v01-01	v01	DISABLED	NODEVICE	24000	CONCAT	- RW
sd	disk01-06	v0101	disk01	245759	24000	0	c1t5d1 ENA

The only plex of the volume is shown in the line beginning with `pl`. The `STATE` field for the plex named `v01-01` is `NODEVICE`. The plex has space on a disk that has been replaced, removed, or reinstalled. The plex is no longer valid and must be removed.

4. Because `v01-01` was the only plex of the volume, the volume contents are irrecoverable except by restoring the volume from a backup. The volume must also be removed. If a backup copy of the volume exists, you can restore the volume later. Keep a record of the volume name and its length, as you will need it for the backup procedure.

Remove irrecoverable volumes (such as `v01`) using the following command:

```
# vxedit -r rm v01
```

5. It is possible that only part of a plex is located on the failed disk. If the volume has a striped plex associated with it, the volume is divided between several disks. For example, the volume named `v02` has one striped plex striped across three disks, one of which is the reinstalled disk `disk01`. The `vxprint -th v02` command produces the following output:

V	NAME	USETYPE	KSTATE	STATE	LENGTH	READPOL	PREFPLEX
PL	NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WIDMODE
SD	NAME	PLEX	DISK	DISKOFFSLENGTH	COL/]OFF	DEVICE	MODE
v	v02	fsgen	DISABLED	ACTIVE	30720	SELECT	v02-01
pl	v02-01	v02	DISABLED	NODEVICE	30720	STRIPE	3/128 RW
sd	disk02-02	v02-01	disk01	424144	10240	0/0	c1t5d2 ENA
sd	disk01-05	v02-01	disk01	620544	10240	1/0	c1t5d3 DIS
sd	disk03-01	v02-01	disk03	620544	10240	2/0	c1t5d4 ENA

The display shows three disks, across which the plex `v02-01` is striped (the lines starting with `sd` represent the stripes). One of the stripe areas is located on a failed disk. This disk is no longer valid, so the plex named `v02-01` has a state of `NODEVICE`.

Since this is the only plex of the volume, the volume is invalid and must be removed. If a copy of v02 exists on the backup media, it can be restored later. Keep a record of the volume name and length of any volume you intend to restore from backup.

Remove invalid volumes (such as v02) using the following command:

```
# vxedit -r rm v02
```

6. A volume that has one mirror on a failed disk can also have other mirrors on disks that are still valid. In this case, the volume does not need to be restored from backup, since the data is still valid on the valid disks.

The output of the `vxprint -th` command for a volume with one plex on a failed disk (disk01) and another plex on a valid disk (disk02) is similar to the following:

V	NAME	USETYPE	KSTATE	STATE	LENGTH	READPOL	PREFPLEX
PL	NAME	VOLUME	KSTATE	STATE	LENGTH	LAYOUT	NCOL/WIDMODE
SD	NAME	PLEX	DISK	DISKOFFS	LENGTH	COL/]OFF	DEVICE MODE
v	v03	fsgen	DISABLED	ACTIVE	0720	SELECT	-
pl	v03-01	v03	DISABLED	ACTIVE	30720	CONCAT	- RW
sd	disk02-01	v03-01	disk01	620544	30720	0	c1t5d5 ENA
pl	v03-02	v03	DISABLED	NODEVICE	30720	CONCAT	- RW
sd	disk01-04	v03-02	disk03	262144	30720	0	c1t5d6 DIS

This volume has two plexes, v03-01 and v03-02. The first plex (v03-01) does not use any space on the invalid disk, so it can still be used. The second plex (v03-02) uses space on invalid disk disk01 and has a state of NODEVICE. Plex v03-02 must be removed. However, the volume still has one valid plex containing valid data. If the volume needs to be mirrored, another plex can be added later. Note the name of the volume to create another plex later.

To remove an invalid plex, use the `vxplex` command to dissociate and then remove the plex from the volume. For example, to dissociate and remove the plex v03-02, use the following command:

```
# vxplex -o rm dis v03-02
```

7. Once all the volumes have been cleaned up, clean up the disk configuration as described in the following section, [“Clean up Disk Configuration.”](#)



Clean up Disk Configuration

Once all invalid volumes and plexes have been removed, the disk configuration can be cleaned up. Each disk that was removed, reinstalled, or replaced (as determined from the output of the `vxdisk list` command) must be removed from the configuration.

To remove the disk, use the `vxdbg` command. To remove the failed disk `disk01`, use the following command:

```
# vxdbg rmdisk disk01
```

If the `vxdbg` command returns an error message, some invalid mirrors exist. Repeat the processes described in [“Clean up Volumes”](#) on page 40 until all invalid volumes and mirrors are removed.

Reconfigure Rootability

Once all the invalid disks have been removed, the replacement or reinstalled disks can be added to VERITAS Volume Manager control. If the root disk was originally under VERITAS Volume Manager control or you now wish to put the root disk under VERITAS Volume Manager control, add this disk first.

To add the root disk to VERITAS Volume Manager control, use the `vxdiskadm` command:

```
# vxdiskadm
```

From the `vxdiskadm` main menu, select menu item 2 (Encapsulate a disk). Follow the instructions and encapsulate the root disk for the system.

When the encapsulation is complete, reboot the system to multi-user mode.

Final Volume Reconfiguration

Once the root disk is encapsulated, any other disks that were replaced should be added using the `vxdiskadm` command. If the disks were reinstalled during the operating system reinstallation, they should be encapsulated; otherwise, they can be added.

Once all the disks have been added to the system, any volumes that were completely removed as part of the configuration cleanup can be recreated and their contents restored from backup. The volume recreation can be done by using the `vxassist` command or the graphical user interface.

For example, to recreate the volumes `v01` and `v02`, use the following command:

```
# vxassist make v01 24000
# vxassist make v02 30720 layout=stripe nstripe=3
```

Once the volumes are created, they can be restored from backup using normal backup/restore procedures.

Recreate any plexes for volumes that had plexes removed as part of the volume cleanup. To replace the plex removed from volume `v03`, use the following command:

```
# vxassist mirror v03
```

Once you have restored the volumes and plexes lost during reinstallation, recovery is complete and your system is configured as it was prior to the failure.

The final step is to start up hot-relocation, if this is required.

Start up Hot-Relocation

To start up the hot-relocation service, either reboot the system or manually start the relocation watch daemon, `vxrelocd` (this also starts the `vxnotify` process).

Note Hot-relocation should only be started when you are sure that it will not interfere with other reconfiguration procedures.

See “Modifying the Behavior of Hot-Relocation” in the “Administering Hot-Relocation” chapter of the *VERITAS Volume Manager Administrator’s Guide* for more information about running `vxrelocd` and about modifying its behavior.

To determine if hot-relocation has been started, use the following command to search for its entry in the process table:

```
# ps -ef | grep vxrelocd | grep -v grep
```





Introduction

This chapter provides information on error messages associated with the VERITAS Volume Manager (VxVM) configuration daemon (`vxconfigd`), the kernel, and other utilities. It covers most informational, failure, and error messages displayed on the console by `vxconfigd`, and by the VERITAS Volume Manager kernel driver, `vxio`. These include some errors that are infrequently encountered and difficult to troubleshoot.

Note Some error messages described here may not apply to your system.

Clarifications are included to elaborate on the situation or problem that generated a particular message. Wherever possible, a recovery procedure (*Action*) is provided to help you to locate and correct the problem.

Logging Error Messages

VxVM provides the option of logging console output to a file. This logging is useful in that any messages output just before a system crash will be available in the log file (presuming that the crash does not result in file system corruption). `vxconfigd` controls whether such logging is turned on or off. If enabled, the default log file is `/var/vxvm/vxconfigd.log`.

`vxconfigd` also supports the use of `syslog` to log all of its regular console messages. When this is enabled, all console output is directed through the `syslog` interface.

`syslog` and log file logging can be used together to provide reliable logging to a private log file, along with distributed logging through `syslogd`.

Both `syslog` and log file logging are disabled by default.

To enable logging of console output to a file, edit the VERITAS Volume Manager startup script as described in “[Configuring Logging in the Startup Script](#)” on page 48, or invoke `vxconfigd` as shown here:

```
# vxconfigd [-x [1-9]] -x log
```



There are 9 possible levels of debug logging. Specifying `-x 9 -x log` writes all debug and error messages to the file `/var/vxvm/vxconfigd.log`.

To enable `syslog` logging of console output, specify the option `-x syslog` to `vxconfigd`:

```
# vxconfigd -x syslog
```

If only `-x syslog` is specified, *Debug* messages are not logged. Only *Error*, *Fatal Error*, *Warning*, and *Notice* messages are logged.

If you specify `-x 9 -x log -x syslog`, *Debug* messages are only written to `/var/vxvm/vxconfigd.log`, but error messages are also written to `/var/adm/syslog/syslog.log`.

Configuring Logging in the Startup Script

To enable log file or `syslog` logging, you can edit the following portion of the `/etc/init.d/vxvm-sysboot` script that starts VxVM:

```
# comment-out or uncomment any of the following lines to enable or
# disable the corresponding feature in vxconfigd.

#opts="$opts -x syslog"           # use syslog for console messages
#opts="$opts -x log"             # messages to vxconfigd.log
#opts="$opts -x logfile=/foo/bar" # specify an alternate log file
#opts="$opts -x timestamp"       # timestamp console messages

# to turn on debugging console output, uncomment the following
# line.
# The debug level can be set higher for more output.  The highest
# debug level is 9.

#debug=1                          # enable debugging console output
```

Uncomment the lines corresponding to the features that you want enabled at startup. For example, to set up `vxconfigd` to automatically use file logging, uncomment the `opts="$opts -x log"` string.

For more information on logging options available through `vxconfigd`, refer to the `vxconfigd(1M)` manual page.

Understanding Error Messages

VxVM is fault-tolerant and resolves most problems without system administrator intervention. If the configuration daemon (`vxconfigd`) recognizes the actions that are necessary, it queues up the transactions that are required. VxVM provides atomic changes of system configurations; either a transaction completes fully, or the system is left in the same state as though the transaction was never attempted. If `vxconfigd` is unable to recognize and fix system problems, the system administrator needs to handle the task of problem solving using the diagnostic messages that are returned from the software. The following sections list error messages that may be seen along with a more detailed description of the likely cause of the problem and suggestions for any actions that can be taken.

Kernel Panic Messages

A panic is a severe event as it halts a system during its normal operation. A panic message from the kernel indicates the nature of the hardware problem or software inconsistency that is so severe that the system cannot continue. The operating system may also provide a dump of the CPU register contents and a stack trace to aid in identifying the cause of the panic.

```
vxvm:vxio:PANIC: Object association depth overflow
```

- ◆ *Description:* This internal error should not occur unless there is a bug in VxVM.
- ◆ *Action:* Contact Customer Support.

Kernel Warning Messages

A warning message from the kernel indicates that a non-critical operation has failed, possibly because some resource is not available. Corrective action should be taken immediately.

```
vxvm:vxio:WARNING: Cannot find device number for boot_path
```

- ◆ *Description:* The boot path retrieved from the system PROMs cannot be converted to a valid device number.
- ◆ *Action:* Check your PROM settings for the correct boot string.



```
vxvm:vxio:WARNING: check_ilocks: overlapping ilocks: offset for length,  
offset for length  
vxvm:vxio:WARNING: check_ilocks: stranded ilock on object_name start  
offset len length
```

- ◆ **Description:** These internal errors do not occur unless there is a bug in VxVM.
- ◆ **Action:** Contact Customer Support.

```
vxvm:vxio:WARNING: detaching RAID-5 volume
```

- ◆ **Description:** Either a double-failure condition in the RAID-5 volume has been detected in the kernel or some other fatal error is preventing further use of the array.
- ◆ **Action:** If two or more disks have been lost due to a controller or power failure, use the `vxrecover` utility to recover them once they have been re-attached to the system. Check for other console error messages that may provide additional information about the failure.

```
vxvm:vxio:WARNING: Device major, minor: Received spurious close
```

- ◆ **Description:** A close was received for an object that was not open. This can only happen if the operating system is not correctly tracking opens and closes.
- ◆ **Action:** No action is necessary; the system will continue.

```
vxvm:vxio:WARNING: Double failure condition detected on RAID-5 volume
```

- ◆ **Description:** I/O errors have been received in more than one column of a RAID-5 volume. This could be caused by:
 - a controller failure making more than a single drive unavailable
 - the loss of a second drive while running in degraded mode
 - two separate disk drives failing simultaneously (unlikely)
- ◆ **Action:** Correct the hardware failures if possible. Then recover the volume using the `vxrecover` command.

```
vxvm:vxio:WARNING: DRL volume volume is detached
```

- ◆ **Description:** A Dirty Region Logging volume became detached because a DRL log entry could not be written. If this is due to a media failure, other errors may have been logged to the console.
- ◆ **Action:** The volume containing the DRL log continues in operation. If the system fails before the DRL has been repaired, a full recovery of the volume's contents may be necessary and will be performed automatically when the system is restarted. To recover from this error, use the `vxassist addlog` command add a new DRL log to the volume.

`vxvm:vxio:WARNING: Failed to log the detach of the DRL volume volume`

- ◆ *Description:* An attempt failed to write a kernel log entry indicating the loss of a DRL volume. The attempted write to the log failed either because the kernel log is full, or because of a write error to the drive. The volume becomes detached.
- ◆ *Action:* Messages about log failures are usually fatal, unless the problem is transient. However, the kernel log is sufficiently redundant that such errors are unlikely to occur.

If the problem is not transient (that is, the drive cannot be fixed and brought back online without data loss), recreate the disk group from scratch and restore all of its volumes from backups. Even if the problem is transient, reboot the system after correcting the problem.

If error messages are seen from the disk driver, it is likely that the last copy of the log failed due to a disk error. Replace the failed drive in the disk group. The log re-initializes on the new drive. Finally force the failed volume into an active state and recover the data.

`vxvm:vxio:WARNING: Failure in RAID-5 logging operation`
`vxvm:vxio:WARNING: log object object_name detached from RAID-5 volume`

- ◆ *Description:* Together, these errors indicate that a RAID-5 log has failed.
- ◆ *Action:* To restore RAID-5 logging to a RAID-5 volume, create a new log plex and attach it to the volume.

`vxvm:vxio:WARNING: Illegal vminor encountered`

- ◆ *Description:* An attempt was made to open volume device other than the root volume device before the `vxconfigd` loaded the volume configuration.
- ◆ *Action:* None; under normal startup conditions, this message should not occur. If necessary, start VxVM and re-attempt the operation.

`vxvm:vxio:WARNING: Kernel log full: volume detached`

- ◆ *Description:* A plex detach failed because the kernel log was full. As a result, the mirrored volume will become detached.
- ◆ *Action:* It is unlikely that this condition ever occurs. The only corrective action is to reboot the system.

`vxvm:vxio:WARNING: Kernel log update failed: volume detached`

- ◆ *Description:* Detaching a plex failed because the kernel log could not be flushed to disk. As a result, the mirrored volume became detached. This may be caused by all the disks containing a kernel log going bad.
- ◆ *Action:* Repair or replace the failed disks so that kernel logging can once again function.



`vxvm:vxio:WARNING: mod_install returned errno`

- ◆ **Description:** A call made to the operating system `mod_install` function to load the `vxio` driver failed.
- ◆ **Action:** Check for additional console messages that may explain why the load failed. Also check the console messages log file for any additional messages that were logged but not displayed on the console.

`vxvm:vxio:WARNING: object plex detached from volume volume`

- ◆ **Description:** An uncorrectable error was detected by the mirroring code and a mirror copy was detached.
- ◆ **Action:** To restore redundancy, it may be necessary to add another mirror. The disk on which the failure occurred should be reformatted or replaced.

`vxvm:vxio:WARNING: object subdisk detached from RAID-5 volume at column column offset offset`

- ◆ **Description:** A subdisk was detached from a RAID-5 volume because of the failure of a disk or an uncorrectable error occurring on that disk.
- ◆ **Action:** Check for other console error messages indicating the cause of the failure. Replace a failed disk as soon as possible.

`vxvm:vxio:WARNING: object_type object_name block offset:Uncorrectable read error ...`

`vxvm:vxio:WARNING: object_type object_name block offset:Uncorrectable write error ...`

- ◆ **Description:** A read or write operation from or to the specified VERITAS Volume Manager object failed. An error is returned to the application.
- ◆ **Action:** These errors may represent lost data. Data may need to be restored and failed media may need to be repaired or replaced. Depending on the type of object failing and on the type of recovery suggested for the object type, an appropriate recovery operation may be necessary.

`vxvm:vxio:WARNING: Overlapping mirror plex detached from volume volume`

- ◆ **Description:** An error has occurred on the last complete plex in a mirrored volume. Any sparse mirrors that map the failing region are detached so that they cannot be accessed to satisfy that failed region inconsistently.
- ◆ **Action:** The message indicates that some data in the failing region may no longer be stored redundantly.

`vxvm:vxio:WARNING: RAID-5 volume entering degraded mode operation`

- ◆ *Description:* An uncorrectable error has forced a subdisk to detach. At this point, not all data disks exist to provide the data upon request. Instead, parity regions are used to regenerate the data for each stripe in the array. Consequently, access will consequently take longer and will involve reading from all drives in the stripe.
- ◆ *Action:* Check for other console error messages that indicate the cause of the failure. Replace any failed disks as soon as possible.

`vxvm:vxio:WARNING: read error on mirror plex of volume volume offset offset length length`

- ◆ *Description:* An error was detected while reading from a mirror. This error may lead to further action shown by later error messages.
- ◆ *Action:* If the volume is mirrored, no further action is necessary since the alternate mirror's contents will be written to the failing mirror; this is often sufficient to correct media failures. If this error occurs often but never leads to a plex detach, there may be a marginally defective region on the disk at the position indicated. It may eventually be necessary to remove data from this disk (see the `vxevac(1M)` manual page) and then to reformat the drive.

If the volume is not mirrored, this message indicates that some data could not be read. The file system or other application reading the data may report an additional error, but in either event, data has been lost. The volume can be partially salvaged and moved to another location if desired.

`vxvm:vxio:WARNING: Root volumes are not supported on your PROM version.`

- ◆ *Description:* If your system's PROMs are not a recent OpenBoot PROM type, root volumes are unusable.
- ◆ *Action:* If you have set up a root volume, undo the configuration by running `vxunroot` or removing the `rootdev` line from `/etc/system` as soon as possible. Contact your hardware vendor for an upgrade to your PROM level.

`vxvm:vxio:WARNING: subdisk subdisk failed in plex plex in volume volume`

- ◆ *Description:* The kernel has detected a subdisk failure, which may mean that the underlying disk is failing.
- ◆ *Action:* Check for obvious problems with the disk (such as a disconnected cable). If hot-relocation is enabled and the disk is failing, recovery from subdisk failure is handled automatically.



`vxvm:vxio:WARNING: write error on mirror plex of volume volume offset offset length length`

- ◆ **Description:** An error was detected while writing to a mirror. This error will generally be followed by a detach message, unless the volume is not mirrored.
- ◆ **Action:** The disk reporting the error is failing to correctly store written data. If the volume is not mirrored, consider removing the data and reformatting the disk. If the volume is mirrored, it will become detached and you should replace or reformat the disk.

If this error occurs often but never leads to a plex detach, there may be a marginally defective region on the disk at the position shown. It may eventually be necessary to remove data from this disk (see the `vxevac(1M)` manual page) and then to reformat the drive.

Kernel Notice Messages

A notice message indicates that an error has occurred that should be monitored. Shutting down the system is unnecessary, although you may need to take action to remedy the fault at a later date.

`vxvm:vxio:NOTICE: Can't close disk disk in group disk_group. If it is removable media (like a floppy), it may have been removed. Otherwise, there may be problems with the drive. Kernel error code public_region_error/private_region_error`

- ◆ **Description:** This is unlikely to happen; closes should not fail.
- ◆ **Action:** None.

`vxvm:vxio:NOTICE: Can't open disk disk in group disk_group. If it is removable media (like a floppy), it may not be mounted or ready. Otherwise, there may be problems with the drive. Kernel error code number`

- ◆ **Description:** The named disk cannot be accessed in the named disk group.
- ◆ **Action:** Ensure that the disk exists, is connected and powered on, and is visible to the system.

`vxvm:vxio:NOTICE: read error on object subdisk of mirror plex in volume volume (start offset, length length) corrected.`

- ◆ **Description:** A read error occurred, which caused a read of an alternate mirror and a writeback to the failing region. This writeback was successful and the data was corrected on disk.

- ◆ *Action:* None; the problem was corrected automatically. Note the location of the failure for future reference. If the same region of the subdisk fails again, this may indicate a more insidious failure and the disk should be reformatted at the next reasonable opportunity.

vxvm:vxio:NOTICE: *string* on volume *device_#* (*device_name*) in disk group *group_name*

- ◆ *Description:* An application running on top of VxVM has requested the output of the message *string*.
- ◆ *Action:* Refer to the application documentation for more information.

vxassist Error Messages

An error message from the `vxassist` command indicates that the requested operation cannot be performed. Follow the recommended course of action given below.

vxvm:vxassist: ERROR: Volume record id *rid* is not found in the configuration.

- ◆ *Description:* An error was detected while reattaching a snapshot volume using `snapback`. This happens if a volume's record identifier (*rid*) changes as a result of a disk group split that moved the original volume to a new disk group. The snapshot volume is unable to recognize the original volume because its record identifier has changed.
- ◆ *Action:* Use the following command to perform the `snapback`:

```
# vxplex [-g diskgroup] -f snapback volume plex
```

vxassist Warning Messages

A warning message from the `vxassist` command indicates a problem with its operation. Action should be taken to correct the problem as soon as possible. Follow the recommended course of action given below.

vxvm:vxassist: WARNING: volume *volume* already has at least one snapshot plex

Snapshot volume created with these plexes will have a dco log volume with no associated dco log plex.

- ◆ *Description:* An error was detected while adding a DCO object and DCO log volume to a mirrored volume. There is at least one snapshot plex already created on the volume. Because this snapshot plex was created when no DCO was associated with the volume, there is no DCO log plex allocated for it.



- ◆ *Action:* See the section “Enabling Persistent FastResync on Existing Volumes with Associated Snapshots” in the chapter “Administering Volumes” of the *VERITAS Volume Manager Administrator’s Guide*.

vxconfigd Fatal Error Messages

A fatal error message from the configuration daemon, vxconfigd, indicates a severe problem with the operation of VxVM that prevents it from running.

```
vxvm:vxconfigd: FATAL ERROR: Disk group rootdg: Inconsistency -- Not loaded into kernel
```

```
vxvm:vxconfigd: FATAL ERROR: Group group: Cannot update kernel
```

```
vxvm:vxconfigd: FATAL ERROR: Interprocess communication failure: reason
```

```
vxvm:vxconfigd: FATAL ERROR: Invalid status stored in kernel
```

- ◆ *Description:* These internal errors should not occur unless there is a bug in VxVM.
- ◆ *Action:* Contact Customer Support.

```
vxvm:vxconfigd: FATAL ERROR: Memory allocation failure during startup
```

- ◆ *Description:* This implies that there is insufficient memory to start up VxVM and to get the volumes for the root and /usr file systems running.
- ◆ *Action:* This error should not normally occur, unless your system has very small amounts of memory. Adding swap space probably will not help, because this error is most likely to occur early in the boot sequence, before swap areas have been added.

```
vxvm:vxconfigd: FATAL ERROR: Rootdg cannot be imported during boot
```

- ◆ *Description:* This internal error should not occur unless there is a bug in VxVM.
- ◆ *Action:* Contact Customer Support.

```
vxvm:vxconfigd: FATAL ERROR: Unexpected threads failure: reason
```

- ◆ *Description:* This unexpected operating system error should not occur unless there is a bug in VxVM or in the operating system multithreading libraries.
- ◆ *Action:* Contact Customer Support.

vxconfigd Error Messages

An error message from the configuration daemon, `vxconfigd`, indicates a problem with the operation of VxVM that may prevent it from running effectively. Action should be taken to correct the problem immediately.

```
vxvm:vxconfigd: ERROR: Cannot get all disks from the kernel: reason
vxvm:vxconfigd: ERROR: Cannot get all disk groups from the kernel:
reason
vxvm:vxconfigd: ERROR: Cannot get kernel transaction state: reason
vxvm:vxconfigd: ERROR: Cannot get private storage from kernel: reason
vxvm:vxconfigd: ERROR: Cannot get private storage size from kernel:
reason
vxvm:vxconfigd: ERROR: Cannot get record record_name from kernel: reason
```

- ◆ **Description:** These internal errors should not occur unless there is a bug in VxVM.
- ◆ **Action:** Contact Customer Support.

```
vxvm:vxconfigd: ERROR: Cannot kill existing daemon, pid=process_ID
```

- ◆ **Description:** The `-k` (kill existing `vxconfigd` process) option was specified, but a running configuration daemon process could not be killed. A configuration daemon process, for purposes of this discussion, is any process that opens the `/dev/vx/config` device (only one process can open that device at a time). If there is a configuration daemon process already running, then the `-k` option causes a `SIGKILL` signal to be sent to that process. If, within a certain period of time, there is still a running configuration daemon process, the above error message is displayed.
- ◆ **Action:** This error can result from a kernel error that has made the configuration daemon process unkillable, from some other kind of kernel error, or from some other user starting another configuration daemon process after the `SIGKILL` signal. This last condition can be tested for by running `vxconfigd -k` again. If the error message appears again, contact Customer Support.

```
vxvm:vxconfigd: ERROR: Cannot make directory directory_path: reason
```

- ◆ **Description:** `vxconfigd` failed to create a directory that it expects to be able to create. Directories that `vxconfigd` might try to create are: `/dev/vx/dsk`, `/dev/vx/rdsk`, and `/var/vxvm/tempdb`. Also, for each disk group, `/dev/vx/dsk/diskgroup` and `/dev/vx/rdsk/diskgroup` directories are created. The system error related to the failure is given in *reason*. A system error of "No such file or directory" indicates that one of the prefix directories (for example, `/var/vxvm`) does not exist.



This type of error normally implies that the VERITAS Volume Manager packages were installed incorrectly. Such an error can also occur if alternate file or directory locations are specified on the command line, using the `-x` option. The `_VXVM_ROOT_DIR` environment variable may also relocate to a directory that lacks a `var/vxvm` subdirectory.

- ◆ **Action:** Try to create the directory manually and then issue the command `vxctl enable`. If the error is due to incorrect installation of the VERITAS Volume Manager packages, try to add the packages again.

```
vxvm:vxconfigd: ERROR: cannot open /dev/vx/config: reason
```

- ◆ **Description:** The `/dev/vx/config` device could not be opened. `vxconfigd` uses this device to communicate with the VERITAS Volume Manager kernel drivers. The most likely reason is “Device is already open.” This indicates that some process (most likely `vxconfigd`) already has `/dev/vx/config` open. Less likely reasons are “No such file or directory” or “No such device or address.” For either of these reasons, likely causes are:

- The VERITAS Volume Manager package installation did not complete correctly.
- The device node was removed by the administrator or by an errant shell script.

- ◆ **Action:** If the reason is “Device is already open,” stop or kill the old `vxconfigd` by running the command:

```
# vxctl -k stop
```

For other failure reasons, consider re-adding the base VERITAS Volume Manager package. This will reconfigure the device node and re-install the VERITAS Volume Manager kernel device drivers. See the *VERITAS Volume Manager Installation Guide* for information on how to add the package. If you cannot re-add the package, contact Customer Support for more information.

```
vxvm:vxconfigd: ERROR: Cannot open /etc/vfstab: reason
```

- ◆ **Description:** `vxconfigd` could not open the `/etc/vfstab` file, for the reason given. The `/etc/vfstab` file is used to determine which volume (if any) to use for the `/usr` file system.
- ◆ **Action:** This error implies that your `root` file system is currently unusable. You may be able to repair the `root` file system by mounting it after booting from a network or CD-ROM `root` file system. If the `root` file system is defined on a volume, then see the procedures defined for recovering from a failed `root` file system in [“Recovery from Boot Disk Failure”](#) on page 19.

```
vxvm:vxconfigd: ERROR: Cannot recover operation in progress
Failed to get group group from the kernel: error
```

- ◆ *Description:* This internal error should not occur unless there is a bug in VxVM.
- ◆ *Action:* Contact Customer Support.

```
vxvm:vxconfigd: ERROR: Cannot reset VxVM kernel: reason
```

- ◆ *Description:* The `-r` reset option was specified to vxconfigd, but the VxVM kernel drivers could not be reset. The most common reason is “A virtual disk device is open.” This implies that a VxVM tracing or volume device is open.
- ◆ *Action:* If you want to reset the kernel devices, track down and kill all processes that have a volume or VERITAS Volume Manager tracing device open. Also, if any volumes are mounted as file systems, unmount those file systems.

Any reason other than “A virtual disk device is open” does not normally occur unless there is a bug in the operating system or in VxVM.

```
vxvm:vxconfigd: ERROR: Cannot start volume volume, no valid plexes
vxvm:vxconfigd: ERROR: Cannot start volume volume, no valid complete
plexes
```

- ◆ *Description:* These errors indicate that the volume for either the `root` or `/usr` file system cannot be started because the volume contains no valid plexes. This can happen, for example, if disk failures have caused all plexes to be unusable. It can also happen as a result of actions that caused all plexes to become unusable (for example, forcing the dissociation of subdisks or detaching, dissociation, or offlining of plexes).
- ◆ *Action:* It is possible that this error results from a drive that failed to spin up. If so, rebooting may fix the problem. If that does not fix the problem, then the only recourse is to restore the `root` or `/usr` file system or to reinstall the system. Restoring the `root` or `/usr` file system requires that you have a valid backup. See [“Repairing root or /usr File Systems on Mirrored Volumes”](#) on page 30 for information on how to fix problems with `root` or `/usr` file system volumes.

```
vxvm:vxconfigd: ERROR: Cannot start volume volume, volume state is
invalid
```

- ◆ *Description:* The volume for the `root` or `/usr` file system is in an unexpected state (not ACTIVE, CLEAN, SYNC or NEEDSYNC). This should not happen unless the system administrator circumvents the mechanisms used by VxVM to create these volumes.
- ◆ *Action:* The only recourse is to bring up VxVM on a CD-ROM or NFS-mounted `root` file system and to fix the state of the volume. See [“Repairing root or /usr File Systems on Mirrored Volumes”](#) on page 30 for further information.



vxvm:vxconfigd: ERROR: Cannot store private storage into the kernel:
error

- ◆ *Description:* This internal error should not occur unless there is a bug in VxVM.
- ◆ *Action:* Contact Customer Support.

vxvm:vxconfigd: ERROR: /dev/vx/info: *reason*

- ◆ *Description:* The /dev/vx/info device could not be opened, or did not respond to a VERITAS Volume Manager kernel request. This error most likely indicates one of the following:
 - The VERITAS Volume Manager package installation did not complete correctly.
 - The device node was removed by the administrator or by an errant shell script.
- ◆ *Action:* Consider re-adding the base VERITAS Volume Manager package. This will reconfigure the device node and re-install the VERITAS Volume Manager kernel device drivers. See the *VERITAS Volume Manager Installation Guide* for information on how to add the package.

vxvm:vxconfigd: ERROR: DG move: can't import diskgroup, giving up

- ◆ *Description:* The specified disk group cannot be imported during a disk group move operation. (The disk group ID is obtained from the disk group that could be imported.)
- ◆ *Action:* The disk group may have been moved to another host. One option is to locate it and use the `vxdg recover` command on both the source and target disk groups. Specify the `-o clean` option with one disk group, and the `-o remove` option with the other disk group. See [“Recovering from Incomplete Disk Group Moves”](#) on page 15 for more information.

vxvm:vxconfigd: ERROR: dg_move_recover: can't locate disk(s), giving up

- ◆ *Description:* Disks involved in a disk group move operation cannot be found, and one of the specified disk groups cannot be imported.
- ◆ *Action:* Manual use of the `vxdg recover` command may be required to clean the disk group to be imported. See [“Recovering from Incomplete Disk Group Moves”](#) on page 15 for more information.

vxvm:vxconfigd: ERROR: Differing version of vxconfigd installed

- ◆ *Description:* A `vxconfigd` daemon was started after stopping an earlier `vxconfigd` with a non-matching version number. This can happen, for example, if you upgrade VxVM and then run `vxconfigd` without first rebooting.
- ◆ *Action:* To fix, reboot the system.

vxvm:vxconfigd: ERROR: Disk *disk*, group *group*, device *device*: not updated with new host ID
Error: *reason*

- ◆ *Description:* This can result from using `vxctl hostid` to change the VERITAS Volume Manager host ID for the system. The error indicates that one of the disks in a disk group could not be updated with the new host ID. Most likely, this indicates that the given disk has become inaccessible or has failed in some other way.
- ◆ *Action:* Try running the following command to determine whether the disk is still operational:

```
# vxdisk check device
```

If the disk is no longer operational, `vxdisk` should print a message such as:

```
device: Error: Disk write failure
```

This will result in the disk being taken out of active use in its disk group, if it has not already been taken out of use. If the disk is still operational, which should not be the case, `vxdisk` prints:

```
device: Okay
```

If the disk is listed as “Okay,” try running `vxctl hostid` again. If it still results in an error, contact Customer Support.

vxvm:vxconfigd: ERROR: Disk group *group*: Cannot recover temp database:
reason
Consider use of "`vxconfigd -x cleartempdir`" [see `vxconfigd(1M)`].

- ◆ *Description:* This can happen if you kill and restart `vxconfigd` or you if you disable and enable it with `vxctl disable` and `vxctl enable`. This error indicates a failure related to reading the file `/var/vxvm/tempdb/group`. This is a temporary file used to store information that is used when recovering the state of an earlier `vxconfigd`. The file is recreated on a reboot, so this error should never survive a reboot.
- ◆ *Action:* If you can reboot, do so. If you do not want to reboot, then do the following:
 - a. Ensure that no `vxvol`, `vxplex`, or `vxsd` processes are running.
Use `ps -e` to search for such processes, and use `kill` to kill any that you find. You may have to run `kill` twice to make these processes go away. Killing utilities in this way may make it difficult to make administrative changes to some volumes until the system is rebooted.
 - b. Recreate the temporary database files for all imported disk groups using the following command:


```
# vxconfigd -x cleartempdir 2> /dev/console
```



The `vxvol`, `vxplex`, and `vxsd` commands make use of these `tempdb` files to communicate locking information. If the file is cleared, then locking information can be lost. Without this locking information, two utilities can end up making incompatible changes to the configuration of a volume.

```
vxvm:vxconfigd: ERROR: Disk group group: Disabled by errors
```

- ◆ *Description:* This message indicates that some error condition has made it impossible for VxVM to continue to manage changes to a disk group. The major reason for this is that too many disks have failed, making it impossible for `vxconfigd` to continue to update configuration copies. There should be a preceding error message that indicates the specific error that was encountered.

If the disk group that was disabled is the `rootdg` disk group, then the following additional error is displayed:

```
vxvm:vxconfigd: ERROR: All transactions are disabled
```

This additional message indicates that `vxconfigd` has entered the disabled state, which makes it impossible to change the configuration of any disk group, not just `rootdg`.

- ◆ *Action:* If the underlying error resulted from a transient failure, such as a disk cabling error, then you may be able to repair the situation by rebooting. Otherwise, the disk group may have to be recreated and restored from a backup. Failure of the `rootdg` disk group may require reinstallation of the system if your system uses a `root` or `/usr` file system defined on a volume.

```
vxvm:vxconfigd: ERROR: Disk group group,Disk disk:Cannot auto-import  
group: reason
```

- ◆ *Description:* On system startup, `vxconfigd` failed to import the disk group associated with the named disk. A message related to the specific failure is given in *reason*. Additional error messages may be displayed that give more information on the specific error. In particular, this is often followed by:

```
vxvm:vxconfigd: ERROR: Disk group group: Errors in some  
configuration copies:  
Disk device, copy number: Block bno: error ...
```

The most common reason for auto-import failures is excessive numbers of disk failures, making it impossible for VxVM to find correct copies of the disk group configuration database and kernel update log. Disk groups usually have enough copies of this configuration information to make such import failures unlikely.

A more serious failure is indicated by errors such as:

```
Format error in configuration copy
Invalid magic number
Invalid block number
Duplicate record in configuration
Configuration records are inconsistent
```

These errors indicate that all configuration copies have become corrupt (due to disk failures, writing on the disk by an application or the administrator, or bugs in VxVM).

Some correctable errors may be indicated by other error messages that appear in conjunction with the auto-import failure message. Look up those other errors for more information on their cause.

Failure of an auto-import implies that the volumes in that disk group will not be available for use. If there are file systems on those volumes, then the system may yield further errors resulting from inability to access the volume when mounting the file system.

- ◆ **Action:** If the error is clearly caused by excessive disk failures, then you may have to recreate the disk group and restore contents of any volumes from a backup. There may be other error messages that appear which provide further information. See those other error messages for more information on how to proceed. If those errors do not make it clear how to proceed, contact Customer Support.

```
vxvm:vxconfigd: ERROR: Disk group group, Disk disk: Group name
collides with record in rootdg
```

- ◆ **Description:** The name of a disk group that is being imported conflicts with the name of a record in the rootdg disk group. VxVM does not allow this kind of conflict because of the way the /dev/vx/dsk directory is organized: devices corresponding to records in the root disk group share this directory with subdirectories for each disk group.
- ◆ **Action:** Either remove or rename the conflicting record in the root disk group, or rename the disk group on import. See the vxdg(1M) manual page for information on how to use the import operation to rename a disk group.

```
vxvm:vxconfigd: ERROR: Disk group group, Disk disk: Skip disk group
with duplicate name
```

- ◆ **Description:** Two disk groups with the same name are tagged for auto-importing by the same host. Disk groups are identified both by a simple name and by a long unique identifier (disk group ID) assigned when the disk group is created. Thus, this error indicates that two disks indicate the same disk group name but a different disk group ID.



VxVM does not allow you to create a disk group or import a disk group from another machine, if that would cause a collision with a disk group that is already imported. Therefore, this error is unlikely to occur under normal use. However, this error can occur in the following two cases:

- A disk group cannot be auto-imported due to some temporary failure. If you create a new disk group with the same name as the failed disk group and reboot, the new disk group is imported first. The auto-import of the older disk group fails because more recently modified disk groups have precedence over older disk groups.
 - A disk group is deported from one host using the `-h` option to cause the disk group to be auto-imported on reboot from another host. If the second host was already auto-importing a disk group with the same name, then reboot of that host will yield this error.
- ◆ *Action:* If you want to import both disk groups, then rename the second disk group on import. See the `vxchg(1M)` manual page for information on how to use the `import` operation to rename a disk group.

```
vxvm:vxconfigd: ERROR: Disk group group: Errors in some configuration
copies: Disk disk, copy number: [Block number]: reason ...
```

- ◆ *Description:* During a failed disk group import, some of the configuration copies in the named disk group were found to have format or other types of errors which make those copies unusable. This message lists all configuration copies that have uncorrected errors, including any appropriate logical block number. If no other reasons are displayed, then this may be the cause of the disk group import failure.
- ◆ *Action:* If some of the copies failed due to transient errors (such as cable failures), then a reboot or reimport may succeed in importing the disk group. Otherwise, the disk group may have to be recreated from scratch.

```
vxvm:vxconfigd: ERROR: Disk group group: Reimport of disk group failed:
reason
```

- ◆ *Description:* After `vxconfigd` was stopped and restarted (or disabled and then enabled), VxVM failed to recreate the import of the indicated disk group. The reason for failure is specified. Additional error messages may be displayed that give further information describing the problem.
- ◆ *Action:* A major cause for this kind of failure is disk failures that were not addressed before `vxconfigd` was stopped or disabled. If the problem is a transient disk failure, then rebooting may take care of the condition.

```
vxvm:vxconfigd: ERROR: Disk group group: update failed: reason
```

- ◆ **Description:** I/O failures have prevented vxconfigd from updating any active copies of the disk group configuration. This usually implies a large number of disk failures. This error will usually be followed by the error:

```
vxvm:vxconfigd: ERROR: Disk group group: Disabled by errors
```

- ◆ **Action:** If the underlying error resulted from a transient failure, such as a disk cabling error, then you may be able to repair the situation by rebooting. Otherwise, the disk group may have to be recreated and restored from a backup.

```
vxvm:vxconfigd: ERROR: enable failed: reason
```

Regular startup of vxconfigd failed for the stated reason. This error can also result from the command vxctl enable. The error may include the following additional text:

```
aborting
```

The failure was fatal and vxconfigd was forced to exit. The most likely cause is that the operating system is unable to create interprocess communication channels to other utilities.

```
Error check group configuration copies. Database file not found
```

The directory /var/vxvm/tempdb is inaccessible. This may be because of root file system corruption, or if /var is a separate file system, because it has become corrupted or has not been mounted.

If /var is a separate file system, make sure that it has an entry in /etc/vfstab. Otherwise, look for I/O error messages during the boot process that indicate either a hardware problem or misconfiguration of any logical volume management software being used for the /var file system. Also verify that the encapsulation (if configured) of your boot disk is complete and correct.

```
transactions are disabled
```

vxconfigd is continuing to run, but no configuration updates are possible until the error condition is repaired.

Additionally, this may be followed with:

```
vxvm:vxconfigd: ERROR: Disk group group: Errors in some
configuration copies:
Disk device, copy number: Block bno: error ...
```

Other error messages may be displayed that further indicate the underlying problem. If the “Errors in some configuration copies” error occurs again, that may indicate the real problem.

Evaluate the error messages to determine the root cause of the problem. Make changes suggested by the errors and then try rerunning the command.



vxvm:vxconfigd: ERROR: Failed to store commit status list into kernel:
reason

vxvm:vxconfigd: ERROR: GET_VOLINFO ioctl failed: *reason*

vxvm:vxconfigd: ERROR: Get of current rootdg failed: *reason*

- ◆ *Description:* These internal errors should not occur unless there is a bug in VxVM.
- ◆ *Action:* Contact Customer Support.

vxvm:vxconfigd: ERROR: Memory allocation failure

- ◆ *Description:* This implies that there is insufficient memory to start VxVM and to get the volumes for the root and /usr file systems running.
- ◆ *Action:* This error should not normally occur, unless your system has very small amounts of memory. Adding swap space will probably not help because this error is most likely to occur early in the boot sequence, before swap areas have been added.

vxvm:vxconfigd: ERROR: *mode*: Unrecognized operating mode

- ◆ *Description:* An invalid string was specified as an argument to the -m option. Valid strings are: enable, disable, and boot.
- ◆ *Action:* Supply a correct option argument.

vxvm:vxconfigd: ERROR: Mount point *path*: volume not in rootdg disk group

- ◆ *Description:* The volume device listed in the /etc/vfstab file for the given mount-point directory (normally /usr) is listed as in a disk group other than rootdg. This error should not occur if the standard VERITAS Volume Manager procedures are used for encapsulating the disk containing the /usr file system.
- ◆ *Action:* Boot VxVM from a network or CD-ROM mounted root file system. Then, start up VxVM using fixmountroot on a valid mirror disk of the root file system. After starting VxVM, mount the root file system volume and edit the /etc/vfstab file. Change the file to use a direct partition for the file system. There should be a comment in the /etc/vfstab file that indicates which partition to use, for example:

vxvm:vxconfigd: ERROR: No convergence between root disk group and disk list

Disks in one version of rootdg: device type=*device_type* info=*devinfo* ...
Disks in alternate version of rootdg: device type=*device_type* info=*devinfo*
...

- ◆ *Description:* This message can appear when vxconfigd is not running in autoconfigure mode (see the vxconfigd(1M) manual page) and after several retries it cannot resolve the set of disks belonging to the root disk group. The algorithm for non-autoconfigure disks scans disks listed in the /etc/vx/volboot file and then examines the disks to find a database copy for the rootdg disk group. It next reads the database copy to find the list of disk access records for disks contained in the group. These disks are then examined to ensure that they contain the same database

copy. The algorithm expects to gain convergence on the set of disks and on the database copies that they contain. If a loop is entered and convergence cannot be reached, this message is displayed and the root disk group importation fails.

- ◆ **Action:** Reorganize the physical locations of the devices attached to the system to try and break the deadlock. If this does not succeed, contact Customer Support.

vxvm:vxconfigd: ERROR: Open of directory *directory* failed: *reason*

- ◆ **Description:** An open failed for `/dev/vx/dsk`, `/dev/vx/rdsk`, or one of their subdirectories. The only likely cause of such a failure is that the directory was removed by the administrator or by an errant program. If this is the case, the *reason* should be “No such file or directory.” An alternate possible cause is an I/O failure.
- ◆ **Action:** If the reason was “No such file or directory,” use `mkdir` to recreate the directory. Then run the command `vxdctl enable`.

If the error was an I/O error, there may be other serious damage to the root file system. You may need to reformat your root disk and restore the root file system from backup. Contact your system vendor or consult your system documentation.

vxvm:vxconfigd: ERROR: Read of directory *directory* failed: *reason*

- ◆ **Description:** There was a failure in reading `/dev/vx/dsk`, `/dev/vx/rdsk`, or one of their subdirectories. The only likely cause of this error is an I/O failure on the root file system.
- ◆ **Action:** If the error was an I/O error, then there may be other serious damage to the root file system. You may need to reformat your root disk and restore the root file system from backup. Contact your system vendor or consult your system documentation.

vxvm:vxconfigd: ERROR: *signal* [- core dumped]

- ◆ **Description:** The vxconfigd daemon encountered an unexpected signal while starting up. If the signal caused the vxconfigd process to dump core, then that will be indicated. This could be caused by a bug in vxconfigd, particularly if *signal* is “Segmentation fault.” Alternately, this could have been caused by a user sending vxconfigd a signal with the `kill` utility.
- ◆ **Action:** Contact Customer Support.

vxvm:vxconfigd: ERROR: System boot disk does not have a valid rootvol
plex

Please boot from one of the following disks:

DISK	MEDIA DEVICE	BOOT	COMMAND
<i>diskname</i>	<i>device</i>	<i>boot</i>	<i>vx-diskname...</i>



- ◆ *Description:* The system is configured to use a volume for the `root` file system, but was not booted on a disk containing a valid mirror of the `root` volume. Disks containing valid `root` mirrors are listed as part of the error message. A disk is usable as a boot disk if there is a `root` mirror on that disk which is not stale or offline.
- ◆ *Action:* Try to boot from one of the named disks using the associated boot command that is listed in the message.

```
vxvm:vxconfigd: ERROR: System startup failed
```

- ◆ *Description:* Either the `root` or the `/usr` file system volume could not be started, rendering the system unusable. The error that resulted in this condition should appear prior to this error message.
- ◆ *Action:* Look up other error messages appearing on the console and take the actions suggested in the descriptions of those messages.

```
vxvm:vxconfigd: ERROR: There is no volume configured for the root device
```

- ◆ *Description:* The system is configured to boot from a `root` file system defined on a volume, but there is no `root` volume listed in the configuration of the `rootdg` disk group.

There are two possible causes of this error:

- Case 1: The `/etc/system` file was erroneously updated to indicate that the `root` device is `/pseudo/vxio@0:0`. This can happen only as a result of direct manipulation by the administrator.
 - Case 2: The system somehow has a duplicate `rootdg` disk group, one of which contains a `root` file system volume and one of which does not, and `vxconfigd` somehow chose the wrong one. Since `vxconfigd` chooses the more recently accessed version of `rootdg`, this error can happen if the system clock was updated incorrectly at some point (reversing the apparent access order of the two disk groups). This can also happen if some disk group was deported and renamed to `rootdg` with locks given to this host.
- ◆ *Action:* In case 1, boot the system on a CD-ROM or networking-mounted root file system, directly mount the disk partition of the root file system, and remove the following lines from `/etc/system`:

```
rootdev:/pseudo/vxio@0:0
set vxio:vol_rootdev_is_volume=1
```

In case 2, either boot with all drives in the offending version of `rootdg` turned off, or import and rename (see `vxvg(1M)`) the offending `rootdg` disk group from another host. If you turn off drives, run the following command after booting:

```
# vxvg flush rootdg
```


This updates time stamps on the imported version of `rootdg`, which should make the correct version appear to be the more recently accessed. If this does not correct the problem, contact Customer Support.

```
vxvm:vxconfigd: ERROR: Unexpected configuration tid for group group
found in kernel
vxvm:vxconfigd: ERROR: Unexpected error during volume volume
reconfiguration: reason
vxvm:vxconfigd: ERROR: Unexpected error fetching disk for disk volume:
reason
vxvm:vxconfigd: ERROR: Unexpected values stored in the kernel
```

- ◆ **Description:** These internal errors should not occur unless there is a bug in VxVM.
- ◆ **Action:** Contact Customer Support.

```
vxvm:vxconfigd: ERROR: Version number of kernel does not match
vxconfigd
```

- ◆ **Description:** The release of `vxconfigd` does not match the release of the VERITAS Volume Manager kernel drivers. This should happen only as a result of upgrading VxVM, and then running `vxconfigd` without a reboot.
- ◆ **Action:** Reboot the system. If that does not cure the problem, re-add the VxVM packages.

```
vxvm:vxconfigd:ERROR:volume_name:vxconfigd cannot boot-start RAID-5
volumes
```

- ◆ **Description:** A volume that `vxconfigd` should start immediately upon booting the system (that is, the volume for the `/usr` file system) has a RAID-5 layout. The `/usr` file system should never be defined on a RAID-5 volume.
- ◆ **Action:** It is likely that the only recovery for this is to boot VxVM from a network-mounted root file system (or from a CD-ROM), and reconfigure the `/usr` file system to be defined on a regular non-RAID-5 volume.

```
vxvm:vxconfigd: ERROR: Volume volume for mount point /usr not found in
rootdg disk group
```

- ◆ **Description:** The system is configured to boot with `/usr` mounted on a volume, but the volume associated with `/usr` is not listed in the configuration of the `rootdg` disk group. There are two possible causes of this error:
 - Case 1: The `/etc/vfstab` file was erroneously updated to indicate the device for the `/usr` file system is a volume, but the volume named is not in the `rootdg` disk group. This should happen only as a result of direct manipulation by the administrator.
 - Case 2: The system somehow has a duplicate `rootdg` disk group, one of which contains the `/usr` file system volume and one of which does not (or uses a different volume name), and `vxconfigd` somehow chose the wrong `rootdg`.



Since `vxconfigd` chooses the more recently accessed version of `rootdg`, this error can happen if the system clock was updated incorrectly at some point (causing the apparent access order of the two disk groups to be reversed). This can also happen if some disk group was deported and renamed to `rootdg` with locks given to this host.

- ◆ *Action:* In case 1, boot the system on a CD-ROM or networking-mounted `root` file system. If the `root` file system is defined on a volume, then start and mount the `root` volume using the procedures defined in “[Recovery from Boot Disk Failure](#)” on page 19. If the `root` file system is not defined on a volume, mount the `root` file system directly. Edit the `/etc/vfstab` file to correct the entry for the `/usr` file system.

In case 2, either boot with all drives in the offending version of `rootdg` turned off, or import and rename (see `vxchg(1M)`) the offending `rootdg` disk group from another host. If you turn off drives, run the following command after booting:

```
# vxdbg flush rootdg
```

This updates time stamps on the imported version of `rootdg`, which should make the correct version appear to be the more recently accessed. If this does not correct the problem, contact Customer Support.

vxconfigd Warning Messages

A warning message from the configuration daemon, `vxconfigd`, indicates a problem that may affect the operation of VxVM. Action should be taken to correct the problem as soon as possible.

```
vxvm:vxconfigd: WARNING: Bad request number: client number, portal  
[REQUEST|DIAG], size number
```

- ◆ *Description:* This diagnostic message indicates that a utility sent an invalid request to `vxconfigd`.
- ◆ *Action:* If you are developing a new utility, this error indicates a bug in your code. Otherwise, it indicates a bug in VxVM. Contact Customer Support.

```
vxvm:vxconfigd: WARNING: Cannot change disk group record in kernel:  
reason
```

- ◆ *Description:* This internal error should not occur unless there is a bug in VxVM.
- ◆ *Action:* Contact Customer Support.

```
vxvm:vxconfigd: WARNING: Cannot create device device_path: reason
```

- ◆ *Description:* `vxconfigd` cannot create a device node either under `/dev/vx/dsk` or under `/dev/vx/rdsk`. This should happen only if the `root` file system has run out of inodes.

- ◆ **Action:** Remove some unwanted files from the `root` file system. Then, regenerate the device node using the command:

```
# vxctl enable
```

```
vxvm:vxconfigd: WARNING: Cannot exec /usr/bin/rm to remove directory:
reason
```

- ◆ **Description:** The given directory could not be removed because the `/usr/bin/rm` utility could not be executed by `vxconfigd`. This is not a serious error. The only side effect of a directory not being removed is that the directory and its contents continue to use space in the `root` file system. However, this does imply that the `/usr` file system is not mounted, or on some systems, that the `rm` utility is missing or is not in its usual location. This may be a serious problem for the general running of your system.

- ◆ **Action:** If the `/usr` file system is not mounted, you need to determine how to get it mounted. If the `rm` utility is missing, or is not in the `/usr/bin` directory, restore it.

```
vxvm:vxconfigd: WARNING: Cannot fork to remove directory directory:
reason
```

- ◆ **Description:** The given directory could not be removed because `vxconfigd` could not fork in order to run the `rm` utility. This is not a serious error. The only side effect of a directory not being removed is that the directory and its contents will continue to use space in the `root` file system. The most likely cause for this error is that your system does not have enough memory or paging space to allow `vxconfigd` to fork.

- ◆ **Action:** If your system is this low on memory or paging space, your overall system performance is probably substantially degraded. Consider adding more memory or paging space.

```
vxvm:vxconfigd: WARNING: Cannot issue internal transaction: reason
```

- ◆ **Description:** This problem usually occurs only if there is a bug in VxVM. However, it may also occur if memory is low.

- ◆ **Action:** Contact Customer Support.

```
vxvm:vxconfigd: WARNING: Cannot open log file log_filename: reason
```

- ◆ **Description:** The `vxconfigd` console output log file could not be opened for the given reason.

- ◆ **Action:** Create any needed directories, or use a different log file path name as described in “[Logging Error Messages](#)” on page 47.

```
vxvm:vxconfigd: WARNING: cannot remove group group from kernel: reason
vxvm:vxconfigd: WARNING: client number not recognized by VxVM library
vxvm:vxconfigd: WARNING: client number not recognized
```

- ◆ **Description:** These internal errors should not occur unless there is a bug in VxVM.
- ◆ **Action:** Contact Customer Support.



vxvm:vxconfigd: WARNING: Detaching plex *plex* from volume *volume*

- ◆ *Description:* This error only happens for volumes that are started automatically by vxconfigd at system startup (that is, for the root and /usr file system volumes). The plex is being detached as a result of I/O failure, disk failure during startup or prior to the last system shutdown or crash, or disk removal prior to the last system shutdown or crash.
- ◆ *Action:* To ensure that the root or /usr file system retains the same number of active mirrors, remove the given plex and add a new mirror using the vxassist mirror operation. Also consider replacing any bad disks before running this command.

vxvm:vxconfigd: WARNING: Disk *disk* in group *group* flagged as shared;
Disk skipped

- ◆ *Description:* The given disk is listed as shared, but the running version of VxVM does not support shared disk groups.
- ◆ *Action:* This message can usually be ignored. If you want to use the disk on this system, use vxdiskadd to add the disk. Do not do this if the disk really is in shared with other systems.

vxvm:vxconfigd: WARNING: Disk *disk* in group *group* locked by host *hostid*
Disk skipped

- ◆ *Description:* The given disk is listed as locked by the host with the VERITAS Volume Manager host ID (usually the same as the system hostname).
- ◆ *Action:* This message can usually be ignored. If you want to use the disk on this system, use vxdiskadd to add the disk. Do not do this if the disk really is in shared with other systems.

vxvm:vxconfigd: WARNING: Disk *disk* in group *group*: Disk device not found

- ◆ *Description:* No physical disk can be found that matches the named disk in the given disk group. This is equivalent to failure of that disk. (Physical disks are located by matching the disk IDs in the disk group configuration records against the disk IDs stored in the VERITAS Volume Manager header on the physical disks.) This error message is displayed for any disk IDs in the configuration that are not located in the disk header of any physical disk. This may result from a transient failure such as a poorly-attached cable, or from a disk that fails to spin up fast enough. Alternately, this may happen as a result of a disk being physically removed from the system, or from a disk that has become unusable due to a head crash or electronics failure.

Any RAID-5 plexes, DRL log plexes, RAID-5 subdisks or mirrored plexes containing subdisks on this disk are unusable. Such disk failures (particularly on multiple disks) may cause one or more volumes to become unusable.

- ◆ *Action:* If hot-relocation is enabled, VERITAS Volume Manager objects affected by the disk failure are taken care of automatically. Mail is sent to root indicating what actions were taken by VxVM and what further actions the administrator should take.

vxvm:vxconfigd: WARNING: Disk *disk* in kernel is not a recognized type

- ◆ *Description:* This internal error should not occur unless there is a bug in VxVM.
- ◆ *Action:* Contact Customer Support.

vxvm:vxconfigd: WARNING: Disk *disk* names group *group*, but group ID differs

- ◆ *Description:* As part of a disk group import, a disk was discovered that had a mismatched disk group name and disk group ID. This disk is not imported. This can only happen if two disk groups have the same name but have different disk group ID values. In such a case, one group is imported along with all its disks and the other group is not. This message appears for disks in the un-selected group.
- ◆ *Action:* If the disks should be imported into the group, this must be done by adding the disk to the group at a later stage, during which all configuration information for the disk is lost.

vxvm:vxconfigd: WARNING: Disk group *group*: Disk group log may be too small
Log size should be at least *number* blocks

- ◆ *Description:* The log areas for the disk group have become too small for the size of configuration currently in the group. This message only occurs during disk group import; it can only occur if the disk was inaccessible while new database objects were added to the configuration, and the disk was then made accessible and the system restarted. This should not normally happen without first displaying a message about the database area size.
- ◆ *Action:* Reinitialize the disks in the group with larger log areas. Note that this requires that you restore data on the disks from backups. See the `vxdisk(1M)` manual page. To reinitialize all of the disks, detach them from the group with which they are associated, reinitialize and re-add them. Then deport and re-import the disk group to make the changes to the log areas for the group take effect.

vxvm:vxconfigd: WARNING: Disk group *group*: Errors in some configuration copies: Disk *disk*, copy *number*: [Block *number*]: *reason* ...

- ◆ *Description:* During a disk group import, some of the configuration copies in the named disk group were found to have format or other types of errors which make those copies unusable. This message lists all configuration copies that have uncorrected errors, including any appropriate logical block number.
- ◆ *Action:* There are usually enough configuration copies in any disk group to ensure that such errors do not become a serious problem. No action is usually necessary.



vxvm:vxconfigd: WARNING: Disk group *group* is disabled, disks not updated with new host ID

- ◆ *Description:* As a result of failures, the named disk group has become disabled. Earlier error messages should indicate the cause. This message indicates that disks in that disk group were not updated with a new VERITAS Volume Manager host ID. This warning message should result only from a `vxctl hostid` operation.
- ◆ *Action:* Typically, unless a disk group was disabled due to transient errors, there is no way to repair a disabled disk group. The disk group may have to be reconstructed from scratch. If the disk group was disabled due to a transient error such as a cabling problem, then a future reboot may not automatically import the named disk group, due to the change in the system's VERITAS Volume Manager host ID. In such a case, import the disk group directly using `vxvg import` with the `-C` option.

vxvm:vxconfigd: WARNING: Error in volboot file: *reason* Entry: disk device *disk_type disk_info*

- ◆ *Description:* The `/etc/vx/volboot` file includes an invalid disk entry. This error should occur only if the file was edited directly.
- ◆ *Action:* Correct the offending entry, or remove it using the following command:

```
# vxctl rm disk device
```

vxvm:vxconfigd: WARNING: Failed to store commit status list into kernel: *reason*

vxvm:vxconfigd: WARNING: Failed to update voldinfo area in kernel: *reason*

- ◆ *Description:* These internal errors should not occur unless there is a bug in VxVM.
- ◆ *Action:* Contact Customer Support.

vxvm:vxconfigd: WARNING: Field too long in volboot file: Entry: disk device *disk_type disk_info*

- ◆ *Description:* The `/etc/vx/volboot` file includes a disk entry with a field that is larger than the size supported by VxVM. This error should occur only if the file was edited directly.
- ◆ *Action:* Correct the offending entry, or remove it using the command:

```
# vxctl rm disk device
```

vxvm:vxconfigd: WARNING: Get of record *record_name* from kernel failed: *reason*

- ◆ *Description:* This internal error should not occur unless there is a bug in VxVM.
- ◆ *Action:* Contact Customer Support.

```
vxvm:vxconfigd: WARNING: Group group: Duplicate virtual device
number(s):
Volume volume remapped from major,minor to major,minor ...
```

- ◆ *Description:* The configuration of the named disk group includes conflicting device numbers. A disk group configuration lists the recommended device number to use for each volume in the disk group. If two volumes in two disk groups happen to list the same device number, then one of the volumes must use an alternate device number. This is called device number remapping. Remapping is a temporary change to a volume. If the other disk group is deported and the system is rebooted, then the volume that was remapped may no longer be remapped. Also, volumes that are remapped once are not guaranteed to be remapped to the same device number in further reboots.
- ◆ *Action:* Use the `vx dg reminor` command to renumber all volumes in the offending disk group permanently. See the `vx dg(1M)` manual page for more information.

```
vxvm:vxconfigd: WARNING: Internal transaction failed: reason
```

- ◆ *Description:* This problem usually occurs only if there is a bug in VxVM. However, it may also occur if memory is low.
- ◆ *Action:* Contact Customer Support.

```
vxvm:vxconfigd: WARNING: library and vxconfigd disagree on existence
of client number
vxvm:vxconfigd: WARNING: library specified non-existent client number
vxvm:vxconfigd: WARNING: response to client number failed: reason
vxvm:vxconfigd: WARNING: vold_turnclient(number) failed: reason
```

- ◆ *Description:* These internal errors do not occur unless there is a bug in VxVM.
- ◆ *Action:* Contact Customer Support.

vxconfigd Notice Messages

A notice message from the configuration daemon, `vxconfigd`, indicates that VxVM has taken some action that you may wish to monitor. Action should be taken to correct any associated hardware problem as soon as possible.

```
vxvm:vxconfigd: NOTICE: Detached disk disk
```

- ◆ *Description:* The named disk appears to have become unusable and was detached from its disk group. Additional messages may appear to indicate other records detached as a result of the disk detach.
- ◆ *Action:* If hot-relocation is enabled, VERITAS Volume Manager objects affected by the disk failure are taken care of automatically. Mail is sent to `root` indicating what actions were taken by VxVM and what further actions the administrator should take.



vxvm:vxconfigd: NOTICE: Detached log for volume *volume*

- ◆ *Description:* The DRL or RAID-5 log for the named volume was detached as a result of a disk failure, or as a result of the administrator removing a disk with `vxdbg -k rmdisk`. A failing disk is indicated by a “Detached disk” message.
- ◆ *Action:* If the log is mirrored, hot-relocation tries to relocate the failed log automatically. Use either `vxplex dis` or `vxsd dis` to remove the failing logs. Then, use `vxassist addlog` (see the `vxassist(1M)` manual page) to add a new log to the volume.

vxvm:vxconfigd: NOTICE: Detached *plex* plex in volume *volume*

- ◆ *Description:* The specified plex was disabled as a result of a disk failure, or as a result of the administrator removing a disk with `vxdbg -k rmdisk`. A failing disk is indicated by a “Detached disk” message.
- ◆ *Action:* If hot-relocation is enabled, VERITAS Volume Manager objects affected by the disk failure are taken care of automatically. Mail is sent to `root` indicating what actions were taken by VxVM and what further actions the administrator should take.

vxvm:vxconfigd: NOTICE: Detached subdisk *subdisk* in volume *volume*

- ◆ *Description:* The specified subdisk was disabled as a result of a disk failure, or as a result of the administrator removing a disk with `vxdbg -k rmdisk`. A failing disk is indicated by a “Detached disk” message.
- ◆ *Action:* If hot-relocation is enabled, VERITAS Volume Manager objects affected by the disk failure are taken care of automatically. Mail is sent to `root` indicating what actions were taken by VxVM and what further actions the administrator should take.

vxvm:vxconfigd: NOTICE: Detached volume *volume*

- ◆ *Description:* The specified volume was detached as a result of a disk failure, or as a result of the administrator removing a disk with `vxdbg -k rmdisk`. A failing disk is indicated by a “Detached disk” message. Unless the disk error is transient and can be fixed with a reboot, the contents of the volume should be considered lost.
- ◆ *Action:* Contact Customer Support.

vxvm:vxconfigd: NOTICE: Offlining config copy *number* on disk *disk*:
Reason: *reason*

- ◆ *Description:* An I/O error caused the indicated configuration copy to be disabled. This is a notice only, and does not normally imply serious problems, unless this is the last active configuration copy in the disk group.
- ◆ *Action:* Consider replacing the indicated disk, since this error implies that the disk has deteriorated to the point where write errors cannot be repaired automatically. The error can also result from transient problems with cabling or power.

vxvm:vxconfigd: NOTICE: Volume *volume* entering degraded mode

- ◆ *Description:* Detaching a subdisk in the named RAID-5 volume has caused the volume to enter “degraded” mode. While in degraded mode, performance of the RAID-5 volume is substantially reduced. More importantly, failure of another subdisk may leave the RAID-5 volume unusable. Also, if the RAID-5 volume does not have an active log, then failure of the system may leave the volume unusable.
- ◆ *Action:* If hot-relocation is enabled, VERITAS Volume Manager objects affected by the disk failure are taken care of automatically. Mail is sent to `root` indicating what actions were taken by VxVM and what further actions the administrator should take.

vxdg Error Messages

An error message from the `vxdg` command indicates that the requested operation cannot be performed. Follow the recommended course of action given below.

vxvm: vxdg: ERROR: *diskgroup*: Cannot remove last disk group configuration copy

- ◆ *Description:* The requested disk group move, split or join operation would leave the disk group without any configuration copies.
- ◆ *Action:* None. The operation is not supported.

vxvm: vxdg: ERROR: *diskgroup*: Configuration too large for configuration copies

- ◆ *Description:* The disk group's configuration database is too small to hold the expanded configuration after a disk group move or join operation.
- ◆ *Action:* None.

vxvm: vxdg: ERROR: *diskgroup*: Disk group does not exist

- ◆ *Description:* The disk group does not exist or is not imported
- ◆ *Action:* Use the correct name, or import the disk group and try again.

vxvm: vxdg: ERROR: *diskgroup*: Disk group version doesn't support feature; see the `vxdg upgrade` command

- ◆ *Description:* The version of the specified disk group does not support disk group move, split or join operations.
- ◆ *Action:* Use the `vxdg upgrade diskgroup` command to update the disk group version.



vxvm: vxvg: ERROR: *diskname*: Disk is not usable

- ◆ *Description*: The specified disk has become unusable.
- ◆ *Action*: Do not include the disk in any disk group move, split or join operation until it has been replaced or repaired.

vxvm: vxvg: ERROR: *object*: Name conflicts with imported diskgroup

- ◆ *Description*: The target disk group of a split operation already exists as an imported disk group.
- ◆ *Action*: Choose a different name for the target disk group.

vxvm: vxvg: ERROR: *object*: Operation is not supported

- ◆ *Description*: DCO and snap objects dissociated by Persistent FastResync, and VVR objects cannot be moved between disk groups.
- ◆ *Action*: None. The operation is not supported.

vxvm: vxvg: ERROR: *object*: Record already exists in disk group

- ◆ *Description*: The target disk group already contains an object with the same name.
- ◆ *Action*: Rename one of the objects, or correct the request.

vxvm: vxvg: ERROR: *subdisk*: Record is associated

- ◆ *Description*: The named subdisk is not a top-level object.
- ◆ *Action*: Objects specified for a disk group move, split or join must be either disks or top-level volumes.

vxvm: vxvg: ERROR: *diskdevice*: Request crosses disk group boundary

- ◆ *Description*: The specified disk device is not configured in the source disk group for a disk group move or split operation.
- ◆ *Action*: Correct the name of the disk object specified in the disk group move or split operation.

vxvm: vxvg: ERROR: *diskgroup*: split failed: Error in cluster processing

- ◆ *Description*: The host is not the master node in the cluster.
- ◆ *Action*: Perform the operation from the master node.

vxvm: vxvg: ERROR: Transaction already in progress

- ◆ *Description*: One of the disk groups specified in a disk group move, split or join operation is currently involved in another unrelated disk group move, split or join operation (possibly as the result of recovery from a system failure).

- ◆ **Action:** Use the `vxprint` command to display the status of the disk groups involved. If `vxprint` shows that the `TUTIL0` field for a disk group is set to `MOVE`, and you are certain that no disk group move, split or join should be in progress, use the `vx dg` command to clear the field as described in “[Recovering from Incomplete Disk Group Moves](#)” on page 15. Otherwise, retry the operation.

```
vxvm: vx dg: ERROR: volume: Volume or plex device is open or mounted
```

- ◆ **Description:** An attempt was made to perform a disk group move, split or join on a disk group containing an open volume.
- ◆ **Action:** It is most likely that a file system configured on the volume is still mounted. Stop applications that access volumes configured in the disk group, and unmount any file systems configured in the volumes.

```
vxvm: vx dg: ERROR: vx dg join sourcedg targetdg failed
```

```
vxvm: vx dg: ERROR: object: Record already exists in disk group
```

- ◆ **Description:** A disk group join operation failed because the name of an object in one disk group is the same as the name of an object in the other disk group. Such name clashes are most likely to occur for snap objects and snapshot plexes.
- ◆ **Action:** Use the following command to change the object name in either one of the disk groups:

```
# vxedit -g diskgroup rename old_name new_name
```

For more information about using the `vxedit` command, see the `vxedit(1M)` manual page.

```
vxvm: vx dg: ERROR: vx dg listmove sourcedg targetdg failed
```

```
vxvm: vx dg: ERROR: diskname : Disk not moving, but subdisks on it are
```

- ◆ **Description:** Some volumes have subdisks that are not on the disks implied by the supplied list of objects.
- ◆ **Action:** Use the `-o expand` option to `vx dg listmove` to produce a self-contained list of objects.

vxdmp Notice Messages

A notice message from the Dynamic Multipathing (DMP) driver, `vxdmp`, indicates that it has taken some action that you may wish to monitor. Action should be taken to correct any associated hardware problem as soon as possible.

```
vxvm: vxdmp: NOTICE: added disk array disk_array_serial_number
```

- ◆ **Description:** A new disk array has been added to the host.
- ◆ **Action:** None.



vxvm:vxvmp:NOTICE:Attempt to disable controller *controller_name* failed.
Rootdisk has just one enabled path.

- ◆ *Description:* An attempt is being made to disable the one remaining active path to the root disk controller.

- ◆ *Action:* The path cannot be disabled.

vxvm:vxvmp:NOTICE: Could not install sd driver
vxvm:vxvmp:NOTICE: Could not install ssd driver
vxvm:vxvmp:NOTICE: Could not load sd driver
vxvm:vxvmp:NOTICE: Could not load ssd driver

- ◆ *Description:* During initialization, the vxvmp driver failed to load or install the sd or ssd driver.

- ◆ *Action:* None.

vxvm:vxvmp:NOTICE: Could not lock sd driver
vxvm:vxvmp:NOTICE: Could not lock ssd driver

- ◆ *Description:* The sd or ssd driver could not be locked during vxvmp driver initialization to avoid unloading of the driver.

- ◆ *Action:* None.

vxvm:vxvmp:NOTICE:disabled controller *controller_name* connected to disk array *disk_array_serial_number*

- ◆ *Description:* All paths through the controller connected to the disk array are disabled. This usually happens if a controller is disabled for maintenance.

- ◆ *Action:* None.

vxvm:vxvmp:NOTICE:disabled dmpnode *dmpnode_device_number*

- ◆ *Description:* A DMP node has been marked disabled in the DMP database. It will no longer be accessible for further IO requests. This occurs when all paths controlled by a DMP node are in the disabled state, and therefore inaccessible.

- ◆ *Action:* Check hardware or enable the appropriate controllers to enable at least one path under this DMP node.

vxvm:vxvmp:NOTICE:disabled path *path_device_number* belonging to dmpnode *dmpnode_device_number*

- ◆ *Description:* A path has been marked disabled in the DMP database. This path is controlled by the DMP node indicated by the specified device number. This may be due to a hardware failure.

- ◆ *Action:* Check the underlying hardware if you want to recover the desired path.

Note vxvm:vxdmp:NOTICE:enabled controller *controller_name* connected to disk array *disk_array_serial_number*

- ◆ *Description:* All paths through the controller connected to the disk array are enabled. This usually happens if a controller is enabled after maintenance.

- ◆ *Action:* None.

vxvm:vxdmp:NOTICE:enabled dmpnode *dmpnode_device_number*

- ◆ *Description:* A DMP node has been marked enabled in the DMP database. This happens when at least one path controlled by the DMP node has been enabled.

- ◆ *Action:* None.

vxvm:vxdmp:NOTICE:enabled path *path_device_number* belonging to dmpnode *dmpnode_device_number*

- ◆ *Description:* A path has been marked enabled in the DMP database. This path is controlled by the DMP node indicated by the specified device number. This happens if a previously disabled path has been repaired, the user has reconfigured the DMP database using the vxctl(1M) command, or the DMP database has been reconfigured automatically.

- ◆ *Action:* None.

vxvm:vxdmp:NOTICE: Path failure on *major/minor*

- ◆ *Description:* A path under the control of the DMP driver failed. The device major and minor numbers of the failed device is supplied in the message.

- ◆ *Action:* None.

vxvm:vxdmp:NOTICE:removed disk array *disk_array_serial_number*

- ◆ *Description:* A disk array has been disconnected from the host, or some hardware failure has resulted in the disk array becoming inaccessible to the host.

- ◆ *Action:* Replace disk array hardware if this has failed.

vxdmpadm Error Messages

An error message from the Dynamic Multipathing (DMP) administration utility, vxdmpadm, indicates a problem with the requested DMP operation.

vxvm:vxdmpadm: ERROR: Attempt to disable controller failed. One (or more) devices can be accessed only through this controller. Use the -f option if you still want to disable this controller.

- ◆ *Description:* Disabling the controller could lead to some devices becoming inaccessible.



- ◆ **Action:** To disable the only path connected to a disk, use the `-f` option.

`vxvm:vxddmpadm:ERROR:Attempt to enable a controller that is not available`

- ◆ **Description:** This message is returned by the `vxddmpadm` utility when an attempt is made to enable a controller that is not working or is not physically present.
- ◆ **Action:** Check hardware and see if the controller is present and whether I/O can be performed through it

`vxvm:vxddmpadm: ERROR:The VxVM restore daemon is already running. To change any of its parameters, stop and restart the restore daemon with the new set of arguments.`

- ◆ **Description:** The `vxddmpadm start restore` command has been executed while the restore daemon is already running.
- ◆ **Action:** Stop the restore daemon and restart it with the required set of parameters.

Cluster Error Messages

This section lists error messages that may occur with VxVM in a cluster environment. Some of these messages may appear on the console; others are returned by `vxclust`.

`Cannot assign minor minor`

- ◆ **Description:** A slave attempted to join, but an existing volume on the slave has the same minor number as a shared volume on the master.

This message should be accompanied by the following console message:

`vxvm:vxconfigd minor number minor disk group group in use`

- ◆ **Action:** Before retrying the join, use `vx dg reminor` (see the `vx dg(1M)` manual page) to choose a new minor number range either for the disk group on the master or for the conflicting disk group on the slave. If there are open volumes in the disk group, the `reminor` operation will not take effect until the disk group is deported and updated (either explicitly or by rebooting the system).

`Cannot find disk on slave node`

- ◆ **Description:** A slave node in a cluster cannot find a shared disk. This is accompanied by the `syslog` message:

`vxvm:vxconfigd cannot find disk disk`

- ◆ **Action:** Make sure that the same set of shared disks is online on both nodes. Examine the disks on both the master and the slave with the command `vx disk list` and make sure that the same set of disks with the `shared` flag is visible on both nodes. If not, check the connections to the disks.

Clustering license restricts operation

- ◆ *Description:* An operation requiring a full clustering license was attempted, and such a license is not available.
- ◆ *Action:* If the error occurs when a disk group is being activated, dissociate all but one plex from mirrored volumes before activating the disk group. If the error occurs during a transaction, deactivate the disk group on all nodes except the master.

CVM protocol version out of range

- ◆ *Description:* When a node joins a cluster, it tries to join at the protocol version that is stored in its `volboot` file. If the cluster is running at a different protocol version, the master rejects the join and sends the current protocol version to the slave. The slave re-tries with the current version (if that version is supported on the joining node), or the join fails.
- ◆ *Action:* Make sure that the joining node has a VERITAS Volume Manager release installed that supports the current protocol version of the cluster.

Disk in use by another cluster

- ◆ *Description:* An attempt was made to import a disk group whose disks are stamped with the ID of another cluster.
- ◆ *Action:* If the disk group is not imported by another cluster, retry the import using the `-C` (clear import) flag.

Disk reserved by other host

- ◆ *Description:* An attempt was made to online a disk whose controller has been reserved by another host.
- ◆ *Action:* No action is necessary. The cluster manager frees the disk and VxVM puts it online when the node joins the cluster.

Error in cluster processing

- ◆ *Description:* This may be due to an operation inconsistent with the current state of a cluster (such as an attempt to import or deport a shared disk group to or from the slave). It may also be caused by an unexpected sequence of commands from `vxclust`.
- ◆ *Action:* Make sure that the operation can be performed in the current environment.

ERROR: upgrade operation failed: Already at highest version

- ◆ *Description:* An upgrade operation has failed because a cluster is already running at the highest protocol version supported by the master.
- ◆ *Action:* No further action is possible as the master is already running at the highest protocol version it can support.

Incorrect protocol version *number* in volboot file

- ◆ *Description:* A node attempted to join a cluster where VxVM software was incorrectly upgraded or the `volboot` file is corrupted.
- ◆ *Action:* Verify the supported cluster protocol versions using `vxctl` protocol version, and reinstall VxVM if necessary.

Incorrect protocol version (*number*) in `volboot` file

- ◆ *Description:* The `volboot` file contains an incorrect protocol version. It has been corrupted, possibly by being edited manually. The `volboot` file should contain a supported protocol version before trying to bring the node into the cluster.
- ◆ *Action:* Run `vxctl init`. This writes a valid protocol version to the `volboot` file. Restart `vxconfigd` and retry the join.

Insufficient DRL log size: logging is disabled.

- ◆ *Description:* A volume with an insufficient DRL log size was started successfully, but DRL logging is disabled and a full recovery is performed.
- ◆ *Action:* Create a new DRL of sufficient size.

Join in progress

- ◆ *Description:* An attempt was made to import or deport a shared disk group during a cluster reconfiguration.
- ◆ *Action:* Retry when the cluster reconfiguration has completed.

Join not allowed now

- ◆ *Description:* A slave attempted to join a cluster when the master was not ready. The slave will retry automatically. If the retry succeeds, the following message appears:

```
vxclust: slave join complete
```
- ◆ *Action:* No action is necessary if the join eventually completes. Otherwise, investigate the cluster monitor on the master.

Master sent no data

- ◆ *Description:* During the slave join protocol, a message without data was received from the master. This message is only likely to be seen in the case of a programming error.
- ◆ *Action:* Contact Customer Support.

Missing `vxconfigd`

- ◆ *Description:* The `vxconfigd` daemon is not running.
- ◆ *Action:* Restart the `vxconfigd` daemon.

Node activation conflict

- ◆ *Description:* The disk group could not be activated because it is activated in a conflicting mode on another node in a cluster.

- ◆ *Action:* Retry later, or deactivate the disk group on conflicting nodes.

Not in cluster

- ◆ *Description:* Checking for the current protocol version (using vxctl protocol version) makes sense only if the node is in the cluster.

- ◆ *Action:* Bring the node in the cluster and retry.

NOTICE: commit: NOTE: Reason found for abort: code=2

- ◆ *Description:* This message may appear during a plex detach operation on the master in a cluster.

- ◆ *Action:* None required.

NOTICE: commit: NOTE: Reason found for abort: code=6

- ◆ *Description:* This message may appear during a plex detach operation on a slave in a cluster.

- ◆ *Action:* None required.

NOTICE: ktcvm_check: sent to slave node: node=1 mid=196

- ◆ *Description:* This message may appear during a plex detach operation on the master in a cluster.

- ◆ *Action:* None required.

NOTICE: vol_kmsg_send_wait_callback: got error 22

- ◆ *Description:* This message may appear during a plex detach operation on a slave in a cluster.

Action: None required.

Retry rolling upgrade

- ◆ *Description:* An attempt was made to upgrade the cluster to a higher protocol version when a transaction was in progress.

- ◆ *Action:* Retry at a later time.

Return from cluster_establish is Configuration daemon error 242

- ◆ *Description:* A node failed to join a cluster, or a cluster join is taking too long. If the join fails, the node retries the join automatically.

- ◆ *Action:* No action is necessary if the join is slow or a retry eventually succeeds.

This node was running different CM. Please Reboot.

- ◆ *Description:* VxVM supports clustering under the control of various cluster managers. However, once a node joins the cluster under a particular cluster manager, it cannot be restarted under a different cluster manager until it is rebooted.



- ◆ *Action:* Reboot the host machine if the cluster must be started under a different cluster manager.

Unable to add portal for cluster

- ◆ *Description:* vxconfigd was not able to create a portal for communication with the vxconfigd on the other node. This may happen in a degraded system that is experiencing shortages of system resources such as memory or file descriptors.
- ◆ *Action:* If the system does not appear to be degraded, stop and restart vxconfigd, and try again.

Upgrade operation failed: Error in cluster processing

- ◆ *Description:* The cluster protocol upgrade must be done on the master. It cannot be done from a slave node.
- ◆ *Action:* Retry the vxdctl upgrade command on the master node.

Upgrade operation failed: Retry rolling upgrade

- ◆ *Description:* No transactions should be in progress when an upgrade is tried.
- ◆ *Action:* Retry the upgrade at a later time.

Upgrade operation failed: Version out of range for at least one node

- ◆ *Description:* Before trying to upgrade a cluster by running vxdctl upgrade, all nodes should be able to support the new protocol version. An upgrade can fail if at least one of them does not support the new protocol version.
- ◆ *Action:* Make sure that the VERITAS Volume Manager package that supports the new protocol version is installed on all nodes and retry the upgrade.

Version out of range for at least one node

- ◆ *Description:* One or more nodes in the cluster do not support the protocol version that would result from a protocol upgrade.
- ◆ *Action:* Make sure that the latest version of VxVM is installed on all nodes in the cluster.

Vol recovery in progress

- ◆ *Description:* A node that crashed attempted to rejoin the cluster before its DRL map was merged into the recovery map.
- ◆ *Action:* Retry the join when the merge operation has completed.

vxconfigd not ready

node *number*: vxconfigd is not communicating properly

- ◆ *Description:* The vxconfigd daemon is not responding properly.
- ◆ *Action:* Stop and restart the vxconfigd daemon.

`vxclust not there`

- ◆ *Description:* An error during an attempt to join the cluster caused `vxclust` to fail. This may be caused by the failure of another node during a join or by the failure of `vxclust`.
- ◆ *Action:* Retry the join. An error message on the other node may clarify the problem.

`vxiod count must be above number to join cluster`

- ◆ *Description:* The number of VERITAS Volume Manager kernel daemons (`vxiod`) is less than the minimum number needed to join the cluster.
- ◆ *Action:* Increase the number of daemons using `vxiod`.

`vxvm:vxconfigd: group group exists`

- ◆ *Description:* A slave tried to join a cluster, but a shared disk group already exists in the cluster with the same name as one of its private disk groups.
- ◆ *Action:* Use the `vxdg newname` operation to rename either the shared disk group on the master, or the private disk group on the slave.

`WARNING: vxvm:vxio: Plex plex detached from volume volume`

- ◆ *Description:* This message may appear during a plex detach operation in a cluster.
- ◆ *Action:* None required.

`WARNING: vxvm:vxio: read error on plex plex of shared volume volume
offset offset length length`

- ◆ *Description:* This message may appear during a plex detach operation on the master in a cluster.
- ◆ *Action:* None required.





Index

Symbols

- /etc/system file, missing or damaged 28
- /etc/vfstab file
 - damaged 26
 - purpose 26

A

- ACTIVE plex state 2
- ACTIVE volume state 9
- aliased disks 20

B

- backup tapes, recovery 30
- badlog flag
 - clearing for DCO 16
- BADLOG plex state 8
- boot command
 - a flag 21, 28
 - s flag 29, 31
 - syntax 21
- boot disk
 - using aliases 20
- boot disks
 - alternate 20
 - configurations 19
 - hot-relocation 22
 - listing aliases 20
 - re-adding 33, 34
 - recovering from backup tape 30
 - recovery 19
 - relocating subdisks 22
 - replacing 33
- boot failure
 - cannot open altboot_disk 20
 - cannot open boot device 23
 - damaged /usr entry 27
 - due to stale plexes 24
 - due to unusable plexes 24
 - invalid partition 26

- booting system
 - aliased disks 20
 - recovery from failure 23
 - using CD-ROM 31

C

- CD-ROM, booting 31
- CLEAN plex state 2
- clusters
 - ERROR messages 82
 - NOTICE messages 85
 - WARNING messages 87

D

- data loss, RAID-5 6
- DCO
 - recovering log volumes 16
 - removing badlog flag from 16
- degraded mode, RAID-5 7
- DEGRADED volume state 7
- detached RAID-5 log plexes 11
- detached subdisks 7
- DETACHED volume kernel state 8
- devalias command 20
- devices, dump 23
- DISABLED plex kernel state 2, 8
- disabling VxVM 32
- disk group
 - recovery from failed move, split or join 15
- disk group errors
 - name conflict 63
 - new host ID 61
- disks
 - aliased 20
 - causes of failure 1
 - cleaning up configuration 44
 - failures 7
 - invalid partition 26



reattaching 5
dumpadm command 23

E

eeeprom
 used to allow boot disk aliases 20
EEPROM variables
 use-nvramrc? 20
EMPTY plex state 2
ENABLED plex kernel state 2
ENABLED volume kernel state 9
error messages
 /dev/vx/info error message 60
 A virtual disk device is open 59
 aborting 65
 All transactions are disabled 62
 Already at highest version 83
 Attempt to disable controller failed 81
 Attempt to enable a controller that is not available 82
 can't import diskgroup 60
 Can't locate disk(s) 60
 Can't open boot device 23
 Cannot assign minor 82
 Cannot auto-import group 62
 Cannot find disk on slave node 82
 Cannot get all disk groups from the kernel 57
 Cannot get all disks from the kernel 57
 Cannot get kernel transaction state 57
 Cannot get private storage from the kernel 57
 Cannot get private storage size from the kernel 57
 Cannot get record from the kernel 57
 Cannot kill existing daemon 57
 Cannot make directory 57
 cannot open /dev/vx/config 58
 Cannot open /etc/vfstab 58
 cannot open altboot_disk 20
 Cannot recover operation in progress 59
 Cannot recover temp database 61
 Cannot remove last disk group configuration copy 77
 Cannot reset VxVM kernel 59
 Cannot start volume 59
 Cannot store private storage into the kernel 60
 Clustering license restricts operation 83

Configuration records are inconsistent 63
Configuration too large for configuration copies 77
core dumped 67
CVM protocol version out of range 83
Database file not found 65
default log file 47
Device is already open 58
Differing version of vxconfig installed 60
Disabled by errors 62
Disk group does not exist 15, 77
Disk group errors
 multiple disk failures 62
Disk group version doesn't support feature 77
Disk in use by another cluster 83
Disk is not usable 78
Disk not moving, but subdisks on it are 79
Disk reserved by another host 83
Disk write failure 61
Duplicate record in configuration 63
enable failed 65
Error check group configuration copies 65
Error in cluster processing 78, 83, 86
Errors in some configuration copies 62, 64, 65
Failed to get group from the kernel 59
Failed to store commit status list into kernel 66
File just loaded does not appear to be executable 26
Format error in configuration copy 63
Get of current rootdg failed 66
GET_VOLINFO ioctl failed 66
group exists 87
Group name collides with record in rootdg 63
Incorrect protocol version in volboot file 83
Insufficient DRL log size, logging is disabled 84
Invalid block number 63
Invalid magic number 63
Join in progress 84
Join not allowed now 84



- logging 47
- Master sent no data 84
- Memory allocation failure 66
- Missing vxconfigd 84
- Name conflicts with imported diskgroup 78
- No convergence between root disk group and disk list 66
- No such device or address 58
- No such file or directory 57, 58, 67
- no valid complete plexes 59
- no valid plexes 59
- Node activation conflict 84
- Not in cluster 85
- not updated with new host ID 61
- Open of directory failed 67
- Operation is not supported 78
- RAID-5 plex does not map entire volume length 13
- Read of directory failed 67
- Record already exists in disk group 78, 79
- Record is associated 78
- Reimport of disk group failed 64
- Request crosses disk group boundary 78
- Retry rolling upgrade 85, 86
- Return from cluster_establish is Configuration daemon error 85
- Segmentation fault 67
- Skip disk group with duplicate name 63
- slave join complete 84
- some subdisks are unusable and the parity is stale 13
- split failed 78
- startup script 48
- System boot disk does not have a valid root plex 25
- System boot disk does not have a valid rootvol plex 67
- System startup failure 25, 68
- The VxVM restore daemon is already running 82
- There is no volume configured for the root device 68
- This node was running different CM 85
- Transaction already in progress 78
- transactions are disabled 65
- Unable to add portal for cluster 86
- Unexpected configuration tid for group

- found in kernel 69
- Unexpected error during volume reconfiguration 69
- Unexpected error fetching disk for disk volume 69
- Unexpected values stored in kernel 69
- Unrecognized operating mode 66
- update failed 64
- Upgrade operation failed 86
- upgrade operation failed 83
- Version number of kernel does not match vxconfigd 69
- Version out of range for at least one node 86
- Vol recovery in progress 86
- Volume for mount point /usr not found in rootdg disk group 69
- Volume is not startable 13
- volume not in rootdg disk group 66
- Volume or plex device is open or mounted 79
- Volume record id is not found in the configuration 55
- volume state is invalid 59
- vxclust not there 87
- vxconfigd cannot boot-start RAID-5 volumes 69
- vxconfigd is not communicating properly 86
- vxconfigd minor number in use 82
- vxconfigd not ready 86
- vx dg join sourcedg targetdg failed 79
- vx dg listmove failed 79
- vxiod count must be above number to join cluster 87

F

- failures
 - disk 7
 - system 6
- fatal error messages
 - Cannot update kernel 56
 - Inconsistency -- Not loaded into kernel 56
 - Interprocess communication failure 56
 - Invalid status stored in kernel 56
 - Memory allocation failure during startup 56
 - Rootdg cannot be imported during boot



-
- 56
 - Unexpected threads failure 56
 - H**
 - hardware failure, recovery from 1
 - hot-relocation
 - boot disks 22
 - defined 1
 - RAID-5 9
 - root disks 22
 - starting up 45
 - I**
 - install-db file 32, 39
 - IOFAIL plex state 3
 - K**
 - kernel
 - NOTICE messages 54
 - PANIC messages 49
 - WARNING messages 49
 - L**
 - listing
 - alternate boot disks 20
 - unstartable volumes 4
 - LOG plex state 8
 - log plexes
 - importance for RAID-5 6
 - recovering RAID-5 11
 - log volumes
 - recovering for DCO 16
 - M**
 - mirrored volumes, recovering 4
 - MOVE flag
 - set in TUTILO field 15
 - N**
 - NEEDSYNC volume state 10
 - notice messages
 - added disk array 79
 - Attempt to disable controller failed 80
 - Can't close disk 54
 - Can't open disk 54
 - Could not install sd driver 80
 - Could not install ssd driver 80
 - Could not load sd driver 80
 - Could not load ssd driver 80
 - Could not lock sd driver 80
 - Could not lock ssd driver 80
 - Detached disk 75
 - Detached log for volume 76
 - Detached plex in volume 76
 - Detached subdisk in volume 76
 - Detached volume 76
 - disabled controller connected to disk array 80
 - disabled dmpnode 80
 - disabled path belonging to dmpnode 80
 - enabled controller connected to disk array 81
 - enabled dmpnode 81
 - enabled path belonging to dmpnode 81
 - ktcvn_check sent to slave node 85
 - Offlining config copy 76
 - Path failure 81
 - read error on object 54
 - Reason found for abort 85
 - removed disk array 81
 - Rootdisk has just one enabled path 80
 - vol_kmsg_sent_wait_callback got error 85
 - Volume entering degraded mode 77
 - O**
 - OpenBoot PROMs (OPB) 21
 - P**
 - panic messages
 - Object association depth overflow 49
 - parity
 - regeneration checkpointing 11
 - resynchronization for RAID-5 10
 - stale 6
 - partitions, invalid 26
 - plex kernel states
 - DISABLED 2, 8
 - ENABLED 2
 - plex states
 - ACTIVE 2
 - BADLOG 8
 - CLEAN 2
 - EMPTY 2
 - IOFAIL 3
 - LOG 8
 - STALE 5
 - plexes
 - defined 2
 - mapping problems 12
 - recovering mirrored volumes 4
 - primary boot disk failure 20



PROMs, boot 21

R

RAID-5

- detached subdisks 7
- failures 6
- hot-relocation 9
- importance of log plexes 6
- parity resynchronization 10
- recovering log plexes 11
- recovering stale subdisk 11
- recovering volumes 9
- recovery process 8
- stale parity 6
- starting forcibly 14
- starting volumes 12
- startup recovery process 8
- subdisk move recovery 12
- unstartable volumes 12

reattaching disks 5

reconstructing-read mode, stale subdisks 7

recovery

- disk 5

reinstalling entire system 36

replacing

- boot disks 33

REPLAY volume state 8

restarting disabled volumes 4

restoring /etc/system file 28

resynchronization

- RAID-5 parity 10

root disks

- booting alternate 20
- configurations 19
- hot-relocation 22
- re-adding 33, 34
- recovering from backup tape 30
- recovery 19
- repairing 30
- replacing 33

root file system

- backing up 30
- configurations 19
- restoring 31

root file system, damaged 36

rootability

- cleaning up 40
- reconfiguring 44

S

stale parity 6

stale subdisks 7

subdisks

- detached 7
- marking as non-stale 14
- recovering after moving for RAID-5 12
- recovering stale RAID-5 11
- stale, starting volume 14
- unrelocating to replaced boot disk 22

swap space

- configurations 19

SYNC volume state 8, 10

system

- reinstalling 36

system failures 6

T

TUTIL0 field

- clearing MOVE flag 15

U

ufsdump 30

ufsrestore

- used to restore UFS file system 31

use-nvramrc? 20

usr file system

- backing up 30
- configurations 19
- repairing 30
- restoring 31

V

VM disks, aliased 20

volume kernel states

- DETACHED 8
- ENABLED 9

volume states

- ACTIVE 9
- DEGRADED 7
- NEEDSYNC 10
- REPLAY 8
- SYNC 8, 10

volumes

- cleaning up 40
- listing unstartable 4
- RAID-5 data loss 6
- reconfiguring 44
- recovering mirrors 4
- recovering RAID-5 9



- restarting disabled 4
 - stale subdisks, starting 14
- VRTSexplorer x
- vxassist
 - ERROR messages 55
 - WARNING messages 55
- vxconfigd
 - ERROR messages 57
 - FATAL ERROR messages 56
 - NOTICE messages 75
 - WARNING messages 70
- vxconfigd.log file 47
- vxdc
 - used to remove badlog flag from DCO 16
- vxdg
 - ERROR messages 77
 - used to recover from failed disk group move, split or join 15
- vxdump
 - NOTICE messages 79
- vxdumpadm
 - ERROR messages 81
- vxinfo command 4
- vxmend command 4
- vxplex command 11
- vxreattach command 5
- vxunreloc command 22
- VxVM
 - disabling 32
 - obtaining system information x
 - RAID-5 recovery process 8
 - recovering configuration of 38
 - reinstalling 38
- vxvol recover command 11
- vxvol resync command 10
- vxvol start command 5

W

- warning messages
 - Bad request 70
 - Cannot change disk group record in kernel 70
 - Cannot create device 70
 - Cannot exec /bin/rm to remove directory 71
 - Cannot exec /usr/bin/rm to remove directory 71
 - Cannot find device number 49

- Cannot fork to remove directory 71
- Cannot issue internal transaction 71
- Cannot open log file 71
- Cannot remove group from kernel 71
- check_ilocks 50
- client not recognized 71
- client not recognized by VxVM 71
- corrupt label_sdo 26
- Detaching plex from volume 72
- detaching RAID-5 50
- Disk device not found 72
- Disk group is disabled 74
- Disk group log may be too small 73
- Disk in group flagged as shared 72
- Disk in group locked by host 72
- Disk in kernel is not a recognized type 73
- Disk names group but group ID differs 73
- Disk skipped 72
- disks not updated with new host ID 74
- Double failure condition detected on RAID-5 50
- Duplicate virtual device number(s) 75
- Error in volboot file 74
- Errors in some configuration copies 73
- Failed to log the detach of the DRL volume 51
- Failed to store commit status list into kernel 74
- Failed to update voldinfo area in kernel 74
- Failure in RAID-5 logging operation 51
- Field too long in volboot file 74
- Get of record from kernel failed 74
- Illegal vminor encountered 51
- Internal transaction failed 75
- Kernel log full 51
- Kernel log update failed 51
- library and vxconfigd disagree on existence of client 75
- library specified non-existent client 75
- log object detached from RAID-5 volume 51
- Log size should be at least 73
- mod_install returned errno 52
- object detached from RAID-5 volume 52
- object plex detached from volume 52
- overlapping ilocks 50
- Overlapping mirror plex detached from



volume 52	stranded ilock 50
Plex detached from volume 87	subdisk failed in plex 53
Plex for root volume is stale or unusable	unable to read label 26
25	Uncorrectable read error 52
RAID-5 volume entering degraded	Uncorrectable write error 52
mode operation 53	vold_turnclient failed 75
read error on mirror plex of volume 53	volume already has at least one snapshot
read error on plex of shared volume 87	plex 55
Received spurious close 50	volume is detached 50
response to client failed 75	Volume remapped 75
Root volumes are not supported on your	write error on mirror plex of volume 54
PROM version 53	

