Veritas Storage Foundation™ and High Availability Solutions Troubleshooting Guide

Solaris

6.0.1

July 2012

Veritas Storage Foundation™ and High Availability Solutions Troubleshooting Guide

The software described in this book is furnished under a license agreement and may be used only in accordance with the terms of the agreement.

Product version: 6.0.1

Document version: 6.0.1 Rev 0

Legal Notice

THE DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID. SYMANTEC CORPORATION SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS DOCUMENTATION. THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

The Licensed Software and Documentation are deemed to be commercial computer software as defined in FAR 12.212 and subject to restricted rights as defined in FAR Section 52.227-19 "Commercial Computer Software - Restricted Rights" and DFARS 227.7202, "Rights in Commercial Computer Software or Commercial Computer Software Documentation", as applicable, and any successor regulations. Any use, modification, reproduction release, performance, display or disclosure of the Licensed Software and Documentation by the U.S. Government shall be solely in accordance with the terms of this Agreement.

Symantec Corporation
350 Ellis Street
Mountain View, CA 94043

http://www.symantec.com

Technical Support

Symantec Technical Support maintains support centers globally. Technical Support's primary role is to respond to specific queries about product features and functionality. The Technical Support group also creates content for our online Knowledge Base. The Technical Support group works collaboratively with the other functional areas within Symantec to answer your questions in a timely fashion. For example, the Technical Support group works with Product Engineering and Symantec Security Response to provide alerting services and virus definition updates.

Symantec's support offerings include the following:

- A range of support options that give you the flexibility to select the right amount of service for any size organization
- Telephone and/or Web-based support that provides rapid response and up-to-the-minute information
- Upgrade assurance that delivers software upgrades
- Global support purchased on a regional business hours or 24 hours a day, 7 days a week basis
- Premium service offerings that include Account Management Services

For information about Symantec's support offerings, you can visit our Web site at the following URL:

www.symantec.com/business/support/index.jsp

All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policy.

Contacting Technical Support

Customers with a current support agreement may access Technical Support information at the following URL:

www.symantec.com/business/support/contact_techsupp_static.jsp

Before contacting Technical Support, make sure you have satisfied the system requirements that are listed in your product documentation. Also, you should be at the computer on which the problem occurred, in case it is necessary to replicate the problem.

When you contact Technical Support, please have the following information available:

- Product release level

- Hardware information
- Available memory, disk space, and NIC information
- Operating system
- Version and patch level
- Network topology
- Router, gateway, and IP address information
- Problem description:
  - Error messages and log files
  - Troubleshooting that was performed before contacting Symantec
  - Recent software configuration changes and network changes

Licensing and registration

If your Symantec product requires registration or a license key, access our technical support Web page at the following URL:

www.symantec.com/business/support/

Customer service

Customer service information is available at the following URL:

www.symantec.com/business/support/

Customer Service is available to assist with non-technical questions, such as the following types of issues:

- Questions regarding product licensing or serialization
- Product registration updates, such as address or name changes
- General product information (features, language availability, local dealers)
- Latest information about product updates and upgrades
- Information about upgrade assurance and support contracts
- Information about the Symantec Buying Programs
- Advice about Symantec's technical support options
- Nontechnical presales questions
- Issues that are related to CD-ROMs or manuals

Support agreement resources

If you want to contact Symantec regarding an existing support agreement, please contact the support agreement administration team for your region as follows:

Asia-Pacific and Japan	customercare_apac@symantec.com
Europe, Middle-East, and Africa	semea@symantec.com
North America and Latin America	supportsolutions@symantec.com

Documentation

Product guides are available on the media in PDF format. Make sure that you are using the current version of the documentation. The document version appears on page 2 of each guide. The latest product documentation is available on the Symantec Web site.

https://sort.symantec.com/documents

Your feedback on product documentation is important to us. Send suggestions for improvements and reports on errors or omissions. Include the title and document version (located on the second page), and chapter and section titles of the text on which you are reporting. Send feedback to:

doc_feedback@symantec.com

For information regarding the latest HOWTO articles, documentation updates, or to ask a question regarding product documentation, visit the Storage and Clustering Documentation forum on Symantec Connect.

https://www-secure.symantec.com/connect/storage-management/forums/storage-and-clustering-documentation

About Symantec Connect

Symantec Connect is the peer-to-peer technical community site for Symantec's enterprise customers. Participants can connect and share information with other product users, including creating forum posts, articles, videos, downloads, blogs and suggesting ideas, as well as interact with Symantec product teams and Technical Support. Content is rated by the community, and members receive reward points for their contributions.

http://www.symantec.com/connect/storage-management

Contents

Chapter 1

Introduction

This chapter includes the following topics:

- About troubleshooting Veritas Storage Foundation and High Availability Solutions products
- About Symantec Operations Readiness Tools
- About unique message identifiers

About troubleshooting Veritas Storage Foundation and High Availability Solutions products

This document describes common issues that you might encounter when using Veritas Storage Foundation and High Availability Solutions and provides possible solutions for those issues. In addition to the troubleshooting information in this document, see the appropriate Symantec product *Release Notes* document for known issues and software limitations.

About Symantec Operations Readiness Tools

Symantec Operations Readiness Tools (SORT) is a Web site that automates and simplifies some of the most time-consuming administrative tasks. SORT helps you manage your datacenter more efficiently and get the most out of your Symantec products.

SORT can help you do the following:

Prepare for your next installation or upgrade	■ List product installation and upgrade requirements, including operating system versions, memory, disk space, and architecture. ■ Analyze systems to determine if they are ready to install or upgrade Symantec products. ■ Download the latest patches, documentation, and high availability agents from a central repository. ■ Access up-to-date compatibility lists for hardware, software, databases, and operating systems.
Manage risks	■ Get automatic email notifications about changes to patches, array-specific modules (ASLs/APMs/DDIs/DDLs), and high availability agents from a central repository. ■ Identify and mitigate system and environmental risks. ■ Display descriptions and solutions for hundreds of Symantec error codes.
Improve efficiency	■ Find and download patches based on product version and platform. ■ List installed Symantec products and license keys. ■ Tune and optimize your environment.

Note: Certain features of SORT are not available for all products. Access to SORT is available at no extra cost.

To access SORT, go to:

https://sort.symantec.com

Enhancements to collecting a VxExplorer troubleshooting archive

The SORT data collector contains functionality to collect and submit a VxExplorer archive. You can send this archive to Symantec Technical Support for problem diagnosis and troubleshooting. VxExplorer does not collect customer data.

The legacy VxExplorer script is deprecated and is not included in the Veritas Storage Foundation and High Availability Solutions 6.0.1 release.

You can use the SORT data collector to optionally do the following:

- Collect VxExplorer data from multiple servers using operating system based Remote Shell (RSH) or Secure Shell (SSH) remote functionality.
- Automatically upload your data to the `/incoming` directory of ftp.veritas.com.

- Determine if a specified server is part of a cluster and add the other cluster node(s) to the data collection/VxExplorer process (partial cluster check).
- Automatically upgrade the data collector script. If the server does not have an Internet connection, this functionality is automatically disabled. You can also manually disable automatic upgrades by updating the `sortdc.conf` configuration file.

About the SORT data collector

The SORT data collector is included in the VRTSspt package and is installed on the system during SFHA Solutions installation.

Standard data collector features include the following:

- It is written in a standard readable Perl script.
- The same tool is used to generate SORT custom reports as well as collect a VxExplorer archive.
- The data collector log and the SFHA Solutions installation log have the same format. Data collector logs are stored in the `~/sort/log/` directory.

To learn more about using the data collector to collect a VxExplorer archive, see:

http://www.symantec.com/docs/HOWTO32575

About unique message identifiers

Symantec enterprise products display Unique Message Identifier (UMI) message codes. UMIs include errors, warnings, and informational messages. If you get a UMI, you can use the Symantec Operations Readiness Tools (SORT) to find the message description and solution.

See “About Symantec Operations Readiness Tools” on page 15.

See “Using Symantec Operations Readiness Tools to find a Unique Message Identifier description and solution” on page 17.

Using Symantec Operations Readiness Tools to find a Unique Message Identifier description and solution

You can use Symantec Operations Readiness Tools (SORT) to find a Unique Message Identifier (UMI) description and solution.

To find a Unique Message Identifier description and solution

1. Point your Web browser to the following URL:

   http://sort.symantec.com

2. In the search field on the top right of any SORT page, enter the UMI code, and then click the search icon.

3. On the **Search Result** page, in the **Error codes** pane, click the link to your message code. If you have a large number of search results, use the check boxes at the top of the page to display only error codes to find your code more easily.

   The **Error Code details** page for the UMI code displays, which provides the description and any possible solutions.

4. If the information on the page does not provide an adequate solution to your issue, you can click one of the links on the page to do one of the following things:

   - Comment on the UMI or its solution.
   - Request a solution.
   - Add a solution of your own.

Section 1

Troubleshooting Veritas File System

Chapter 2

Diagnostic messages

This chapter includes the following topics:

File system response to problems

When the file system encounters problems, it responds in one of the following ways:

Marking an inode bad

Inodes can be marked bad if an inode update or a directory-block update fails. In these types of failures, the file system does not know what information is on the disk, and considers all the information that it finds to be invalid. After an inode is marked bad, the kernel still permits access to the file name, but any attempt to access the data in the file or change the inode fails.

Disabling transactions

If the file system detects an error while writing the intent log, it disables transactions. After transactions are disabled, the files in the file system can still be read or written, but no block or inode frees or allocations, structural changes, directory entry changes, or other changes to metadata are allowed.

Disabling a file system

If an error occurs that compromises the integrity of the file system, VxFS disables itself. If the intent log fails or an inode-list error occurs, the super-block is ordinarily updated (setting the `VX_FULLFSCK` flag) so that the next `fsck` does a full structural check. If this super-block update fails, any further changes to the file system can cause inconsistencies that are undetectable by the intent log replay. To avoid this situation, the file system disables itself.

Recovering a disabled file system

When the file system is disabled, no data can be written to the disk. Although some minor file system operations still work, most simply return `EIO`. The only thing that can be done when the file system is disabled is to do a umount and run a full `fsck`.

Although a log replay may produce a clean file system, do a full structural check to be safe.

The file system usually becomes disabled because of disk errors. Disk failures that disable a file system should be fixed as quickly as possible.

See the `fsck_vxfs`(1M) manual page.

To execute a full structural check

- Use the `fsck` command to execute a full structural check:

```
# fsck -F vxfs -o full -y /dev/vx/rdsk/diskgroup/volume
```

Warning: Be careful when running this command. By specifying the `-y` option, all `fsck` user prompts are answered with a “yes", which can make irreversible changes if it performs a full file system check.

About kernel messages

Kernel messages are diagnostic or error messages generated by the Veritas File System (VxFS) kernel. Each message has a description and a suggestion on how to handle or correct the underlying problem.

When a VxFS kernel message displays on the system console, it is preceded by a numerical ID shown in the msgcnt field. This ID number increases with each instance of the message to guarantee that the sequence of events is known when analyzing file system problems.

Each message is also written to an internal kernel buffer that you can view in the file `/var/adm/messages`.

In some cases, additional data is written to the kernel buffer. For example, if an inode is marked bad, the contents of the bad inode are written. When an error message is displayed on the console, you can use the unique message ID to find the message in `/var/adm/messages` and obtain the additional information.

Section 2

Troubleshooting Veritas Volume Manager

Recovering from hardware failure

This chapter includes the following topics:

About recovery from hardware failure

Symantec's Veritas Volume Manager (VxVM) protects systems from disk and other hardware failures and helps you to recover from such events. Recovery procedures 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 about administering hot-relocation, see the *Veritas Storage Foundation Administrator's Guide*.

Recovery from failures of the boot (`root`) disk and repairing the `root` (`/`) and `usr` file systems require the use of the special procedures.

See "VxVM and boot disk failure" on page 60.

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.

To list unstartable volumes

- Type the following command:

```
# 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
```

Displaying volume and plex states

To display detailed information about the configuration of a volume including its state and the states of its plexes, use the `vxprint` command.

To display volume and plex states

◆ Type the following command:

```
# vxprint [-g diskgroup] -hvt [volume ...]
```

The following example shows a disabled volume, `vol`, which has two clean plexes, `vol-01` and `vol-02`, each with a single subdisk:

```
# vxprint -g mydg -hvt vol
Disk group: mydg

V   NAME    RVG/VSET/CO  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   NVOLLAYR LENGTH    [COL/]OFF  AM/NM    MODE
SC  NAME    PLEX         CACHE     DISKOFFS LENGTH    [COL/]OFF  DEVICE   MODE
DC  NAME    PARENTVOL    LOGVOL
SP  NAME    SNAPVOL      DCO

v   vol        -         DISABLED  ACTIVE   212880   SELECT     -        fsgen
pl  vol-01     vol       DISABLED  CLEAN    212880   CONCAT     -        RW
sd  mydg11-01  vol-01    mydg11    0        212880   0          c1t0d0   ENA
pl  vol-02     vol       DISABLED  CLEAN    212880   CONCAT     -        RW
sd  mydg12-01  vol-02    mydg12    0        212880   0          c1t1d0   ENA
```

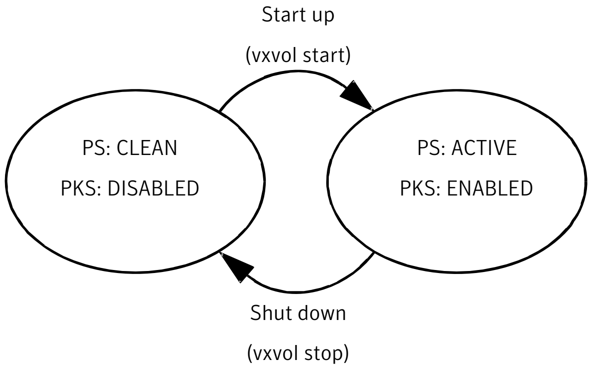
See the *Veritas Storage Foundation Administrator’s Guide* for a description of the possible plex and volume states.

The plex state cycle

Changing plex states are part of normal operations, and do not necessarily indicate abnormalities that must be corrected. A clear understanding of the various plex states and their interrelationship is necessary if you want to be able to perform any recovery procedures.

Figure 3-1 shows the main transitions that take place between plex states in VxVM.

Figure 3-1 Main plex state cycle



PS = plex state

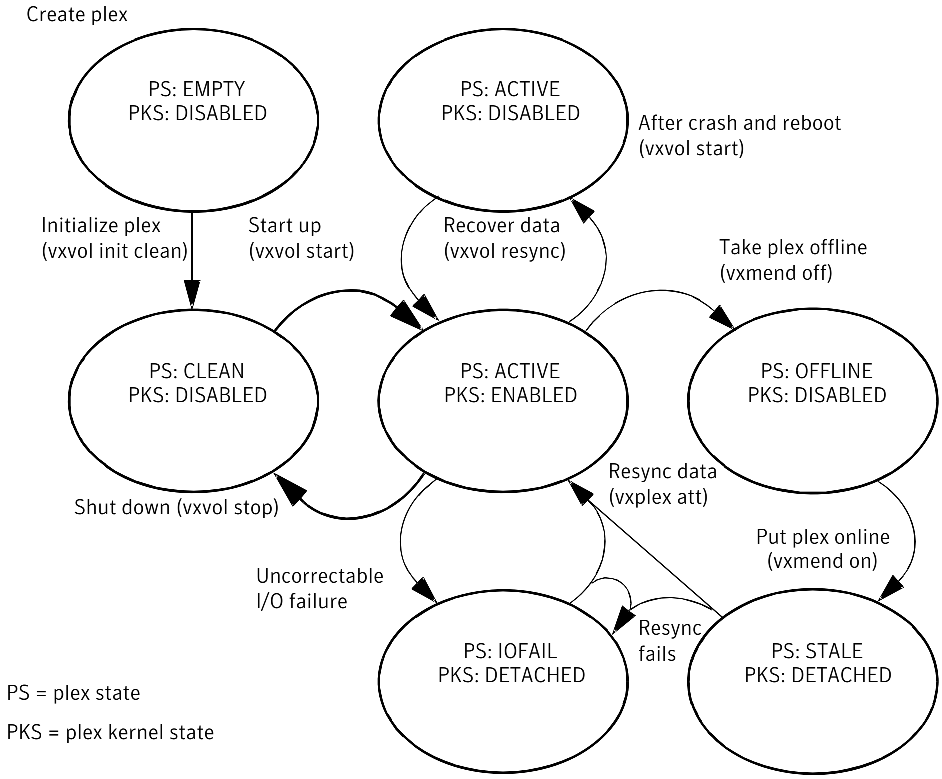
PKS = plex kernel state

For more information about plex states, see the *Veritas Storage Foundation Administrator's Guide*.

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.

Figure 3-2 shows additional transitions that are possible between plex states as a result of hardware problems, abnormal system shutdown, and intervention by the system administrator.

Figure 3-2 Additional plex state transitions



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.

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.

There are various 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.

See “Recovering an unstartable mirrored volume” on page 31.

See “Failures on RAID-5 volumes” on page 36.

Recovering an unstartable 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.

To recover an unstartable mirrored volume

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

   ```
   # vxmend [-g diskgroup] fix clean plex
   ```

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

   ```
   # vxmend -g mydg 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 [-g diskgroup] fix stale plex
   ```

   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.

   See the vxmend(1M) manual page.

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

   ```
   # vxvol [-g diskgroup] start volume
   ```

   For example, to recover volume vol01:

   ```
   # vxvol -g mydg start vol01
   ```

   See the vxvol(1M) manual page.

Recovering an unstartable volume with a disabled plex in the RECOVER state

A plex is shown in the RECOVER state if its contents are out-of-date with respect to the volume. This can happen if a disk containing one or more of the plex's subdisks has been replaced or reattached. If a plex is shown as being in this state, it can be recovered by using the `vxmend` and `vxvol` commands.

To recover an unstartable volume with a disabled plex in the RECOVER state

1. Use the following command to force the plex into the OFFLINE state:

   ```
   # vxmend [-g diskgroup] -o force off plex
   ```

2. Place the plex into the STALE state using this command:

   ```
   # vxmend [-g diskgroup] on plex
   ```

3. If there are other ACTIVE or CLEAN plexes in the volume, use the following command to reattach the plex to the volume:

   ```
   # vxplex [-g diskgroup] att volume plex
   ```

   If the volume is already enabled, resynchronization of the plex is started immediately.

   If there are no other clean plexes in the volume, use this command to make the plex DISABLED and CLEAN:

   ```
   # vxmend [-g diskgroup] fix clean plex
   ```

4. If the volume is not already enabled, use the following command to start it, and preform any resynchronization of the plexes in the background:

   ```
   # vxvol [-g diskgroup] -o bg start volume
   ```

   If the data in the plex was corrupted, and the volume has no ACTIVE or CLEAN redundant plexes from which its contents can be resynchronized, it must be restored from a backup or from a snapshot image.

Forcibly restarting a disabled volume

If a disk failure caused a volume to be disabled, and the volume does not contain any valid redundant plexes, 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.

To forcibly restart a disabled volume

- Type the following command:

  ```
  # vxvol [-g diskgroup] -o bg -f start volume
  ```

  The `-f` option forcibly restarts the volume, and the `-o bg` option resynchronizes its plexes as a background task. For example, to restart the volume `myvol` so that it can be restored from backup, use the following command:

  ```
  # vxvol -g mydg -o bg -f start myvol
  ```

Clearing the failing flag on a disk

If I/O errors are intermittent rather than persistent, Veritas Volume Manager sets the `failing` flag on a disk, rather than detaching the disk. Such errors can occur due to the temporary removal of a cable, controller faults, a partially faulty LUN in a disk array, or a disk with a few bad sectors or tracks.

If the hardware fault is not with the disk itself (for example, it is caused by problems with the controller or the cable path to the disk), you can use the `vxedit` command to unset the `failing` flag after correcting the source of the I/O error.

Warning: Do not unset the `failing` flag if the reason for the I/O errors is unknown. If the disk hardware truly is failing, and the flag is cleared, there is a risk of data loss.

To clear the failing flag on a disk

1. Use the `vxdisk list` command to find out which disks are failing:

   ```
   # vxdisk list

   DEVICE         TYPE           DISK       GROUP    STATUS
      c1t1d0s2       auto:sliced    mydg01     mydg     online
      c1t2d0s2       auto:sliced    mydg02     mydg     online failing
      c1t3d0s2       auto:sliced    mydg03     mydg     online
      . . .
   ```

2. Use the `vxedit set` command to clear the flag for each disk that is marked as `failing` (in this example, `mydg02`):

   ```
   # vxedit -g mydg set failing=off mydg02
   ```

3. Use the `vxdisk list` command to verify that the `failing` flag has been cleared:

   ```
   # vxdisk list

   DEVICE         TYPE           DISK       GROUP    STATUS
      c1t1d0s2       auto:sliced    mydg01     mydg     online
      c1t2d0s2       auto:sliced    mydg02     mydg     online
      c1t3d0s2       auto:sliced    mydg03     mydg     online
      . . .
   ```

Reattaching failed disks

You can perform a reattach operation if a disk could not be found at system startup, 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 (such as a cable or controller fault), 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 that it was located in before and retains its original disk media name.

To reattach a failed disk

1 Use the `vxdisk list` command to see which disks have failed, as shown in the following example:

```
# vxdisk list
  DEVICE     TYPE          DISK     GROUP   STATUS
  c1t1d0s2   auto:sliced   mydg01   mydg    online
  c1t2d0s2   auto:sliced   mydg02   mydg    online
  -          -             mydg03   mydg    failed was: c1t3d0s2
  -          -             mydg04   mydg    failed was: c1t4d0s2
```

2 Once the fault has been corrected, the disks can be reattached by using the following command to rescan the device list:

```
# /usr/sbin/vxdctl enable
```

3 Use the `vxreattach` command with no options to reattach the disks:

```
# /etc/vx/bin/vxreattach
```

After reattachment takes place, recovery may not be necessary unless a disk was faulty and had to be replaced. 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.

Recovering from a failed plex attach or synchronization operation

A plex attach operation requires that the plex be synchronized with the existing plexes in the volume. Other operations such as creating a mirror for a volume also require plex synchronization. The synchronization of the plex can be a long-running operation, depending on the size of the volume and the amount of data that needs to be synchronized.

The recovery behavior in this release has changed from previous releases of VxVM. Previously, if the plex synchronization was interrupted, VxVM dissociated the plex from the volume during volume recovery. The plex synchronization needed to be restarted manually. This behavior remains if the disk group version is less than 170.

When the disk group version is 170 or 180 and the plex resynchronization is triggered by the `vxplex att`, `vxassist mirror`, `vxsnap addmir`, or `vxsnap reattach` commands, the plex remains associated with the volume during volume recovery. VxVM detects that the synchronization was interrupted, and resumes the synchronization. If the volume has an associated DCO (version 20 or later), VxVM tracks the changes while the plex is synchronizing, If the synchronization fails due to a system crash or `vxconfigd` failure, VxVM resumes the synchronization from the point where it failed. The synchronization occurs in the background, so the volume is available without delay. If the volume does not have an associated DCO, but has a version 170 or later disk group, the synchronization restarts from the beginning.

When you create a volume and add a mirror in one operation (`vxassist make nmirror=2`), the synchronization of the mirror is not tracked for automatic recovery. To ensure that the synchronization resumes from the point where it failed, create the volume first and then add the mirror with the `vxassist mirror` command.

VxVM automatically recovers volumes in some cases. If you need to recover the volumes manually, the `vxrecover` command triggers the synchronization for any plexes that have a failed synchronization process. These plexes have a state of TEMP, TEMPRM, or TEMPRMSD.

In the CVM environment, if a master node crashes while a plex synchronization is in progress, the new master restarts the synchronization from the point where the master node failed, after the master recovers. The disk group must be version 170 or later. If the disk group is version 170 but no DCO is attached, the synchronization restarts from the beginning.

You can abort a plex attach operation or synchronization operation with Ctrl-C or the `vxtask abort` command. In this case, VxVM dissociates the plex from the volume.

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.

Warning: 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

An uncorrectable I/O error occurs when disk failure, cabling or other problems cause the data on a disk to become unavailable. For a RAID-5 volume, this means that a subdisk becomes unavailable. 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/VSET/COKSTATE  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   NVOLLAYR LENGTH   [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-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
pl  r5vol-02  r5vol    ENABLED   LOG      1440     CONCAT     -          RW
sd  disk04-01 r5vol-02disk04     0        1440     0          c2t12d0    ENA
pl  r5vol-03  r5vol    ENABLED   LOG      1440     CONCAT     -          RW
sd  disk05-01 r5vol-03disk05     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.

Warning: 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`.

In the following example, the RAID-5 log plex `r5vol-02` has failed:

```
V   NAME       RVG/VSET/COKSTATE   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   NVOLLAYR  LENGTH     [COL/]OFF    AM/NM       MODE
...
v   r5vol      -         ENABLED   ACTIVE    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      ENA
sd   disk03-01   r5vol-01disk03       0          102400      2/0           c2t11d0      ENA
pl   r5vol-02    r5vol    DISABLED    BADLOG     1440        CONCAT        -            RW
sd   disk04-01   r5vol-02disk04       0          1440        0             c2t12d0      ENA
pl   r5vol-03    r5vol    ENABLED     LOG        1440        CONCAT        -            RW
sd   disk05-01   r5vol-12disk05       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:

- 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 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.

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

- Any existing log plexes are zeroed and enabled. If all logs fail during this process, the start process is aborted.
- 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.

Recovery of RAID-5 volumes

The following types of recovery may be required for RAID-5 volumes:

- Resynchronization of parity
- Reattachment of a failed RAID-5 log plex
- Recovery of a stale subdisk

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.

See “Unstartable RAID-5 volumes” on page 43.

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 only be possible to recover the volume by removing the volume, recreating it on hardware that is functioning correctly, and restoring the contents of the volume from a backup.

Resynchronizing parity on a RAID-5 volume

In most cases, a RAID-5 volume 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.

The following example is output from the `vxprint -ht` command for a stale RAID-5 volume:

```
V   NAME       RVG/VSET/COKSTATE 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  NVOLLAYR  LENGTH   [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.

Parity is regenerated by issuing `VOL_R5_RESYNC ioctls` 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 may shut down, or the system may crashe 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 checkpt=`*`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.

To resynchronize parity on a RAID-5 volume

◆ Type the following command:

```
# vxvol -g diskgroup resync r5vol
```

Reattaching a failed RAID-5 log plex

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 a failed RAID-5 log plex

- Type the following command:

```
# vxplex -g diskgroup att r5vol r5vol-plex
```

Recovering a stale subdisk in a RAID-5 volume

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 by using the `vxvol recover` command.

To recover a stale subdisk in the RAID-5 volume

- Type the following command:

```
# vxvol -g diskgroup recover r5vol subdisk
```

  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:

```
# vxvol -g diskgroup 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 a volume becoming unusable, the `vxsd` command 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.

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

Unstartable 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.

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

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

- The RAID-5 plex is 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:

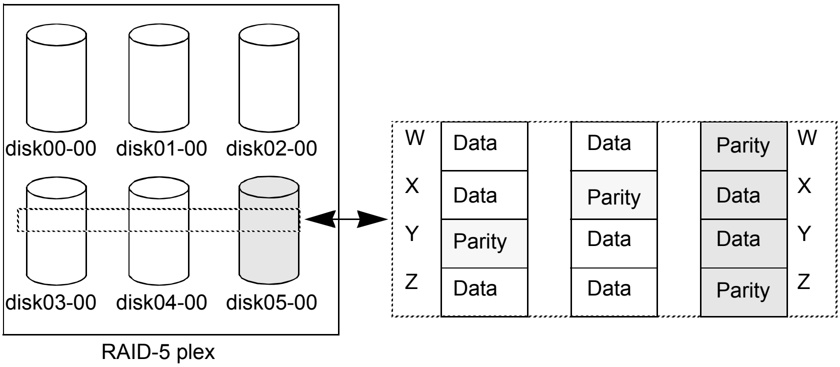
```
VxVM vxvol ERROR V-5-1-1236 Volume r5vol is not startable; RAID-5 plex
does not map entire volume length.
```

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.

Figure 3-3 illustrates a RAID-5 volume that has become invalid due to stale parity and a failed subdisk.

Figure 3-3 Invalid RAID-5 volume



There are 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 V-5-1-1237 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 37.

Forcibly starting a RAID-5 volume with stale subdisks

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 volume whenever possible.

To forcibly start a RAID-5 volume with stale subdisks

- Specify the -f option to the vxvol start command.

```
# vxvol [-g diskgroup] -f start r5vol
```

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. You can mark individual subdisks as non-stale by using the following command:

```
# vxmend [-g diskgroup] 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 if the volume were 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.

Warning: 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 the recovery of any subdisk fails, and if 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.

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.

Recovering from an incomplete disk group move

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. However, automatic recovery may not be possible if, for example, one of the disk groups has been imported on another host.

To recover from an incomplete disk group move

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 V-5-1-2907 diskgroup: Disk group does not exist
   ```

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

3. 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
   ```

4. 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 failure of a DCO volume

The procedure to recover from the failure of a data change object (DCO) volume depends on the DCO version number.

For information about DCO versioning, see the *Veritas Storage Foundation Administrator's Guide*.

Persistent FastResync uses a DCO volume to perform tracking of changed regions in a volume. If an error occurs while reading or writing a DCO volume, it is detached and the badlog flag is set on the DCO. All further writes to the volume are not tracked by the DCO.

The following sample output from the vxprint command shows a complete volume with a detached DCO volume (the TUTIL0 and PUTIL0 fields are omitted for clarity):

```
TY NAME             ASSOC          KSTATE         LENGTH      PLOFFS     STATE ...
dg mydg             mydg           -              -           -          -
dm mydg01           c4t50d0s2      -              35521408    -          -
dm mydg02           c4t51d0s2      -              35521408    -          -
dm mydg03           c4t52d0s2      -              35521408    -          FAILING
dm mydg04           c4t53d0s2      -              35521408    -          FAILING
dm mydg05           c4t54d0s2      -              35521408    -          -

v  SNAP-vol1        fsgen          ENABLED        204800      -          ACTIVE
pl vol1-03          SNAP-vol1      ENABLED        204800      -          ACTIVE
sd mydg05-01        vol1-03        ENABLED        204800      0          -
dc SNAP-vol1_dco    SNAP-vol1      -              -           -          -
v  SNAP-vol1_dcl    gen            ENABLED        144         -          ACTIVE
pl vol1_dcl-03      SNAP-vol1_dcl  ENABLED        144         -          ACTIVE
sd mydg05-02        vol1_dcl-03    ENABLED        144         0          -
sp vol1_snp         SNAP-vol1      -              -           -          -

v  vol1             fsgen         ENABLED        204800      -          ACTIVE
pl vol1-01          vol1          ENABLED        204800      -          ACTIVE
sd mydg01-01        vol1-01       ENABLED        204800      0          -
pl vol1-02          vol1          ENABLED        204800      -          ACTIVE
sd mydg02-01        vol1-01       ENABLED        204800      0          -
dc vol1_dco         vol1          -              -           -          BADLOG
v  vol1_dcl         gen           DETACHED       144         -          DETACH
pl vol1_dcl-01      vol1_dcl      ENABLED        144         -          ACTIVE
sd mydg03-01        vol1_dcl-01   ENABLED        144         0          -
pl vol1_dcl-02      vol1_dcl      DETACHED       144         -          IOFAIL
sd mydg04-01        vol1_dcl-02   ENABLED        144         0          RELOCATE
sp SNAP-vol1_snp    vol1          -              -           -          -
```

This output shows the mirrored volume, vol1, its snapshot volume, SNAP-vol1, and their respective DCOs, vol1_dco and SNAP-vol1_dco. The two disks, mydg03 and mydg04, that hold the DCO plexes for the DCO volume, vol1_dcl, of vol1 have failed. As a result, the DCO volume, vol1_dcl, of the volume, vol1, has been detached and the state of vol1_dco has been set to BADLOG. For future reference, note the entries for the snap objects, vol1_snp and SNAP-vol1_snp, that point to vol1 and SNAP-vol1 respectively.

You can use such output to deduce the name of a volume’s DCO (in this example, `vol1_dco`), or you can use the following `vxprint` command to display the name of a volume’s DCO:

```
# vxprint [-g diskgroup] -F%dco_name volume
```

You can use the `vxprint` command to check if the `badlog` flag is set for the DCO of a volume as shown here:

```
# vxprint [-g diskgroup] -F%badlog dco_name
```

This command returns the value `on` if the `badlog` flag is set. For the example output, the command would take this form:

```
# vxprint -g mydg -F%badlog vol1_dco
on
```

Use the following command to verify the version number of the DCO:

```
# vxprint [-g diskgroup] -F%version dco_name
```

This returns a value of 0 or 20. For the example output, the command would take this form:

```
# vxprint -g mydg -F%version vol1_dco
```

The DCO version number determines the recovery procedure that you should use.

See “Recovering a version 0 DCO volume” on page 48.

See “Recovering an instant snap DCO volume (version 20 or later)” on page 51.

Recovering a version 0 DCO volume

To recover a version 0 DCO volume

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

   ```
   # vxdco [-g diskgroup] -o force enable dco_name
   ```

   For the example output, the command would take this form:

   ```
   # vxdco -g mydg -o force enable vol1_dco
   ```

   The entry for `vol1_dco` in the output from `vxprint` now looks like this:

   ```
      dc vol1_dco        vol1     -     -    -  -
   ```

3 Restart the DCO volume using the following command:

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

For the example output, the command would take this form:

```
# vxvol -g mydg start vol1_dcl
```

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.

Warning: 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.

For the example output, the command would take this form:

```
# vxassist -g mydg snapclear vol1 SNAP-vol1_snp
```

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.

For the example output, the commands would take this form if `SNAP-vol1` had been moved to the disk group, `snapdg`:

```
# vxassist -g mydg snapclear vol1 SNAP-vol1_snp
# vxassist -g snapdg snapclear SNAP-vol1 vol1_snp
```

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

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

   For the example output, the command would take this form:

   ```
   # vxplex -f -g mydg snapback vol1 vol1-03
   ```

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

Recovering an instant snap DCO volume (version 20 or later)

To recover an instant snap DCO volume

1. Correct the problem that caused the I/O failure.
2. Use the `vxsnap` command to dissociate each full-sized instant snapshot volume that is associated with the volume:

   ```
   # vxsnap [-g diskgroup] dis snapvol
   ```

   For the example output, the command would take this form:

   ```
   # vxsnap -g mydg dis SNAP-vol1
   ```

3. Unprepare the volume using the following command:

   ```
   # vxsnap [-g diskgroup] unprepare volume
   ```

   For the example output, the command would take this form:

   ```
   # vxsnap -g mydg unprepare vol1
   ```

4. Start the volume using the `vxvol` command:

   ```
   # vxvol [-g diskgroup] start volume
   ```

   For the example output, the command would take this form:

   ```
   # vxvol -g mydg start vol1
   ```

5. Prepare the volume again using the following command:

   ```
   # vxsnap [-g diskgroup] prepare volume [ndcomirs=number] \
        [regionsize=size] [drl=yes|no|sequential] \
        [storage_attribute ...]
   ```

   For the example output, the command might take this form:

   ```
   # vxsnap -g mydg prepare vol1 ndcomirs=2 drl=yes
   ```

   This adds a DCO volume with 2 plexes, and also enables DRL and FastResync (if licensed).

   For full details of how to use the `vxsnap prepare` command, see the *Veritas Storage Foundation Administrator's Guide* and the `vxsnap`(1M) manual page.

Chapter 4

Recovering from instant snapshot failure

This chapter includes the following topics:

Recovering from the failure of vxsnap prepare

If a `vxsnap prepare` operation fails prematurely, the `vxprint` command may show the new DCO volume in the INSTSNAPTMP state. VxVM can usually recover the DCO volume without intervention. However, in certain situations, this recovery may not succeed. If this happens, the DCO volume must be deleted.

To recover from the failure of the vxsnap prepare command

- Type the following command:

  ```
  # vxedit [-g diskgroup] rm DCO_volume
  ```

  Alternatively, the DCO volume is removed automatically when the system is next restarted. When the DCO volume has been removed, run the `vxsnap prepare` command again.

Recovering from the failure of vxsnap make for full-sized instant snapshots

If a `vxsnap make` operation fails during the creation of a full-sized instant snapshot, the snapshot volume may go into the DISABLED state, be marked invalid, and be rendered unstartable. You can use the following command to verify that the `inst_invalid` flag is set to `on`:

```
# vxprint [-g diskgroup] -F%inst_invalid snapshot_volume
```

VxVM usually recovers the snapshot volume without intervention. However, in certain situations, this recovery may not succeed. If this happens, the DCO volume must be deleted.

To recover from the failure of the vxsnap make command for full-sized instant snapshots

1. Clear the snapshot volume's `tutil0` field. Enter the following command:

   ```
   # vxmend [-g diskgroup] clear tutil0 snapshot_volume
   ```

2. Unprepare the snapshot volume. Enter the following command:

   ```
   # vxsnap [-g diskgroup] unprepare snapshot_volume
   ```

3. If the snapshot volume is in DISABLED state, start the snapshot volume. Enter the following command:

   ```
   # vxvol [-g diskgroup] start snapshot_volume
   ```

4 Prepare the snapshot volume again for snapshot operations. Enter the following command:

```
# vxsnap [-g diskgroup] prepare snapshot_volume
```

5 Clear the volume's `tutil0` field (if it is set). Enter the following command.

```
# vxmend [-g diskgroup] clear tutil0 original_volume
```

Recovering from the failure of vxsnap make for break-off instant snapshots

If a `vxsnap make` operation fails during the creation of a third-mirror break-off instant snapshot, the snapshot volume may go into the INSTSNAPTMP state. VxVM can usually recover the snapshot volume without intervention. However, in certain situations, this recovery may not succeed. If this happens, the snapshot volume must be deleted.

To recover from the failure of the vxsnap make command for break-off instant snapshots

◆ Type the following command:

```
# vxedit [-g diskgroup] rm snapshot_volume
```

Alternatively, the snapshot volume is removed automatically when the system is next restarted.

Recovering from the failure of vxsnap make for space-optimized instant snapshots

If a `vxsnap make` operation fails during the creation of a space-optimized instant snapshot, the snapshot volume may go into the INSTSNAPTMP state. VxVM can usually recover the snapshot volume without intervention. However, in certain situations, this recovery may not succeed. If this happens, the snapshot volume must be deleted.

To recover from the failure of the vxsnap make command for space-optimized instant snapshots

- Type the following command:

  ```
  # vxedit [-g diskgroup] rm snapshot_volume
  ```

  Alternatively, the snapshot volume is removed automatically when the system is next restarted.

  If the `vxsnap make` operation was being performed on a prepared cache object by specifying the `cache` attribute, the cache object remains intact after deleting the snapshot. If the `cachesize` attribute was used to specify a new cache object, the cache object does not exist after deleting the snapshot.

Recovering from the failure of vxsnap restore

If a `vxsnap restore` operation fails, the volume being restored may go into the DISABLED state.

To recover from the failure of the vxsnap restore command

- Type the following command:

  ```
  # vxvol [-g diskgroup] start volume
  ```

Recovering from the failure of vxsnap refresh

If a `vxsnap refresh` operation fails, VxVM attempts to restart the resynchronization when the volume is recovered. The volume will be started with the synchronization in the background.

This procedure requires that the disk group version is 170 or later.

The volume being refreshed may go into the DISABLED state, be marked invalid and be rendered unstartable.

To recover from the failure of the vxsnap refresh commands

1. Use the following command to check that the `inst_invalid` flag is set to `on`:

   ```
   # vxprint [-g diskgroup] -F%inst_invalid volume
   ```

2. Use the `vxmend` command to clear the volume's `tutil0` field:

   ```
   # vxmend [-g diskgroup] clear tutil0 volume
   ```

3 Use the following command to start the volume:

```
# vxvol [-g diskgroup] start volume
```

4 Re-run the failed `refresh` command.

This results in a full resynchronization of the volume. Alternatively, remove the snapshot volume and recreate it if required.

Recovering from copy-on-write failure

If an error is encountered while performing an internal resynchronization of a volume's snapshot, the snapshot volume goes into the INVALID state, and is made inaccessible for I/O and instant snapshot operations.

To recover from copy-on-write failure

1 Use the `vxsnap` command to dissociate the volume from the snapshot hierarchy:

```
# vxsnap [-g diskgroup] dis snapshot_volume
```

2 Unprepare the volume using the following command:

```
# vxsnap [-g diskgroup] unprepare snapshot_volume
```

3 Prepare the volume using the following command:

```
# vxsnap [-g diskgroup] prepare volume [ndcomirs=number] \
 [regionsize=size] [drl=yes|no|sequential] \
 [storage_attribute ...]
```

The volume can now be used again for snapshot operations.

Alternatively, you can remove the snapshot volume and recreate it if required.

Recovering from I/O errors during resynchronization

Snapshot resynchronization (started by `vxsnap syncstart`, or by specifying `sync=on` to `vxsnap`) stops if an I/O error occurs, and displays the following message on the system console:

```
VxVM vxsnap ERROR V-5-1-6840 Synchronization of the volume
volume stopped due to I/O error
```

After correcting the source of the error, restart the resynchronization operation.

To recover from I/O errors during resynchronization

- Type the following command:

  ```
  # vxsnap [-b] [-g diskgroup] syncstart volume
  ```

Recovering from I/O failure on a DCO volume

If an I/O failure occurs on a DCO volume, its FastResync maps and DRL log cannot be accessed, and the DCO volume is marked with the BADLOG flag. DRL logging and recovery, and instant snapshot operations are not possible with the volume until you recover its DCO volume.

If the I/O failure also affects the data volume, it must be recovered before its DCO volume can be recovered.

See “Recovering an instant snap DCO volume (version 20 or later)” on page 51.

Recovering from failure of vxsnap upgrade of instant snap data change objects (DCOs)

When you upgrade instant snap data change objects (DCOs), the DCO and the DCO volume is upgraded in place. This operation fails if there is not enough space in the DCO to hold the new, upgraded DCO.

If the operation fails for lack of space, use one of the following methods:

Use the -f option to the `vxsnap upgrade` command. When -f is specified, VxVM uses any available space in the disk group for the new, upgraded DCO. Therefore, the upgrade operation can succeed if there is enough available space in the disk group.

```
# vxsnap [-g diskgroup] -f upgrade
[volume1|volset1][volume2|volset2...]
```

Use the `alloc` attribute to specify storage to be used for the new DCO. VxVM creates the new DCO on the specified storage. See the `vxsnap`(m1) manual page for information about storage attributes.

```
# vxsnap -g diskgroup upgrade
[volume1|volset1][volume2|volset2...]
[alloc=storage_attributes]
```

If both the `alloc` attribute and the `-f` option are specified, VxVM uses the storage specified to the `alloc` attribute.

Recovering from boot disk failure

This chapter includes the following topics:

VxVM and boot disk failure

Veritas Volume Manager (VxVM) protects systems from disk and other hardware failures and helps you to recover from such events. Recovery procedures 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.

The procedures for recovering volumes and their data on boot disks differ from the procedures that are used for non-boot disks.

See “About recovery from hardware failure” on page 25.

Note: The examples assume that the boot (root) disk is configured on the device `c0t0d0s2`. Your system may be configured to use a different device.

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 are some of the possible configurations for the `/usr` file system:

- `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 availablility 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 boot disk group.

There are other restrictions on the configuration of `rootvol` and `usr` volumes.

See the *Veritas Storage Foundation Administrator’s Guide*.

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. This procedure differs between Solaris SPARC systems and Solaris x64 systems.

Booting from an alternate (mirror) boot disk on Solaris x64 systems

On a Solaris x64 system, the alternate boot disk is added to the GRUB boot menu when a boot disk is mirrored. If one root disk fails, the system stays up and lets you replace the disk. No reboot is required to perform this maintenance with internal SAS controllers and other CRU-type drives that are hot swappable. Replace the disk, then rescan with the `vxdctl enable` command to discover the replacement.

Alternatively, the `bootpath` can be redefined in the EEPROM without changing the GRUB configuration.

See “The boot process on x64 systems ” on page 68.

Console access and the ability to select from the GRUB menu is required for the following procedure. The system should not have rebooted because of plex failure, but may have rebooted for other reasons.

To boot from an alternate boot disk on a Solaris x64 system

1 Select the "Alternate" GRUB menu entry:

```
title Solaris 10 11/06 s10x_u3wos_10 x64  <VxVM: Alternate Boot Disk>
       root (hd0,0,a)
       kernel /platform/i64pc/multiboot
       module /platform/i64pc/boot_archive.alt
```

2 After the system has booted, replace the failed drive.

See "Replacement of boot disks" on page 80.

Booting from an alternate boot disk on Solaris SPARC systems

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.

On a Solaris SPARC system, booting from an alternate boot disk requires that some EEPROM settings are changed.

See "The boot process on Solaris SPARC systems" on page 67.

Booting from an alternate boot disk when the GRUB menu entry is missing

If the system has to be rebooted before the failed original primary drive has been replaced, you might be required to change the BIOS boot order to allow the system to get to the GRUB menu on the alternate boot disk. The drive to be booted needs to be "hd0" or the first BIOS-detectable disk. If the failed disk is still present and detected by the BIOS as "hd0," change the BIOS **boot disk priority** to select the surviving disk.

To boot from an alternate boot disk when the GRUB menu entry is missing

1 Boot into failsafe mode from the GRUB menu. This should be the only case where failsafe mode is used, as mounting and writing on one plex outside of VxVM when both plexes are attached corrupts the file system and makes both disks unusable.

Failsafe mode scans the disks to find operating system installations. The scan should only find one.

2 If the scan finds two operating system installations, you must pull the failed drive to proceed.

If the scan finds only one installation or after you pulled the failed drive, the failsafe mode discovers the good mirror and prompts you to mount that root file system on `/a`.

3 Enter **y** at the prompt.

4 List the device that corresponds to the root file system on the root disk mirror:

```
# ls -la /a/dev/dsk/altboot_disk_root_slice
```

Make a note of the PROM path for the root disk mirror that displays. For example:

```
# ls -al /a/dev/dsk/c0t3d0s0
lrwxrwxrwx 1 root root 60 Dec 13 13:17 /dev/dsk/c0t3d0s0 ->
../../devices/pci@0,0/pci1022,7450@2/pci1000,3060@3/sd@3,0:a
/pci@0,0/pci1022,7450@2/pci1000,3060@3/sd@3,0:a
```

5 Save the original `bootenv.rc` file:

```
# cp /a/boot/solaris/bootenv.rc /a/boot/solaris/bootenv.orig
```

6 Edit the `/a/boot/solaris/bootenv.rc` file.

On Solaris x64, the EEPROM storage is simulated by the `/boot/solaris/bootenv.rc` file. Set the `bootpath` in the EEPROM storage file to the path of the boot disk mirror. For example:

```
bootpath=/pci@0,0/pci1022,7450@2/pci1000,3060@3/sd@3,0:a
```

7 Update boot archive

```
# bootadm update-archive -R /a
```

Any changes to the `/a/boot` directory must be updated with this command, otherwise booting may fail.

8 Unmount /a:

```
# unmount /a
```

9 Reboot the system with init 6:

```
# init 6
```

The system finds GRUB and boot the remaining mirror plex.

10 After the system has booted, replace the failed drive and sync the mirror plex.

See the *Veritas Storage Foundation and High Availability Solutions Troubleshooting Guide*.

11 Restore the original bootenv.rc file:

```
# cp /boot/solaris/bootenv.orig /boot/solaris/bootenv.rc
```

12 Configure the system for recovery from a Primary boot drive failure.

See "Manual GRUB procedure to set up an alternate boot disk" on page 64.

Manual GRUB procedure to set up an alternate boot disk

You can set up the root disk mirror to boot from the primary and alternate boot disk. If one root disk fails, the system stays up and allows you to replace the disk. No reboot is required to perform this maintenance with internal SAS controllers and other CRU-type drives that are hot swappable. Replace the disk, then rescan with the vxdctl enable command to discover the replacement.

To set up an alternate boot disk using the manual GRUB procedure

1 List the device that corresponds to the root file system on the root disk mirror:

```
# ls -la /dev/dsk/altboot_disk_root_slice
```

Make a note of the PROM path for the root disk mirror that is displayed. For example:

```
# ls -al /dev/dsk/c0t3d0s0
lrwxrwxrwx 1 root root 60 Dec 13 13:17 /dev/dsk/c0t3d0s0 ->
../../devices/pci@0,0/pci1022,7450@2/pci1000,3060@3/sd@3,0:a
```

In this example, the PROM path is:

```
/pci@0,0/pci1022,7450@2/pci1000,3060@3/sd@3,0:a
```

2 Set the alternate boot path in the EEPROM storage to the path of the boot disk mirror.

On Solaris x64, the EEPROM storage is simulated by the `/boot/solaris/bootenv.rc` file. You can update the EEPROM storage by either manually editing the `bootenv.rc` file or by issuing the **eeprom** command. For example:

```
# eeprom altbootpath="/pci@0,0/pci1022,7450@2/pci1000,3060@3/sd@3,0:a
```

The EEPROM storage is now configured to boot from the primary boot disk:

```
bootpath=primary_boot_disk_path
altbootpath=alternate_boot_disk_path
```

3 Save the primary EEPROM file settings.

```
# cp /boot/solaris/bootenv.rc /boot/solaris/bootenv.pri
```

4 Edit the `/boot/solaris/bootenv.rc` to create an alternate EEPROM storage file.

Swap the device entries for `bootpath` and `altbootpath`:

```
bootpath=alternate_boot_disk_path
altbootpath=primary_boot_disk_path
```

The EEPROM storage is now configured to boot from the mirror boot disk.

5 Save the alternate EEPROM file settings:

```
# cp /boot/solaris/bootenv.rc /boot/solaris/bootenv.alt
```

6 Create the alternate boot archive:

```
# bootadm update-archive
```

Any changes to the `/boot` directory must be updated with this command, otherwise booting may fail.

7 Save the alternate boot archive:

```
# cp /platform/i64pc/boot_archive /platform/i64pc/boot_archive.alt
```

8 Restore the primary boot archive and EEPROM file:

```
# cp /boot/solaris/bootenv.pri /boot/solaris/bootenv.rc
```

9 Create the primary boot archive:

```
# bootadm update-archive
```

Any changes to the `/boot` directory must be updated with this command, otherwise booting may fail.

10 Locate the active GRUB menu in the `/boot/grub/menu.lst` file:

```
# bootadm list-menu
```

Typically the default GRUB menu entry is the entry that is automatically selected when the system boots up. **Default 0** refers to the first menu entry.

11 Edit the /boot/grub/menu.lst and duplicate the active menu entry. For example:

```
title Solaris 10 11/06 s10x_u3wos_10 X64  <Primary>
       root (hd0,0,a)
       kernel /platform/i64pc/multiboot
       module /platform/i64pc/boot_archive
```

Or:

```
title Solaris 10 11/06 s10x_u3wos_10 X64  <Primary>
       kernel /platform/i64pc/multiboot
       module /platform/i64pc/boot_archive
```

If the GRUB root statement is missing from the menu entry, root defaults to root (hd0,0,a).

12 Edit the duplicated entry and change boot_archive to boot_archive.alt. Add Alternate to the title. For example:

```
title Solaris 10 11/06 s10x_u3wos_10 X64  <Alternate>
       root (hd0,0,a)
       kernel /platform/i64pc/multiboot
       module /platform/i64pc/boot_archive.alt
```

The GRUB menu title displays when the system is booting. You can select the alternate menu entry if the primary boot drive fails but was not replaced.

The boot process on Solaris SPARC systems

A Solaris 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]
```

The 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.

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.

The filename argument 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.)

Warning: Do not boot a system running VxVM with rootability enabled using all the defaults presented by the -a flag.

See "Restoring a copy of the system configuration file" on page 74.

Boot flags are not interpreted by the boot program. The boot program passes all boot-flags to the file identified by filename.

See the kadb (1M) manual page.

See the kernel (1) manual page.

The boot process on x64 systems

From Update 1 of the Solaris 10 OS, x64 systems are configured to use the GRUB boot loader. The devices from which a system may be booted are defined in the GRUB configuration file, /boot/grub/menu.lst. From the GRUB menu, you can select from the available bootable partitions that are known to the system. By default, the system will boot from the device that is defined by the bootpath variable in the EEPROM.

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 boot 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.

Unrelocation of 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.

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.

You can use the dumpadm command to view and set the dump device.

See the dumpadm(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 are some possible failure modes:

- The boot device cannot be opened.
- The system cannot be be booted from unusable or stale plexes.
- A UNIX partition is invalid.
- There are incorrect entries in /etc/vfstab.
- Configuration files are missing or damaged.

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
```

The following are common causes for the system PROM being unable to read the boot program from the boot drive:

- 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 ERROR V-5-1-1049: 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.

See “Booting from alternate boot disks” on page 61.

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. 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 a display such as this:

```
DEVICE    TYPE     DISK      GROUP     STATUS
-         -       rootdisk  bootdg    failed was: c0t3d0s2
c0t1d0s2 sliced   disk01    bootdg    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.

See “Re-adding a failed boot disk ” on page 81.

However, if the reattach fails, then the disk needs to be replaced.

See “Replacing a failed boot disk” on page 82.

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.prevm`.

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 on booting the system:

```
INIT: Cannot create /var/adm/utmp or /var/adm/utmpx
INIT: failed write of utmpx entry:" "
```

It is recommended that you first run fsck on the root partition as shown in this example:

```
# fsck -F ufs /dev/rdsk/c0t0d0s0
```

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.

To repair a damaged /usr entry in /etc/vfstab

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/rdsk/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 configuration files

VxVM no longer maintains entries for tunables in /etc/system as was the case for VxVM 3.2 and earlier releases. All entries for Veritas Volume Manager device driver tunables are now contained in files named /kernel/drv/vx*.conf, such as /kernel/drv/vxio.conf.

See the *Veritas Storage Foundation Administrator's Guide*.

Warning: If you need to modify configuration files such as /etc/system, make a copy of the file in the root file system before editing it.

Restoring a copy of the system configuration file

If the system configuration file is damaged, the system can be booted if the file is restored from a saved copy.

To restore a copy of the system configuration file

1 Boot the system with the following command:

```
ok boot -a
```

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

- 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
```

- 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
```

- 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
  ```

Restoring /etc/system if a copy is not available on the root disk

If /etc/system is damaged or missing, and a saved copy of this 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 configuration files

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:

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

Lines of the form *`forceload: drv/driver`* are used to forcibly load the drivers 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,isptwo@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 configuration files were restored.

Repair of 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).

The following 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. This procedure 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 a backup

This procedure assumes that you have the following resources available:

- 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.
- 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 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`.

To recover a root disk and root mirror from a backup

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.

   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. For example, to make a ufs file system on the root partition, enter:

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

   See the mkfs(1M) manual page.

   See the mkfs_ufs(1M) manual page.

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.

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

7. If the /usr file system is separate from the root file system, 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 If the `/usr` file system is separate from the root file system, 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.

10 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
```

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

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

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

These lines should then read:

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

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

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

with this line:

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

15 Remove `/a/dev/vx/dsk/bootdg` and `/a/dev/vx/rdsk/bootdg`:

```
# rm /a/dev/vx/dsk/bootdg
# rm /a/dev/vx/rdsk/bootdg
```

16 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.

17 If there are only root disk mirrors in the old boot disk group, remove any volumes that were associated with the encapsulated root disk (for example, `rootvol`, `swapvol` and `usrvol`) from the `/dev/vx/dsk/bootdg` and `/dev/vx/rdsk/bootdg` directories.

18 If there are other disks in the old boot disk group that are not used as root disk mirrors, remove files involved with the installation that are no longer needed:

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

Start the Veritas Volume Manager I/O daemons:

```
# vxiod set 10
```

Start the Veritas Volume Manager configuration daemon in disabled mode:

```
# vxconfigd -m disable
```

Initialize the `volboot` file:

```
# vxdctl init
```

Enable the old boot disk group excluding the root disk that VxVM interprets as failed::

```
#  vxdctl enable
```

Use the `vxedit` command (or the Veritas Enterprise Administrator (VEA)) to remove the old root disk volumes and the root disk itself from Veritas Volume Manager control.

19 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.

Replacement of 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.

  See "Re-adding a failed boot disk " on page 81.
- If the disk has failed completely, replace it.

  See "Replacing a failed boot disk" on page 82.

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.

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 utility with this command:

```
# vxdisk list
```

vxdisk displays this (example) list:

```
DEVICE       TYPE      DISK      GROUP      STATUS
c0t0d0s2     sliced    -         -          error
c0t1d0s2     sliced    disk02    bootdg     online
-            -         disk01    bootdg     failed was:c0t0d0s2
```

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
c0t0d0s2     sliced    disk01    bootdg      online
c0t1d0s2     sliced    disk02    bootdg      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.

You can then perform any special procedures to correct the problem, such as reformatting the device.

To re-add a failed boot disk

1. Select the `vxdiskadm` “Replace a failed or removed disk” menu item to replace the disk, and specify the same device as the replacement. For example, you might replace `disk01` with the device `c0t0d0s2`.
2. 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 for a failed boot 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 [-g diskgroup] -st -e 'sd_disk="
diskname
"'
```

where diskname is the name of the disk that failed or of one of its mirrors.

The following is sample output from running this command:

```
# vxprint -g rtdg -st -e 'sd_disk="rtdg01"'
Disk group: rtdg

SD NAME        PLEX         DISK    DISKOFFS    LENGTH    [COL/]OFF   DEVICE MODE
...
sd rtdg01-01   swapvol-01   rtdg01  0           1045296   0           c0t0d0 ENA
sd rtdg01-02   rootvol-01   rtdg01  1045296     16751952  0           c0t0d0 ENA
```

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. In this example, the `DISKOFFS` and `LENGTH` values for the subdisk `rtdg01-02` are 1,045,296 and 16,751,952, so the disk size is (1,045,296 + 16,751,952)/2, which equals 8,898,624 kilobytes or approximately 8.5 gigabytes.

Note: Disk sizes reported by manufacturers usually represent the unformatted capacity of disks. Also, most manufacturers use the terms megabyte and gigabyte to mean a million (1,000,000) and a billion (1,000,000,000) bytes respectively, rather than the usual meaning of 1,048,576 and1,073,741,824 bytes.

To replace a failed boot disk:

1. Boot the system from an alternate boot disk.

   See "Booting from alternate boot disks" on page 61.

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.

3. Shut down the system and replace the failed hardware.

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

   Warning: If the replacement disk was previously an encapsulated root disk under VxVM control, select to reorganize the disk when prompted. Otherwise, the disk is left with an invalid VTOC that renders the system unbootable. Ensure that you have made at least one valid copy of any existing data on the disk.

   ---

5. Use `vxdiskadm` to mirror the alternate boot disk to the replacement boot disk.

6. When the volumes on the boot disk have been restored, shut down the system, and test that the system can be booted from the replacement boot 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 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.

Warning: You should assume that reinstallation can potentially destroy the contents of any disks that are touched by the reinstallation process

---

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, 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.

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, the following steps are required:

- Replace any failed disks or other hardware, and detach any disks not involved in the reinstallation.
  See "Prepare the system for reinstallation" on page 85.

- Reinstall the base system and any other unrelated Volume Manager packages.
  See "Reinstall the operating system" on page 85.
- Add the Volume Manager package, but do not execute the `vxinstall` command.
  See "Reinstalling Veritas Volume Manager" on page 86.
- Recover the Veritas Volume Manager configuration.
  See "Recovering the Veritas Volume Manager configuration " on page 86.
- Restore any information in volumes affected by the failure or reinstallation, and recreate system volumes (`rootvol`, `swapvol`, `usr`, and other system volumes).
  See "Cleaning up the system configuration" on page 87.

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.

Reinstalling Veritas Volume Manager

To reinstall Veritas Volume Manager

1 Reinstall the Veritas software from the installation disks.

See the *Installation Guide*.

Warning: Do not use `vxinstall` to initialize VxVM.

---

2 If required, use the `vxlicinst` command to install the Veritas Volume Manager license key.

See the `vxlicinst`(1) manual page.

Recovering 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.

To recover the Veritas Volume Manager configuration

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 which 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.

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.

Cleaning up the system configuration

After reinstalling VxVM, you must clean up the system configuration.

To clean up the system configuration

1. 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 following volumes must be removed:

`rootvol`	Contains the `root` file system.
`swapvol`	Contains the swap area.
`standvol` (if present)	Contains the `stand` file system.
`usr` (if present)	contains the `usr` file system.

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

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

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

2 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 reinstalled or removed disks, these mirrors must be removed. The mirrors can be re-added later.

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   mydg     online
c0t2d0s2  sliced   disk03   mydg     online
-         -        disk01   mydg     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.

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

```
  # vxprint [-g diskgroup] -sF “%vname” -e’sd_disk = “
disk
”’
```

where disk is the access 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.

The following is sample output from running this command:

```
  # vxprint -g mydg -sF “%vname” -e’sd_disk = "disk01"'
v01
```

4 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      RVG/VSET/COKSTATE      STATE     LENGTH   READPOL     PREFPLEXUTYPE
PL   NAME      VOLUME      KSTATE     STATE     LENGTH   LAYOUT      NCOL/WID MODE
SD   NAME      PLEX        DISK       DISKOFFS  LENGTH   [COL/]OFF   DEVICE   MODE

v    v01       -           DISABLED   ACTIVE    24000    SELECT      -        fsgen
pl   v01-01    v01         DISABLED   NODEVICE  24000    CONCAT      -        RW
sd   disk01-06  v01-01     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.

5. 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
   ```

6. 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     RVG/VSET/COKSTATE     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

v    v02      -           DISABLED  ACTIVE    30720      SELECT      v02-01     fsgen
pl   v02-01   v02         DISABLED  NODEVICE  30720      STRIPE      3/128      RW
sd   disk02-02v02-01      disk01    424144    10240      0/0         c1t5d2     ENA
sd   disk01-05v02-01      disk01    620544    10240      1/0         c1t5d3     DIS
sd   disk03-01v02-01      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
   ```

7 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       RVG/VSET/COKSTATE   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

v    v03        -          DISABLED ACTIVE   0720      SELECT     -        fsgen
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
```

8 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 a disk, use the `vxdg` command. For example, to remove the failed disk `disk01`, use the following command:

```
# vxdg rmdisk disk01
```

If the `vxdg` command returns an error message, invalid mirrors exist.

Repeat step 2 through step 7 until all invalid volumes and mirrors are removed.

9 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.

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

11 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.

12 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.

13 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.

14 Start up hot-relocation, if required, by either rebooting the system or manually start the relocation watch daemon, `vxrelocd` (this also starts the `vxnotify` process).

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

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
```

See the *Veritas Storage Foundation Administrator's Guide*.

See the `vxrelocd`(1M) manual page.

Chapter 6

Managing commands and transactions

This chapter includes the following topics:

Command logs

The `vxcmdlog` command allows you to log the invocation of other Veritas Volume Manager (VxVM) commands to a file.

The following examples demonstrate the usage of `vxcmdlog`:

`vxcmdlog -l`	List current settings for command logging.
`vxcmdlog -m on`	Turn on command logging.
`vxcmdlog -s 512k`	Set the maximum command log file size to 512K.
`vxcmdlog -n 10`	Set the maximum number of historic command log files to 10.
`vxcmdlog -n no_limit`	Remove any limit on the number of historic command log files.
`vxcmdlog -m off`	Turn off command logging.

By default command logging is turned on. Command lines are logged to the file `cmdlog`, in the directory `/etc/vx/log`. This path name is a symbolic link to a directory whose location depends on the operating system. If required, you can redefine the directory which is linked.

If you want to preserve the settings of the `vxcmdlog` utility, you must also copy the settings file, `.cmdlog`, to the new directory.

Warning: The `.cmdlog` file is a binary and should not be edited.

The size of the command log is checked after an entry has been written so the actual size may be slightly larger than that specified. When the log reaches a maximum size, the current command log file, `cmdlog`, is renamed as the next available historic log file, *`cmdlog.number`*, where *`number`* is an integer from 1 up to the maximum number of historic log files that is currently defined, and a new current log file is created.

A limited number of historic log files is preserved to avoid filling up the file system. If the maximum number of historic log files has been reached, the oldest historic log file is removed, and the current log file is renamed as that file.

Each log file contains a header that records the host name, host ID, and the date and time that the log was created.

The following are sample entries from a command log file:

```
# 0, 2329, Wed Feb 12 21:19:31 2003
    /usr/sbin/vxdctl mode
# 17051, 2635, Wed Feb 12 21:19:33 2003
    /usr/sbin/vxdisk -q -o alldgs list
# 0, 2722, Wed Feb 12 21:19:34 2003
    /etc/vx/diag.d/vxprivutil dumpconfig /dev/vx/rdmp/Disk_4s2
# 26924, 3001, Thu Feb 13 19:30:57 2003
    /usr/sbin/vxdisk list Disk_1
```

Each entry usually contains a client ID that identifies the command connection to the `vxconfigd` daemon, the process ID of the command that is running, a time stamp, and the command line including any arguments.

If the client ID is 0, as in the third entry shown here, this means that the command did not open a connection to `vxconfigd`.

The client ID is the same as that recorded for the corresponding transactions in the transactions log.

See “Transaction logs” on page 97.

See “Association of command and transaction logs” on page 99.

Most command scripts are not logged, but the command binaries that they call are logged. Exceptions are the `vxdisksetup`, `vxinstall`, and `vxdiskunsetup` scripts, which are logged.

If there is an error reading from the settings file, command logging switches to its built-in default settings. This may mean, for example, that logging remains enabled after being disabled using `vxcmdlog -m off` command. If this happens, use the `vxcmdlog` utility to recreate the settings file, or restore the file from a backup.

See the `vxcmdlog`(1M) manual page.

Transaction logs

The `vxtranslog` command allows you to log VxVM transactions to a file.

The following examples demonstrate the usage of `vxtranslog`:

`vxtranslog -l`	List current settings for transaction logging.
`vxtranslog -m on`	Turn on transaction logging.
`vxtranslog -s 512k`	Set the maximum transaction log file size to 512K.
`vxtranslog -n 10`	Set the maximum number of historic transaction log files to 10.
`vxtranslog -n no_limit`	Remove any limit on the number of historic transaction log files.
`vxtranslog -q on`	Turn on query logging.
`vxtranslog -q off`	Turn off query logging.
`vxtranslog -m off`	Turn off transaction logging.

By default, transaction logging is turned on. Transactions are logged to the file `translog`, in the directory `/etc/vx/log`. This path name is a symbolic link to a directory whose location depends on the operating system. If required, you can redefine the directory which is linked. If you want to preserve the settings of the `vxtranslog` utility, you must also copy the settings file, `.translog`, to the new directory.

Warning: The `.translog` file is a binary and should not be edited.

The size of the transaction log is checked after an entry has been written so the actual size may be slightly larger than that specified. When the log reaches a maximum size, the current transaction log file, `translog`, is renamed as the next available historic log file, *`translog.number`*, where *`number`* is an integer from 1 up to the maximum number of historic log files that is currently defined, and a new current log file is created.

A limited number of historic log files is preserved to avoid filling up the file system. If the maximum number of historic log files has been reached, the oldest historic log file is removed, and the current log file is renamed as that file.

Each log file contains a header that records the host name, host ID, and the date and time that the log was created.

The following are sample entries from a transaction log file:

```
Thu Feb 13 19:30:57 2003
Clid = 26924, PID = 3001, Part = 0, Status = 0, Abort Reason = 0
    DA_GET SENA0_1
    DISK_GET_ATTRS SENA0_1
    DISK_DISK_OP SENA0_1 8
    DEVNO_GET SENA0_1
    DANAME_GET 0x1d801d8 0x1d801a0
    GET_ARRAYNAME SENA 50800200000e78b8
    CTLR_PTOLNAME /pci@1f,4000/pci@5/SUNW,qlc@4/fp@0,0
    GET_ARRAYNAME SENA 50800200000e78b8
    CTLR_PTOLNAME /pci@1f,4000/pci@5/SUNW,qlc@5/fp@0,0
    DISCONNECT <no request data>
```

The first line of each log entry is the time stamp of the transaction. The `Clid` field corresponds to the client ID for the connection that the command opened to `vxconfigd`. The `PID` field shows the process ID of the utility that is requesting the operation. The `Status` and `Abort Reason` fields contain error codes if the transaction does not complete normally. The remainder of the record shows the data that was used in processing the transaction.

The client ID is the same as that recorded for the corresponding command line in the command log.

See "Command logs" on page 95.

See "Association of command and transaction logs" on page 99.

If there is an error reading from the settings file, transaction logging switches to its built-in default settings. This may mean, for example, that logging remains enabled after being disabled using `vxtranslog -m off` command. If this happens, use the vxtranslog utility to recreate the settings file, or restore the file from a backup.

Association of command and transaction logs

The Client and process IDs that are recorded for every request and command assist you in correlating entries in the command and transaction logs. To find out which command issued a particular request in transaction log, use a command such as the following to search for the process ID and the client ID in the command log:

```
# egrep -n PID cmdlog | egrep Clid
```

In this example, the following request was recorded in the transaction log:

```
Wed Feb 12 21:19:36 2003
Clid = 8309, PID = 2778, Part = 0, Status = 0, Abort Reason = 0
    DG_IMPORT foodg
    DG_IMPORT foodg
    DISCONNECT <no request data>
```

To locate the utility that issued this request, the command would be:

```
# egrep -n 2778 cmdlog | egrep 8309
7310:# 8309, 2778, Wed Feb 12 21:19:36 2003
```

The output from the example shows a match at line 7310 in the command log. Examining lines 7310 and 7311 in the command log indicates that the vxdg import command was run on the foodg disk group:

```
# sed -e '7310,7311!d' cmdlog
# 8309, 2778, Wed Feb 12 21:19:36 2003 7311
/usr/sbin/vxdg -m import foodg
```

If there are multiple matches for the combination of the client and process ID, you can determine the correct match by examining the time stamp.

If a utility opens a conditional connection to vxconfigd, its client ID is shown as zero in the command log, and as a non-zero value in the transaction log. You can use the process ID and time stamp to relate the log entries in such cases.

Associating CVM commands issued from slave to master node

When you run commands on the CVM slave node that change the shared disk group configuration, CVM ships the commands to the CVM master node for execution.

For example, on the slave node, you run the following command, to create a volume in a shared disk group. CVM ships the command to the master node, and CVM executes the command on the master node.

```
# vxassist -g shareddg make shared-vol1 200M
```

On the CVM slave node, enter the following command to identify the shipped command from the transaction log (translog):

```
# egrep CMDSHIP_REQUEST translog
```

In this example, the following entry was recorded in the transaction log on slave node:

```
Thu Jul 15 06:30:16 2010
Clid = 5302, PID =  589906, Part = 0, Status = 0, Abort Reason = 0
        DG_SET_CURRENT_ID  shareddg
        DG_SET_CURRENT  shareddg
        DG_GETCFG_ID  0xdde49f shareddg
        DG_GETCFG_NAME  0xdde49f shareddg
        DG_SET_CURRENT_ID  shareddg
        DG_SET_CURRENT  shareddg
        DG_SET_CURRENT_ID  shareddg
        DG_SET_CURRENT  shareddg
        DG_GETCFG_ALL  0x420
        DG_GETCFG_ALL  0x420
        VOL_TRANS   ds4700-0_7 ds4700-0_3 ds4700-0_
        DG_GET_DEFAULT  <no request data>
        CMDSHIP_REQUEST  Command Shipped = /usr/sbin/vxassist -g

shareddg make shared-vol1 200M
 Default dg = nodg
        DROPPED  <no request data>
```

To locate the utility that issued this request on slave node, enter the following command:

```
# egrep -n PID cmdlog | egrep Clid
```

In this example, enter the following command:

```
# egrep -n 589906 cmdlog | egrep 5302
```

```
7310#: 5302, 589906, Thu Jul 15 06:30:14 2010 /usr/sbin/vxassist -g
```

The output from the example shows a match at line 7310 in the command log. Examining lines 7310 and 7311 in the command log indicates that the `vxassist make` command was run on the shareddg disk group:

```
# sed -e '7310,7311!d' cmdlog
```

```
# 5302, 589906, Thu Jul 15 06:30:14 2010
/usr/sbin/vxassist -g shareddg   make  shared-vol1  200M
```

To determine the cluster monitor nodeid (CM nid) of the slave node, enter the following command:

```
# /etc/vx/bin/vxclustadm nidmap
```

If the command uses disk access (DA) names, the shipped command converts the DA names to unique disk IDs (UDID) or Disk media (DM) names. On the CVM master node, the vxconfigd log shows the entry for the received command. To determine the commands received from slave nodes on the master, enter the command:

```
# egrep CMDSHIP_REQUEST /var/adm/messages
```

Note: The file to which the `vxconfigd` messages are logged may differ, depending on where the messages are redirected.

In this example, the following received command would be recorded in the vxconfigd log on master node:

```
07/15 06:29:02: V-5-1-0 receive_cmdship_message:
CMDSHIP_REQUEST: Received command:
Text - /usr/sbin/vxassist -g shareddg make shared-vol1 200M len = 53
CLID = 5302 SlaveID = 0 Defaultdg = nodg
```

The CVM master node executes the received command and sends the response to the slave node.

To find the response that the master node sent to the slave node, enter a command such as the following on the master node:

```
# egrep CMDSHIP_RESPONSE translog | egrep Slave-Clid
```

In this example, enter the following command to find the response that the master node sent:

```
# egrep CMDSHIP_RESPONSE translog | egrep 5302
Thu Jul 15 06:29:03 2010
Clid = 27741, PID = 475212, Part = 0, Status = 0, Abort Reason = 0
 CMDSHIP_RESPONSE  SlaveCLID = 5302 SlaveCMID = 0
 ExitCode = 12 Flags = 1 stdoutlen = 0 stderrlen = 98  Response =

VxVM vxassist ERROR V-5-1-10127 creating volume shared-vol1:

Record already exists in disk group
DROPPED  <no request data>
```

Command completion is not enabled

If the **Tab** key does not automatic complete the command, check the following:

- The command must be in the list of supported commands.
  For the list of supported commands, see the product release notes.
- The shell must be bash version 2.4 or later.

If the above requirements are met, the bash completion entry may have been removed from the bashrc or bash_profile.

To enable the command completion

- Run the following command:

  ```
  # . /etc/bash_completion.d/vx_bash
  ```

Chapter 7

Backing up and restoring disk group configurations

This chapter includes the following topics:

About disk group configuration backup

Disk group configuration backup and restoration allows you to backup and restore all configuration data for Veritas Volume Manager (VxVM) disk groups, and for VxVM objects such as volumes that are configured within the disk groups. Using this feature, you can recover from corruption of a disk group's configuration that is stored as metadata in the private region of a VM disk. After the disk group configuration has been restored, and the volume enabled, the user data in the public region is available again without the need to restore this from backup media.

Warning: The backup and restore utilities act only on VxVM configuration data. They do not back up or restore any user or application data that is contained within volumes or other VxVM objects. If you use `vxdiskunsetup` and `vxdisksetup` on a disk, and specify attributes that differ from those in the configuration backup, this may corrupt the public region and any data that it contains.

The `vxconfigbackupd` daemon monitors changes to the VxVM configuration and automatically records any configuration changes that occur, probably after an hour. Two utilities, `vxconfigbackup` and `vxconfigrestore`, are provided for backing up and restoring a VxVM configuration for a disk group.

When importing a disk group, any of the following errors in `the vxconfigd` log indicates that the disk group configuration and/or disk private region headers have become corrupted:

```
VxVM vxconfigd ERROR V-5-1-569 Disk group group,Disk disk:Cannot
auto-import group: reason
```

The reason for the error is usually one of the following:

```
Configuration records are inconsistent
Disk group has no valid configuration copies
Duplicate record in configuration
Errors in some configuration copies
Format error in configuration copy
Invalid block number
Invalid magic number
```

If VxVM cannot update a disk group's configuration because of disk errors, it disables the disk group and displays the following error:

```
VxVM vxconfigd ERROR V-5-1-123 Disk group group: Disabled by errors
```

If such errors occur, you can restore the disk group configuration from a backup after you have corrected any underlying problem such as failed or disconnected hardware.

Configuration data from a backup allows you to reinstall the private region headers of VxVM disks in a disk group whose headers have become damaged, to recreate a corrupted disk group configuration, or to recreate a disk group and the VxVM objects within it. You can also use the configuration data to recreate a disk group on another system if the original system is not available.

Note: Restoration of a disk group configuration requires that the same physical disks are used as were configured in the disk group when the backup was taken.

See "Backing up a disk group configuration" on page 105.

See "Restoring a disk group configuration" on page 106.

Backing up a disk group configuration

VxVM uses the disk group configuration daemon to monitor the configuration of disk groups, and creates a backup of the configuration after an hour when ever it is changed. This is done to avoid processing multiple backup's being triggered in a short interval.. By default, the five most recent backups are preserved. The backup's are placed at the following location: `/etc/vx/cbr/bk/diskgroup.dgid/bkp_YYYYMMDDD_HHMMSS/`.This format helps you in restoring the diskgroup from a particular backup based on the time at which it was taken. If required, you can also back up a disk group configuration by running the `vxconfigbackup` command.

The following files record disk group configuration information:

/etc/vx/cbr/bk/diskgroup.dgid/dgid.dginfo	Disk group information.
/etc/vx/cbr/bk/diskgroup.dgid/dgid .diskinfo	Disk attributes.
/etc/vx/cbr/bk/diskgroup.dgid/dgid .binconfig	Binary configuration copy.
/etc/vx/cbr/bk/diskgroup.dgid/dgid.cfgrec	Configuration records in `vxprint -m` format.

Here *diskgroup* is the name of the disk group, and *dgid* is the disk group ID. If a disk group is to be recreated on another system, copy these files to that system.

Warning: Take care that you do not overwrite any files on the target system that are used by a disk group on that system.

To back up a disk group configuration

- Type the following command:

  ```
  # /etc/vx/bin/vxconfigbackup [-f] [-l directory] [[diskgroup ...] |
   [dgid ...]]
  ```

`diskgroup(s)`	The diskgroup(s) can be specified either by name or by ID
`-f`	The `-f` option allows you to force a complete backup
`-l`	The `-l`option allows you to specify a directory for the location of the backup configuration files other than the default location, `/etc/vx/cbr/bk`

  To back up all disk groups, use this version of the command:

  ```
  # /etc/vx/bin/vxconfigbackup [-f] [-l directory]
  ```

  See the `vxconfigbackup`(1M) manual page.

Restoring a disk group configuration

You can use the `vxconfigrestore` utility to restore or recreate a disk group from its configuration backup. The restoration process consists of a precommit operation followed by a commit operation. At the precommit stage, you can examine the configuration of the disk group that would be restored from the backup. The actual disk group configuration is not permanently restored until you choose to commit the changes.

Warning: None of the disks or VxVM objects in the disk groups should be open or in use by any application while the restoration is being performed.

You can choose whether or not any corrupted disk headers are to be reinstalled at the precommit stage. If any of the disks' private region headers are invalid, restoration may not be possible without reinstalling the headers for the affected disks.

See the `vxconfigrestore`(1M) manual page.

To perform the precommit operation

◆ Use the following command to perform a precommit analysis of the state of the disk group configuration, and to reinstall the disk headers where these have become corrupted:

```
# /etc/vx/bin/vxconfigrestore -p [-l directory] \
  {diskgroup | dgid}
```

The disk group can be specified either by name or by ID.

The `-l` option allows you to specify a directory for the location of the backup configuration files other than the default location, `/etc/vx/cbr/bk`.

See "Backing up a disk group configuration" on page 105.

To specify that the disk headers are not to be reinstalled

◆ Type the following command:

```
# /etc/vx/bin/vxconfigrestore -n [-l directory] \
  {diskgroup | dgid}
```

At the precommit stage, you can use the `vxprint` command to examine the configuration that the restored disk group will have. You can choose to proceed to commit the changes and restore the disk group configuration. Alternatively, you can cancel the restoration before any permanent changes have been made.

To abandon restoration at the precommit stage

◆ Type the following command:

```
# /etc/vx/bin/vxconfigrestore -d [-l directory] \
  {diskgroup | dgid}
```

To perform the commit operation

- To commit the changes that are required to restore the disk group configuration, use the following command:

  ```
  # /etc/vx/bin/vxconfigrestore -c [-l directory] \
    {diskgroup | dgid}
  ```

  Note: Between the precommit and commit state, any operation that results in change in diskgroup configuration should not be attempted. This may lead to unexpected behavior. User should either abandon the restoration or commit the operation.

  If no disk headers are reinstalled, the configuration copies in the disks' private regions are updated from the latest binary copy of the configuration that was saved for the disk group.

  If any of the disk headers are reinstalled, a saved copy of the disks' attributes is used to recreate their private and public regions. These disks are also assigned new disk IDs. The VxVM objects within the disk group are then recreated using the backup configuration records for the disk group. This process also has the effect of creating new configuration copies in the disk group.

  Volumes are synchronized in the background. For large volume configurations, it may take some time to perform the synchronization. You can use the `vxtask -l list` command to monitor the progress of this operation.

  Disks that are in use or whose layout has been changed are excluded from the restoration process.

If the back-up is taken of a shared disk group, the `vxconfigrestore` command restores it as a private disk group. After the disk group is restored, run the following commands to make the disk group shared.

To make the disk group shared

1 Deport the disk group:

```
# vxdg deport dg_name
```

2 Import the disk group as shared:

```
# vxdg -s import dg_name
```

Resolving conflicting backups for a disk group

In some circumstances where disks have been replaced on a system, there may exist several conflicting backups for a disk group. In this case, you see a message similar to the following from the `vxconfigrestore` command:

```
VxVM vxconfigrestore ERROR V-5-1-6012 There are two backups that
have the same diskgroup name with different diskgroup id :
1047336696.19.xxx.veritas.com

1049135264.31.xxx.veritas.com
```

The solution is to specify the disk group by its ID rather than by its name to perform the restoration. The backup file, */etc/vx/cbr/bk/diskgroup.dgid/dgid.dginfo*, contains a timestamp that records when the backup was taken.

The following is a sample extract from such a backup file that shows the timestamp and disk group ID information:

```
TIMESTAMP
Tue Apr 15 23:27:01 PDT 2003
.
.
.
DISK_GROUP_CONFIGURATION
Group:     mydg
dgid: 1047336696.19.xxx.veritas.com
.
.
.
```

Use the timestamp information to decide which backup contains the relevant information, and use the `vxconfigrestore` command to restore the configuration by specifying the disk group ID instead of the disk group name.

Chapter 8

Troubleshooting issues with importing disk groups

This chapter includes the following topics:

Clearing the udid_mismatch flag for non-clone disks

After you install or upgrade a new Veritas Volume Manager (VxVM) package or an Array Support Library (ASL) package, changes to an ASL may cause the udid_mismatch flag or the clone_disk flag to be set on disks that are not cloned disks. VxVM uses these flags to indicate hardware snapshots or copies of a LUN.

If the disk is not a cloned disk, this behavior could result in errors during disk group import, depending on the import flags. When a disk group is imported, VxVM may skip cloned disks. Also, the import may fail because a disk group has a combination of cloned and non-cloned disks.

If the disk is not a cloned disk, you must manually clear the udid_mismatch flag on the disk before the disk group import can succeed.

Note: In a cluster, perform all the steps on the same node.

To clear the udid_mismatch flag from a disk

1 Retrieve the refreshed list of disks showing the udid_mismatch flag or the clone_disk flag. Use one of the following methods:

- Run the following commands:

```
# vxdisk scandisks
# vxdisk list | egrep "udid_mismatch|clone_disk"
```

- Or, run the following command:

  ```
  # vxdisk -o alldgs list | egrep "udid_mismatch|clone_disk"
  ```

2 If the disks are part of an imported disk group, deport the disk group.

```
# vxdg deport dgname
```

3 Clear the udid_mismatch flag on all non-clone disks identified during step 1.

```
# vxdisk updateudid diskname
```

4 Clear the clone flag on non-clone disks.

```
# vxdisk set diskname clone=off
```

5 Import the disk group if it was deported during step 2

```
# vxdg [-s] import dgname
```

Recovering from CDS errors

This chapter includes the following topics:

- CDS error codes and recovery actions

CDS error codes and recovery actions

Table 9-1 lists the CDS error codes and the action that is required.

Table 9-1 Error codes and required actions

Error number	Message	Action
329	Cannot join a non-CDS disk group and a CDS disk group	Change the non-CDS disk group into a CDS disk group (or vice versa), then retry the join operation.
330	Disk group is for a different platform	Import the disk group on the correct platform. It cannot be imported on this platform.
331	Volume has a log which is not CDS compatible	To get a log which is CDS compatible, you need to stop the volume, if currently active, then start the volume. After the volume has been successfully started, retry setting the CDS attribute for the disk group.
332	License has expired, or is not available for CDS	Obtain a license from Symantec that enables the usage of CDS disk groups.

Table 9-1 Error codes and required actions *(continued)*

Error number	Message	Action
333	Non-CDS disk cannot be placed in a CDS disk group	Do one of the following: ■ Add the disk to another disk group that is a non-CDS disk group. ■ Re-initialize the disk as a CDS disk so that it can be added to the CDS disk group. ■ Change the CDS disk group into a non-CDS disk group and then add the disk.
334	Disk group alignment not CDS compatible	Change the alignment of the disk group to 8K and then retry setting the CDS attribute for the disk group.
335	Sub-disk length violates disk group alignment	Ensure that sub-disk length value is a multiple of 8K.
336	Sub-disk offset violates disk group alignment	Ensure that sub-disk offset value is a multiple of 8K.
337	Sub-disk plex offset violates disk group alignment	Ensure that sub-disk plex offset value is a multiple of 8K.
338	Plex stripe width violates disk group alignment	Ensure that plex stripe width value is a multiple of 8K.
339	Volume or log length violates disk group alignment	Ensure that thelength of the volume is a multiple of 8K. For a log, set the value of the `dgalign_checking` attribute to `round`. This ensures that the length of the log is silently rounded to a valid value.
340	Last disk media offset violates disk group alignment	Reassociate the DM record prior to upgrading.

Table 9-1 Error codes and required actions *(continued)*

Error number	Message	Action
341	Too many device nodes in disk group	Increase the number of device nodes allowed in the disk group, if not already at the maximum. Otherwise, you need to remove volumes from the disk group, possibly by splitting the disk group.
342	Map length too large for current log length	Use a smaller map length for the DRL/DCM log, or increase the log length and retry.
343	Volume log map alignment violates disk group alignment	Remove the DRL/DCM log, then add it back after changing the alignment of the disk group.
344	DIsk device cannot be used as a CDS disk	
345	Disk group contains an old-style RVG which cannot be imported on this platform	Import the disk group on the platform that created the RVG. To import the disk group on this platform, first remove the RVG on the creating platform.
346	Cache object autogrow by max_autogrow violates disk group alignment	Ensure that cache attribute value is a multiple of 8K.
347	User transactions are disabled for the disk group	Retry the command as it was temporarily disallowed by the `vxcdsconvert` command executing at the same time.
348	Disk is in use	Contact Technical Support.

Chapter 10

Error messages

This chapter includes the following topics:

About error messages

Informational, failure, and other error messages may be displayed on the console by the Veritas Volume Manager (VxVM) configuration daemon (`vxconfigd`), the VxVM kernel driver, `vxio`, and the various VxVM commands. These messages may indicate errors that are infrequently encountered and difficult to troubleshoot.

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

You may find it useful to consult the VxVM command and transaction logs to understand the context in which an error occurred.

See "Command logs" on page 95.

How error messages are logged

VxVM provides the option of logging debug messages 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).

If enabled, the default debug log file is `/var/vxvm/vxconfigd.log`.

To enable logging of debug output to the default debug log file, edit the startup script for `vxconfigd`.

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.

Note: syslog logging is enabled by default. Debug message logging is disabled by default.

If syslog output is enabled, messages with a priority higher than Debug are written to /var/log/syslog.

See "Configuring logging in the startup script" on page 118.

Alternatively, you can use the following command to change the debug level:

```
# vxdctl debug level [pathname]
```

There are 10 possible levels of debug logging with the values 0 through 9. Level 1 provides the least detail, and 9 the most. Level 0 turns off logging. If a path name is specified, this file is used to record the debug output instead of the default debug log file. If the vxdctl debug command is used, the new debug logging level and debug log file remain in effect until the VxVM configuration daemon, vxconfigd, is next restarted.

See the vxdctl(1M) manual page.

See the vxconfigd(1M) manual page.

Configuring logging in the startup script

To enable log file or syslog logging on a permanent basis, you can edit the /lib/svc/method/vxvm-sysboot (in Solaris 10) script that starts the VxVM configuration daemon, vxconfigd.

To configure logging in the startup script

1. 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
   ```

   The `opts="$opts -x syslog"` string is usually uncommented so that `vxconfigd` uses `syslog` logging by default. Inserting a `#` character at the beginning of the line turns off `syslog` logging for `vxconfigd`.

   If you do not specify a debug level, only Error, Fatal Error, Warning, and Notice messages are logged. Debug messages are not logged.

   By default, vxconfigd is started at boot time with the `-x syslog` option. This redirects `vxconfigd` console messages to `syslog`. If you want to retain this behavior when restarting `vxconfigd` from the command line, include the `-x syslog` argument, as restarting `vxconfigd` does not preserve the option settings with which it was previously running. Similarly, any Veritas Volume Manager operations that require `vxconfigd` to be restarted may not retain the behavior that was previously specified by option settings.

2. After making changes to the way `vxconfigd` is invoked in the startup file, run the following command on a Solaris 10 system to notify that the service configuration has been changed:

   ```
   # svcadm refresh vxvm/vxvm-sysboot
   ```

Types of 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 describe error message numbers and the types of error message that may be seen, and provide a list of the more common errors, a detailed description of the likely cause of the problem together with suggestions for any actions that can be taken.

Messages have the following generic format:

```
product component severity message_number message_text
```

For Veritas Volume Manager, the product is set to `VxVM`. The component can be the name of a kernel module or driver such as `vxdmp`, a configuration daemon such as `vxconfigd`, or a command such as `vxassist`.

Note: For full information about saving system crash information, see the *Solaris System Administation Guide*.

Messages are divided into the following types of severity in decreasing order of impact on the system:

PANIC

A panic is a severe event as it halts a system during its normal operation. A panic message from the kernel module or from a device driver indicates a hardware problem or software inconsistency 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. The following is an example of such a message:

```
VxVM vxio PANIC V-5-0-239 Object association
depth overflow
```

FATAL ERROR

A fatal error message from a configuration daemon, such as `vxconfigd`, indicates a severe problem with the operation of VxVM that prevents it from running. The following is an example of such a message:

```
VxVM vxconfigd FATAL ERROR V-5-0-591 Disk group
bootdg: Inconsistency -- Not loaded into kernel
```

ERROR

An error message from a command indicates that the requested operation cannot be performed correctly. The following is an example of such a message:

```
VxVM vxassist ERROR V-5-1-5150 Insufficient
number of active snapshot mirrors in
snapshot_volume .
```

WARNING

A warning message from the kernel indicates that a non-critical operation has failed, possibly because some resource is not available or the operation is not possible. The following is an example of such a message:

```
VxVM vxio WARNING V-5-0-55 Cannot find device
number for boot_path
```

NOTICE

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. The following is an example of such a message:

```
VxVM vxio NOTICE V-5-0-252 read error on object
subdisk of mirror plex in volume volume (start
offset, length length) corrected.
```

INFO

An informational message does not indicate an error, and requires no action.

The unique message number consists of an alpha-numeric string that begins with the letter “V”. For example, in the message number, `V-5-1-3141`, “V” indicates that this is a Veritas product error message, the first numeric field (5) encodes the product (in this case, VxVM), the second field (1) represents information about the product component, and the third field (3141) is the message index. The text of the error message follows the message number.

Messages

This section contains a list of messages that you may encounter during the operation of Veritas Volume Manager. However, the list is not exhaustive and the second field may contain the name of different command, driver or module from that shown here.

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

If you encounter a product error message, record the unique message number preceding the text of the message. Search on the message number at the following URL to find the information about that message:

http://sort.symantec.com/

When contacting Veritas Technical Support, either by telephone or by visiting the Veritas Technical Support website, be sure to provide the relevant message number. Veritas Technical Support will use this message number to quickly determine if there are TechNotes or other information available for you.

Chapter 11

Troubleshooting Veritas Volume Replicator

This chapter includes the following topics:

Recovery from RLINK connect problems

This section describes the errors that may be encountered when connecting RLINKs. To be able to troubleshoot RLINK connect problems, it is important to understand the RLINK connection process.

Connecting the Primary and Secondary RLINKs is a two-step operation. The first step, which attaches the RLINK, is performed by issuing the `vradmin startrep` command. The second step, which connects the RLINKs, is performed by the kernels on the Primary and Secondary hosts.

When the `vradmin startrep` command is issued, VVR performs a number of checks to ensure that the operation is likely to succeed, and if it does, the command changes the state of the RLINKs from DETACHED/STALE to ENABLED/ACTIVE. The command then returns success.

If the command is successful, the kernel on the Primary is notified that the RLINK is enabled and it begins to send messages to the Secondary requesting it to connect. Under normal circumstances, the Secondary receives this message and connects. The state of the RLINKs then changes from ENABLED/ACTIVE to CONNECT/ACTIVE.

If the RLINK does not change to the CONNECT/ACTIVE state within a short time, there is a problem preventing the connection. This section describes a number of possible causes. An error message indicating the problem may be displayed on the console.

- If the following error displays on the console:

  ```
  VxVM VVR vxrlink INFO V-5-1-5298 Unable to establish connection
   with remote host <remote_host>, retrying
  ```

  Make sure that the `vradmind` daemon is running on the Primary and the Secondary hosts; otherwise, start the `vradmind` daemon by issuing the following command:

  ```
  # /etc/init.d/vras-vradmind.sh start
  ```

  For an RLINK in a shared disk group, make sure that the virtual IP address of the RLINK is enabled on the logowner.

- If there is no self-explanatory error message, issue the following command on both the Primary and Secondary hosts:

  ```
  # vxprint -g diskgroup -l rlink_name
  ```

  In the output, check the following:
  The *remote_host* of each host is the same as *local_host* of the other host.
  The *remote_dg* of each host is the same as the disk group of the RVG on the other host.
  The *remote_dg_dgid* of each host is the same as the *dgid* (disk group ID) of the RVG on the other host as displayed in the output of the *vxprint -l diskgroup* command.
  The *remote_rlink* of each host is the same as the name of the corresponding RLINK on the other host.
  The *remote_rlink_rid* of each host is the same as the *rid* of the corresponding RLINK on the other host.
  Make sure that the network is working as expected. Network problems might affect VVR, such as prevention of RLINKs from connecting or low performance. Possible problems could be high latency, low bandwidth, high collision counts, and excessive dropped packets.

- For an RLINK in a private disk group, issue the following command on each host.
  For an RLINK in a shared disk group, issue the following command on the logowner on the Primary and Secondary:

  ```
  # ping -s remote_host
  ```

There should be no packet loss or very little packet loss. To ensure that the network can transmit large packets, issue the following command on each host for an RLINK in a private disk group.
For an RLINK in a shared disk group, issue the following command on the logowner on the Primary and Secondary:

```
# ping -I 2 remote_host 8192
```

The packet loss should be about the same as for the earlier ping command.

- Issue the `vxiod` command on each host to ensure that there are active I/O daemons. If the output is `0 volume I/O daemons running`, activate I/O daemons by issuing the following command:

  ```
  # vxiod set 10
  ```

- VVR uses well-known ports to establish communications with other hosts. Issue the following command to display the port number:

  ```
  # vxprint -g diskgroup -l rlink_name
  ```

  Issue the following command to ensure that the heartbeat port number in the output matches the port displayed by `vxprint` command:

  ```
  # vrport
  ```

  Confirm that the state of the heartbeat port is `Idle` by issuing the following command:

  ```
  # netstat -an -P udp
  ```

  The output looks similar to this:

  ```
  UDP: IPv4
      Local Address         Remote Address        State
      --------------------  --------------------  -------
      *.port-number                                  Idle
  ```

- Check for VVR ports on the Primary and Secondary sites.
  Run the `vrport` utility and verify that ports are same at both ends.
  Check whether the required VVR ports are open. Check for UDP 4145, TCP 4145, TCP 8199, and the anonymous port. Enter the following commands:

```
# netstat -an -P udp | grep 4145
*.4145                         Idle
*.4145                                                     Idle
```

```
# netstat -an -P tcp | grep 4145
*.4145          *.*                 0      0 49152      0 LISTEN
*.4145                         *.*                             0      0 49152      0 LISTEN
# netstat -an -P tcp | grep 8199
*.8199          *.*                 0      0 49152      0 LISTEN
10.180.162.41.32990  10.180.162.42.8199   49640      0 49640      0 ESTABLISHED
*.8199                         *.*                             0      0 49152      0 LISTEN
```

Perform a `telnet` test to check for open ports. For example, to determine if port 4145 is open, enter the following:

```
# telnet <remote> 4145
```

- Use the `netstat` command to check if `vradmind` daemons can connect between the Primary site and the Secondary site.

```
# netstat -an -P tcp | grep 8199 | grep ESTABLISHED
 10.180.162.41.32990  10.180.162.42.8199   49640      0 49640      0 ESTABLISHED
```

If there is no established connection, check if the `/etc/hosts` file has entries for the Primary and Secondary sites. Add all participating system names and IP addresses to the `/etc/hosts` files on each system or add the information to the name server database of your name service.

- On Solaris 11, you must manually edit the `/etc/hosts` file to remove the hostname from the lines for loopback addresses.
  For example:

```
::1 seattle localhost

127.0.0.1 seattle loghost localhost
```

needs to be changed to:

```
::1 localhost

127.0.0.1 loghost localhost

129.148.174.232 seattle
```

Recovery from configuration errors

Configuration errors occur when the configuration of the Primary and Secondary RVGs is not identical. Each data volume in the Primary RVG must have a corresponding data volume in the Secondary RVG of exactly the same size; otherwise, replication will not proceed. If a volume set is associated to the RDS, the configuration of the volume set must also match on the Primary and on the Secondary.

Errors in configuration are detected in two ways:

- When an RLINK is attached for the first time, the configuration of the Secondary is checked for configuration errors. If any errors are found, the `attach` command fails and prints error messages indicating the problem. The problem is fixed by correcting the configuration error, and then retrying the attach.
- Changes that affect the configuration on the Primary or Secondary may cause the Secondary to enter the PAUSE state with the `secondary_config_err` flag set. The problem is fixed by correcting the configuration error, and then resuming the RLINK.

Errors during an RLINK attach

During an RLINK attach, VVR checks for errors in the configuration of data volumes. VVR also checks for errors in the configuration of volume sets, if the RDS has a volume set associated to the RVG.

Data volume errors during an RLINK attach

When an RLINK is attached, VVR checks whether for each data volume associated to the Primary RVG, the Secondary RVG has an associated data volume of the same size that is mapped to its counterpart on the Primary. The following example illustrates an attempted attach with every possible problem and how to fix it. Before the `attach`, the Primary has this configuration:

```
TY     Name               Assoc        KSTATE     LENGTH       STATE

rv     hr_rvg         -                DISABLED     -          EMPTY

rl     rlk_london_hr_rvg  hr_rvg       DETACHED     -          STALE

v      hr_dv01            hr_rvg       ENABLED    12800        ACTIVE

pl     hr_dv01-01         hr_dv01      ENABLED    12800        ACTIVE

sd     disk01-05          hr_dv01-01   ENABLED    12800          -

v      hr_dv02            hr_rvg       ENABLED    12800        ACTIVE
```

```
pl     hr_dv02-01         hr_dv02      ENABLED    12880        ACTIVE

sd     disk01-06          hr_dv02-01   ENABLED    12880

v      hr_dv03            hr_rvg       ENABLED    12880        ACTIVE

pl     hr_dv03-01         hr_dv03      ENABLED    12880        ACTIVE

sd     disk01-07          hr_dv03-01   ENABLED    12880        -

v      hr_srl             hr_rvg       ENABLED    12880        ACTIVE

pl     hr_srl-01          hr_srl       ENABLED    12880        ACTIVE

sd     disk01-08          hr_srl-01    ENABLED    12880 0      -
```

The Secondary has the following configuration:

```
TY     Name               Assoc        KSTATE     LENGTH       STATE

rv     hr_rvg             -            ENABLED    -        -   ACTIVE

rl     rlk_seattle_hr_rvg hr_rvg       ENABLED    -        -   ACTIVE

v      hr_dv01            hr_rvg       ENABLED    12700    -   ACTIVE

pl     hr_dv01-01         hr_dv01      ENABLED    13005    -   ACTIVE

sd     disk01-17          hr_dv01-01   ENABLED    13005    0    -

v      hr_dv2             hr_rvg       ENABLED    12880    -   ACTIVE

pl     hr_dv02-01         vol2         ENABLED    13005    -   ACTIVE

sd     disk01-18          hr_dv02-01   ENABLED    13005    0    -

v      hr_srl             hr_rvg       ENABLED    12880    -   ACTIVE

pl     hr_srl-01          hr_srl       ENABLED    13005    -   ACTIVE

sd     disk01-19          hr_srl-01    ENABLED    13005    0    -
```

Note that on the Secondary, the size of volume `hr_dv01` is small, `hr_dv2` is misnamed (must be `hr_dv02`), and `hr_dv03` is missing. An attempt to attach the Primary RLINK to this Secondary using the `attach` command fails.

```
# vxrlink -g hrdg -f att rlk_london_hr_rvg
```

The following messages display:

```
VxVM VVR vxrlink INFO V-5-1-3614 Secondary data volumes detected
    with rvg hr_rvg as parent:
VxVM VVR vxrlink ERROR V-5-1-0 Size of secondary datavol hr_dv01
    (len=12700) does not match size of primary (len=12800)
VxVM VVR vxrlink ERROR V-5-1-3504 primary datavol hr_dv02 is not
    mapped on secondary, yet
VxVM VVR vxrlink ERROR V-5-1-3504 primary datavol hr_dv03 is not
    mapped on secondary, yet
```

To fix the problem, issue the following commands on the Secondary:

1 Resize the data volume `hr_dv01`:

   ```
   # vradmin -g hrdg resizevol hr_rvg hr_dv01 12800
   ```

2 Rename the data volume `hr_dv2` to `hr_dv02`:

   ```
   # vxedit -g hrdg rename hr_dv2 hr_dv02
   ```

3 Associate a new volume, `hr_dv03`, of the same size as the Primary data volume hr_dv03.

   ```
   # vxassist -g hrdg make hr_dv03 12800
   # vxvol -g hrdg assoc hr_rvg hr_dv03
   ```

Alternatively, the problem can be fixed by altering the Primary to match the Secondary, or any combination of the two. When the Primary and the Secondary match, retry the attach.

On the Primary:

```
# vxrlink -g hrdg -f att rlk_london_hr_rvg
VxVM VVR vxrlink INFO V-5-1-3614 Secondary data volumes detected
     with rvg hr_rvg as parent:
VxVM VVR vxrlink INFO V-5-1-0 vol1: len=12800 primary_datavol=hr_dv01
VxVM VVR vxrlink INFO V-5-1-0 vol1: len=12800 primary_datavol=hr_dv02
VxVM VVR vxrlink INFO V-5-1-0 vol1: len=12800 primary_datavol=hr_dv03
```

Volume set errors during an RLINK attach

If a volume set is associated to an RDS, the name of the volume set on the Primary must have the same name as the volume set on the Secondary, and the volume sets must have the same configuration of component volumes.

When an RLINK is attached, VVR checks whether for each volume set associated to the Primary RVG, the Secondary RVG has an associated volume set of the same name. Also, VVR checks whether the volume sets on the Primary and on the Secondary have the same component volumes with the same names, lengths, and indices. (The volume names of the component volumes can be different on the Primary and Secondary if they are mapped, as for independent volumes.) If any of the component volumes do not exist on the Secondary or have a mismatched name, length, or index, the RLINK attach command fails with the appropriate error message.

See “Volume set configuration errors during modification of an RVG” on page 133.

If the volume set does not exist on the Secondary, but all the component volumes exist on the Secondary with the correct names and lengths, VVR creates the volume set on the Secondary and associates it to the RDS. This does not cause a configuration error.

Errors during modification of an RVG

After the initial setup and attach of a Secondary RLINK, incorrect modifications such as adding, resizing, and renaming volumes can cause configuration errors, if they result in a mismatch between the volumes on the Primary and on the Secondary. If an RVG has an associated volume set, modifications to the volume set can also cause configuration errors. These include incorrectly adding, removing, or renaming component volumes of the associated volume set; adding component volumes with different indices on the Primary and Secondary; or renaming the associated volume set.

When a modification of an RVG causes a configuration error, the affected RLINK is PAUSED with the `secondary_config_err` flag set. This prevents replication to the Secondary until the problem is corrected.

Run the *`vxrlink verify rlink`* command at either node to check whether this has occurred. When the configuration error has been corrected, the affected RLINK can be resumed.

Missing data volume error during modifcation of an RVG

If a data volume is added to the Primary RVG and the Secondary has no corresponding data volume, the RLINK state changes to PAUSED with the secondary_config_err flag set. Executing the vxrlink verify command produces the following:

On the Primary:

```
# vxrlink -g hrdg verify rlk_london_hr_rvg
RLINK              REMOTE HOST    LOCAL_HOST    STATUS    STATE
rlk_london_hr_rvg   london           seattle    ERROR     PAUSE
ERROR: hr_dv04 does not exist on secondary (london)
```

On the Secondary:

```
# vxrlink -g hrdg verify rlk_seattle_hr_rvg
RLINK                REMOTE HOST     LOCAL_HOST         STATUS
STATE
rlk_seattle_hr_rvg   seattle         london             ERROR
PAUSE
ERROR: hr_dv04 does not exist on secondary (local host)
```

To correct the problem, either create and associate hr_dv04 on the Secondary or alternatively, dissociate vol04 from the Primary, and then resume the Secondary RLINK. To resume the Secondary RLINK, use the *vradmin resumerep rvg_name* command.

If hr_dv04 on the Primary contains valid data, copy its contents to hr_dv04 on the Secondary before associating the volume to the Secondary RVG.

Data volume mismatch error during modification of an RVG

If a Primary data volume is increased in size, but the Secondary data volume is not, a configuration error results.

On the Primary:

```
# vxassist growby hr_dv04 100
# vxrlink -g hrdg verify rlk_london_hr_rvg
RLINK                REMOTE HOST    LOCAL_HOST       STATUS       STATE
rlk_london_hr_rvg    london         seattle          ERROR        PAUSE
ERROR: hr_dv04 too small (12800). Primary is 12900
```

On the Secondary:

```
# vxrlink -g hrdg verify rlk_seattle_hr_rvg
RLINK                REMOTE HOST    LOCAL_HOST       STATUS       STATE
rlk_seattle_hr_rvg   seattle        london           ERROR        PAUSE
ERROR: hr_dv04 too small (12800). Primary is 12900
```

To correct the problem, increase the size of the Secondary data volume, or shrink the Primary data volume:

```
# vradmin -g hrdg resizevol hr_rvg hr_dv04 12900
```

After resizing a data volume, resume the Secondary RLINK by issuing the following command on any host in the RDS:

```
# vradmin -g hrdg resumerep hr_rvg
```

Data volume name mismatch error during modification of an RVG

If a volume is renamed on the Primary but not on the Secondary, a configuration error results and the RLINK will be disconnected. Use the `vxprint -lP` command to view the RLINK flags. If the `secondary_config_err` flag is set, use one of the following commands to determine if there is a data volume name mismatch error.

On the Primary:

```
# vxrlink -g hrdg verify rlk_london_hr_rvg
RLINK               REMOTE HOST   LOCAL_HOST       STATUS       STATE
rlk_london_hr_rvg   london        seattle          ERROR        PAUSE
ERROR: hr_dv04 on secondary has wrong primary_datavol name (hr_dv04,
should be hr_dv05)
```

On the Secondary:

```
# vxrlink -g hrdg verify rlk_seattle_hr_rvg
RLINK               REMOTE HOST   LOCAL_HOST        STATUS       STATE
rlk_seattle_hr_rvg  seattle       london            ERROR        PAUSE
ERROR: hr_dv04 on secondary has wrong primary_datavol name (hr_dv04,
should be hr_dv05)
```

To fix this error, do one of the following:

- Rename either the Primary or Secondary data volume, and resume the RLINK using the *`vradmin resumerep rvg_name`* command.
  OR
- Set the `primary_datavol` field on the Secondary data volume to refer to the new name of the Primary data volume as follows, and resume the RLINK using the *`vradmin resumerep rvg_name`* command.
  On the Secondary:

  ```
  # vxedit -g hrdg set primary_datavol=hr_dv05 hr_dv04
  ```

where hr_dv05 is the new name on the Primary

Volume set configuration errors during modification of an RVG

If a volume set is associated to the RDS, the name of the volume set on the Secondary must have the same name as the volume set on the Primary in order for replication to occur. In addition, the volume set on the Secondary must have the same component volumes, with the same names, lengths and indices as on the Primary.

If a component volume is resized on the Primary but not on the Secondary, a data volume mismatch error results. Resize the volume and resume replication.

See "Data volume mismatch error during modification of an RVG" on page 131.

The configuration of the Secondary is checked for configuration errors, when an RLINK is attached for the first time. If any errors are found, the `vradmin startrep` command fails and prints error messages indicating the problem. Correct the configuration error, and then retry the command.

Configuration errors may also occur when you modify a volume set or its component volumes. Run the *`vxrlink verify rlink`* command at either node to check whether this has occurred. Correct the configuration error, and then resume the RLINK.

Volume set name mismatch error

If the volume set name differs on the Primary and the Secondary, the following error displays:

```
VSet name vset_name of secondary datavol vol_name does not match
VSet name vset_name of primary datavol vol_name
```

To correct the problem, rename the volume set on either the Primary or the Secondary, using the following command:

```
# vxedit -g diskgroup rename vset_name new_vset_name
```

Volume index mismatch error

If the indices for the component volumes on the Primary volume set and the Secondary volume set are different, the following error displays:

```
VSet index (index_name) of secondary datavol vol_name does not
 match VSet index (index_name) of primary datavol vol_name
```

To correct the problem, perform the following steps on the Secondary:

1. Dissociate each volume from the volume set using the following command:

   ```
   # vxvset -g diskgroup rmvol vset_name compvol_name
   ```

   When you remove the last volume, the volume set is also removed.

2. Create the volume set using the following command:

   ```
   # vxvset -g diskgroup -o index make vset_name \
     compvol_name index
   ```

3. Associate each of the remaining volumes to the volume set, specifying the index of the corresponding volumes on the Primary using the following command:

   ```
   # vxvset -g diskgroup -o index addvol vset_name  \
     compvol_name index
   ```

Component volume mismatch error

If a data volume is removed from the volume set on the Primary RVG only or added to the volume set on the Secondary RVG only, the following error displays:

```
Secondary datavol vol_name is associated to VSet vol_name
whereas primary datavol is not associated to any Vset
```

Similarly, if a data volume is removed from the volume set on the Secondary RVG only or added to the volume set on the Primary RVG only, the following error displays:

```
Primary datavol vol_name is associated to VSet whereas secondary
datavol vol_name is not associated to any Vset
```

To correct the problem, add or remove data volumes from either the Secondary or the Primary volume sets. The volume sets on the Primary and the Secondary should have the same component volumes.

To add a data volume to a volume set, do one of the following:

- To add a data volume to a volume set in an RVG:

  ```
  # vradmin -tovset vset_name addvol rvg_name vol_name
  ```

- To remove a data volume in a volume set in an RVG:

  ```
  # vradmin -fromvset vset_name delvol rvg_name vol_name
  ```

Recovery on the Primary or Secondary

This section describes how to recover from various types of disasters, such as a Primary host crash or an error on the Primary or Secondary data volumes.

About recovery from a Primary-host crash

When a Primary host recovers from a failure, VVR automatically recovers the RVG configuration. When the Primary recovers, VVR recovers the Primary SRL and all volumes in the RVG. Information about the recent activity on the SRL and the data volume is maintained in the SRL header. VVR uses this information to speed up recovery, which is automatic on reboot.

Recovering from Primary data volume error

If a write to a Primary data volume fails, the data volume is detached. The RVG continues to function as before to provide access to other volumes in the RVG. Writes to the failed volume return an error and are not logged in the SRL.

RLINKs are not affected by a data volume failure. If the SRL was not empty at the time of the volume error, those updates will continue to flow from the SRL to the Secondary RLINKs. Any writes for the failed volume that were completed by the application but not written to the volume remain in the SRL. These writes are marked as pending in the SRL and are replayed to the volume when the volume is recovered. If the volume is recovered from the backup and restarted, these writes are discarded.

If the data volume had a permanent failure, such as damaged hardware, you must recover from backup. Recovery from this failure consists of two parts:

- Restoring the Primary data volume from backup
- Resynchronizing any Secondary RLINKs

If the RVG contains a database, recovery of the failed data volume must be coordinated with the recovery requirements of the database. The details of the database recovery sequence determine what must be done to synchronize Secondary RLINKs.

Detailed examples of recovery procedures are given in the examples:

- See “Example - Recovery with detached RLINKs” on page 136.
- See “Example - Recovery with minimal repair” on page 136.
- See “Example - Recovery by migrating the primary” on page 137.

If the data volume failed due to a temporary outage such as a disconnected cable, and you are sure that there is no permanent hardware damage, you can start the data volume without dissociating it from the RVG. The pending writes in the SRL are replayed to the data volume.

See "Example - Reovery from temporary I/O error" on page 137.

Example - Recovery with detached RLINKs

In this example, all the RLINKs are detached before recovery of the failure begins on the Primary. When recovery of the failure is complete, including any database recovery procedures, all the RLINKs must be synchronized using a Primary Storage Checkpoint.

Perform the steps on the Primary. In this example, the Primary host is `seattle`.

To recover from failure

1. Detach all RLINKs

   ```
   # vxrlink -g hrdg det rlk_london_hr_rvg
   ```

2. Fix or repair the data volume.

   If the data volume can be repaired by repairing its underlying subdisks, you need not dissociate the data volume from the RVG. If the problem is fixed by dissociating the failed volume and associating a new one in its place, the dissociation and association must be done while the RVG is stopped.

3. Make sure the data volume is started before restarting the RVG.

   ```
   # vxvol -g hrdg start hr_dv01
   # vxrvg -g hrdg start hr_rvg
   ```

4. Restore the database.
5. Synchronize all the RLINKs using block-level backup and checkpointing.

Example - Recovery with minimal repair

This example does the minimum to repair data volume errors, leaving all RLINKs attached. In this example, restoring the failed volume data from backup, and the database recovery is done with live RLINKs. Because all the changes on the Primary are replicated, all the Secondaries must be consistent with the Primary after the changes have been replicated. This method may not always be practical because it might require replication of large amounts of data. The repaired data volume must also be carefully tested on every target database to be supported.

Perform the steps on the Primary. In this example, the Primary host is `seattle`.

To recover from failure

1. Stop the RVG.

   ```
   # vxrvg -g hrdg stop hr_rvg
   ```

2. Dissociate the failed data volume from the RVG.
3. Fix or repair the data volume or use a new volume.

   If the data volume can be repaired by repairing its underlying subdisks, you need not dissociate the data volume from the RVG. If the problem is fixed by dissociating the failed volume and associating a new one in its place, the dissociation and association must be done while the RVG is stopped.
4. Associate the volume with the RVG.
5. Make sure the data volume is started before restarting the RVG. If the data volume is not started, start the data volume:

   ```
   # vxvol -g hrdg start hr_dv01
   ```

6. Start the RVG:

   ```
   # vxrvg -g hrdg start hr_rvg
   ```

7. Restore the database.

Example - Recovery by migrating the primary

As an alternative recovery method, the Primary role can be transferred to a Secondary host.

After takeover, the original Primary with the failed data volume will not become acting_secondary until the failed data volume is recovered or dissociated.

Example - Reovery from temporary I/O error

If the I/O error on the data volume is temporary and you are sure that all the existing data is intact, you can start the data volume without dissociating it from the RVG. For example, if the SCSI cable was disconnected or there was a power outage of the storage. In this case, follow the steps below.

To recover from a temporary I/O error

1. Fix the temporary failure.
2. Start the data volume:

   ```
   # vxvol -g hrdg start hr_dv01
   ```

   Any outstanding writes in the SRL are written to the data volume.

Primary SRL volume error cleanup and restart

If there is an error accessing the Primary SRL, the SRL is dissociated and the RLINKs are detached. The state of the Primary and Secondary RLINKs is changed to STALE. The RVG state does not change, but the RVG is put into PASSTHRU mode that allows update of the Primary volume to continue until the error is fixed.

See "About RVG PASSTHRU mode" on page 139.

The SRL must be repaired manually and then associated with the RVG. While the SRL is being repaired, no attempt is made to send data to the RLINKs. After the SRL is replaced, all RLINKs must be completely synchronized. Attach the RLINKs and perform a complete synchronization of the Secondaries.

On the Primary (seattle):

To cleanup after a Primary SRL error

1. Dissociate the SRL from the RVG.

   ```
   # vxvol -g hrdg dis hr_srl
   ```

2. Fix or replace the SRL volume.
3. Make sure that the repaired SRL is started before associating it with the RVG. If the repaired SRL is not started, start it:

   ```
   # vxvol -g hrdg start hr_srl
   ```

4. Associate a new SRL with the RVG. After associating the new SRL, the RVG PASSTHRU mode no longer displays in the output of the command `vxprint -lV`.

   ```
   # vxvol -g hrdg aslog hr_rvg hr_srl
   ```

5. Completely synchronize the Secondary.

About RVG PASSTHRU mode

Typically, writes to data volumes associated with an RVG go to the RVG's SRL first, and then to the RLINKs and data volumes. If the Primary SRL is ever detached because of an access error, then the Primary RVG is put into PASSTHRU mode. In PASSTHRU mode, writes to the data volume are passed directly to the underlying data volume, bypassing the SRL. No RLINKs receive the writes. Use `vxprint -l` on the RVG to see if the `passthru` flag is set. Associating a new SRL will clear PASSTHRU mode, and the Secondary node RVGs must be synchronized.

Primary SRL volume error at reboot

If the Primary SRL has an error during reboot, there is a possibility that the disks or arrays containing the SRL have not yet come online. Because of this, instead of placing the RVG in PASSTHRU mode, VVR does not recover the RVG. When the SRL becomes available, issue the following commands to recover the RVG and the RLINK:

```
# vxrvg -g diskgroup recover rvg_name
# vxrlink -g diskgroup recover rlink_name
```

After this error has occurred and you have successfully recovered the RVG, if you dissociate a volume from the RVG, you may see the following message:

```
Because there could be outstanding writes in the SRL, the data volume
being dissociated should be considered out-of-date and inconsistent
```

You can ignore this message.

If the SRL is permanently lost, create a new SRL.

See "Recovering from SRL header error" on page 140.

In this case, it is possible that writes that had succeeded on the old SRL and acknowledged to the application, were not yet flushed to the data volumes and are now lost. Consequently, you must restore the data volumes from backup before proceeding. Because this causes the data volumes to be completely rewritten, it is recommended that you detach the RLINKs and synchronize them after the restore operation is complete.

Primary SRL volume overflow recovery

Because the size of the Primary SRL is finite, prolonged halts in update activity to any RLINK can exceed the log's ability to maintain all the necessary update history to bring an RLINK up-to-date. When this occurs, the RLINK in question is marked as STALE and requires manual recovery before replication can proceed.

A STALE RLINK can only be brought up-to-date by using automatic synchronization or a block-level backup and Storage Checkpoint. The other RLINKs, the RVG, and the SRL volume are all still operational.

SRL overflow protection can be set up to prevent SRL overflow, and is the default. Instead of allowing the RLINK to become STALE, dcm logging is initiated. At a later time when the communication link is not overloaded, you can incrementally resynchronize the RLINK using the `vradmin resync rvg` command.

Primary SRL header error cleanup and recovery

An SRL header failure on the Primary is a serious error. All RLINKs are lost and must be recovered using a Primary Storage Checkpoint. Because information about data volume errors is kept in the SRL header, the correct status of data volumes cannot be guaranteed under all occurrences of this error. For this reason, we recommend that the SRL be mirrored.

If an SRL header error occurs during normal operation and you notice it before a reboot occurs, you can be certain that any data volumes that have also (simultaneously) failed will have a status of DETACHED. If the system is rebooted before the `vxprint` command shows the volumes to be in the DETACHED state, the status of any failed data volumes may be lost. Both these cases involve multiple errors and are unlikely, but it is important to understand that the state of Primary data volumes can be suspect with this type of error.

When a Primary SRL header error occurs, writes to the RVG continue; however, all RLINKs are put in the STALE state. The RVG is operating in PASSTHRU mode.

Recovering from SRL header error

Recovering from an SRL header error requires dissociating the SRL from the RVG, repairing the SRL, and completely synchronizing all the RLINKs.

To recover from an SRL header error

1 Stop the RVG.

```
# vxrvg -g hrdg stop hr_rvg
```

2 Dissociate the SRL from the RVG.

```
# vxvol -g hrdg dis hr_srl
```

3 Repair or restore the SRL. Even if the problem can be fixed by repairing the underlying subdisks, the SRL must still be dissociated and reassociated to initialize the SRL header.

4 Make sure the SRL is started, and then reassociate the SRL:

```
# vxvol -g hrdg start hr_srl
# vxvol -g hrdg aslog hr_rvg hr_srl
```

5 Start the RVG:

```
# vxrvg -g hrdg start hr_rvg
```

6 Restore the data volumes from backup if needed. Synchronize all the RLINKs.

Secondary data volume error cleanup and recovery

If an I/O error occurs during access of a Secondary data volume, the data volume is automatically detached from the RVG and the RLINKs are disconnected. A subsequent attempt by the Primary to connect to the Secondary fails and a message that the Secondary volumes are stopped is displayed. The Primary is unaffected and writes continue to be logged into the SRL. After the Secondary data volume error is fixed and the data volume is started, the RLINKs automatically reconnect.

If there is no suitable Primary or Secondary Storage Checkpoint, detach the RLINKs on both the Primary and Secondary, and then synchronize the RLINKs.

Recovery using a Secondary Storage Checkpoint

This section explains how to recover from a Secondary data volume error using a Secondary Storage Checkpoint.

On the Secondary (london):

1 Repair the failed data volume. You need not dissociate the data volume if the problem can be fixed by repairing the underlying subdisks.

2 Make sure that the data volume is started:

```
# vxvol -g hrdg start hr_dv01
```

3 Restore data from the Secondary Storage Checkpoint backup to all the volumes. If all volumes are restored from backup, the Secondary will remain consistent during the synchronization. Restore the RLINK by issuing the following command:

```
#  vxrlink -g hrdg -c sec_chkpt restore rlk_seattle_hr_rvg
```

Cleanup using a Primary Storage Checkpoint

On the Secondary (`london`):

1. Repair the failed data volume as above. Be sure that the data volume is started before proceeding:

   ```
   # vxvol -g hrdg start hr_dv01
   ```

2. Detach the RLINK to enable writing to the Secondary data volumes:

   ```
   # vxrlink -g hrdg det rlk_seattle_hr_rvg
   ```

3. Restore data from the Primary Storage Checkpoint backup to all data volumes. Unlike restoration from a Secondary Storage Checkpoint, the Primary Storage Checkpoint data must be loaded onto all Secondary data volumes, not just the failed volume. If a usable Primary Storage Checkpoint does not already exist, make a new Storage Checkpoint.

4. Reattach the RLINK.

   ```
   # vxrlink -g hrdg att rlk_seattle_hr_rvg
   ```

On the Primary (`seattle`):

Detach the RLINK and then reattach it from the Primary Storage Checkpoint using the following commands:

```
# vxrlink -g hrdg det rlk_london_hr_rvg
# vxrlink -g hrdg -c primary_checkpoint att rlk_london_hr_rvg
```

Secondary SRL volume error cleanup and recovery

The Secondary SRL is used only during atomic recovery of an RLINK and when an IBC is active. If I/O errors occur during recovery of the Secondary SRL, the recovery fails, the SRL volume is automatically detached, and the RLINK is forced to the pause state. Manual intervention is required to repair the physical problem, reattach the SRL, and resume the RLINK. Upon resumption, an automatic recovery of the RVG is retried and if it succeeds, update activity can continue. The only problem occurs if the Primary SRL overflows before the repair is complete, in which case a full synchronization is required.

If an error occurs in the data portion of the SRL, the RLINK is forced to the PAUSE state with the `secondary_paused` flag set. The SRL is not dissociated.

If an error occurs in the SRL header, the Secondary RVG is forced to the FAIL state and the SRL is dissociated.

On the Secondary (`london`):

1 Dissociate the SRL, fix it, and then re-associate it. The dissociation and re-association is necessary even if the problem can be fixed by repairing the underlying subdisks because this sequence initializes the SRL header.

```
# vxvol -g hrdg dis hr_srl
```

Fix or replace the SRL. Be sure the SRL is started before associating it:

```
# vxvol -g hrdg start hr_srl
# vxvol -g hrdg aslog hr_rvg hr_srl
```

2 Run the RLINK resume operation to clear the `secondary_log_err` flag.

```
# vxrlink -g hrdg resume rlk_seattle_hr_rvg
```

Secondary SRL header error cleanup and recovery

An SRL header failure on the Secondary puts the Secondary RVG into the fail state, and sets the RLINK state to the PAUSE state on both the Primary and Secondary. Because information about data volume errors is kept in the SRL header, the correct state of data volumes is not guaranteed in all cases. If a Secondary SRL header failure occurs during normal operation and is noticed before a reboot occurs, any data volumes that also failed will have a state of DETACHED. If the system is rebooted before the `vxprint` command shows the volumes to be in the DETACHED state, the status of any failed data volumes may be lost. Both these cases involve multiple errors and are unlikely, but it is important to understand that the state of Secondary data volumes can be suspect with this type of error.

To cleanup and recover the SRL header failure

1 Dissociate the SRL volume.

```
# vxvol -g hrdg dis hr_srl
```

2 Repair the SRL volume. Even if the problem can be fixed by repairing the underlying subdisks, the SRL volume must still be dissociated and re-associated to initialize the SRL header.

3 Start the SRL volume. Then, re-associate it.

```
# vxvol -g hrdg start hr_srl
# vxvol -g hrdg aslog hr_rvg hr_srl
```

4 Start the RVG.

```
# vxrvg -g hrdg start hr_rvg
```

5 If the integrity of the data volumes is not suspect, just resume the RLINK.

```
# vxrlink -g hrdg resume rlk_seattle_hr_rvg
```

OR

If the integrity of the data volumes is suspect, and a Secondary Storage Checkpoint backup is available, restore from the Secondary Storage Checkpoint.

```
# vxrlink -g hrdg det rlk_seattle_hr_rvg
# vxrlink -g hrdg -f att rlk_seattle_hr_rvg
# vxrlink -g hrdg -w pause rlk_seattle_hr_rvg
```

Restore the Secondary Storage Checkpoint backup data on to the data volumes.

```
# vxrlink -g hrdg -c secondary_checkpoint restore \
  rlk_seattle_hr_rvg
```

OR

If the integrity of the data volumes is suspect and no Secondary Storage Checkpoint is available, synchronize the Secondary using a block-level backup and Primary Storage Checkpoint.

As an alternative, you can also use automatic synchronization.

```
# vxrlink -g hrdg det rlk_seattle_hr_rvg
```

On the Secondary, restore the Primary Storage Checkpoint backup data to the data volumes.

```
# vxrlink -g hrdg -f att rlk_seattle_hr_rvg
```

On the Primary (`seattle`):

```
# vxrlink -g hrdg -c primary_checkpoint att \
  rlk_london_hr_rvg
```

Secondary SRL header error at reboot

If the secondary SRL has an error after a reboot, it is not possible to recover it, even if the SRL subsequently becomes available. Ignore the following message:

```
VxVM VVR vxrvg ERROR V-5-1-0 RVG rvg_name cannot be recovered
because SRL is not accessible. Try recovering the RVG after the
SRL becomes available using vxrecover -s command
```

To reset the SRL volume

1. Dissociate the SRL:

   ```
   # vxvol -g hrdg -f dis srl
   ```

   Ignore the following messages:

   ```
   VxVM vxvol WARNING V-5-1-0 WARNING: Rvg rvgname has not been
   recovered because the SRL is not available. The data volumes may
   be out-of-date and inconsistent
   VxVM vxvol WARNING V-5-1-0 The data volumes in the rvg rvgname
   cannot be recovered because the SRL is being dissociated.
   Restore the data volumes from backup before starting the applications
   ```

2. Create a new SRL volume, *new_srl* and continue as follows:

   ```
   # vxvol -g hrdg aslog rvg_name new_srl
   # vxrlink -g hrdg recover rlink_name
   # vxrlink -g hrdg -f att rlink_name
   # vxrvg -g hrdg start rvg_name
   ```

   If replication was frozen due to receipt of an IBC, the data in the SRL is lost but there is no indication of this problem. To see whether this was the case, examine the `/var/adm/messages` file for a message such as:

   ```
       WARNING: VxVM VVR vxio V-5-0-259 Replication frozen for rlink
       <rlink>
   ```

   If this is the last message for the RLINK, that is, if there is no subsequent message stating that replication was unfrozen, the Primary RLINK must be completely resynchronized.

Section 3

Troubleshooting Dynamic Multi-Pathing

Chapter 12

Dynamic Multi-Pathing troubleshooting

This chapter includes the following topics:

Displaying extended attributes after upgrading to DMP 6.0

You may see the following changes in functionality when you upgrade to DMP 6.0 from the Storage Foundation 5.1 release:

- The device names that are listed in the `vxdisk list` output do not display the Array Volume IDs (AVIDs).
- The `vxdisk -e list` output does not display extended attributes.
- An Active/Passive (A/P) or ALUA array is claimed as Active/Active (A/A).

This behavior may be because the LUNs are controlled by the native multi-pathing driver, MPxIO.

To check whether LUNs are controlled by the native multi-pathing driver

◆ Check the output of the following command. The LUN is controlled by MPxIO if the controller of the affected LUN has the physical name (PNAME) as /scsi_vhci:

```
# vxdmpadm getctlr all
```

You can migrate the LUNs from the control of the native multi-pathing driver to DMP control.

- To migrate to DMP with Veritas Volume Manager, refer to the section on disabling MPxIO in the *Veritas Storage Foundation Administrator's Guide*.
- To migrate to DMP with OS native volume support, refer to the section on migrating to DMP from MPxIO in the *Veritas Dynamic Multi-Pathing Adminstrator's Guide*.

Downgrading the array support

The array support is available in a single package, `VRTSaslapm`, that includes Array Support Libraries (ASLs) and Array Policy Modules (APMs). Each major release of Veritas Storage Foundation and High Availability Solutions includes the supported `VRTSaslapm` package, which is installed as part of the product installation. Between major releases, Symantec may provide additional array support through updates to the `VRTSaslapm` package.

If you have issues with an updated `VRTSaslapm` package, Symantec may recommend that you downgrade to a previous version of the ASL/APM package. You can only revert to a package that is supported for the installed release of Veritas Storage Foundation and High Availability Solutions . To perform the downgrade while the system is online, do not remove the installed package. Instead, you can install the previous version of the package over the new package. This method prevents multiple instances of the `VRTSaslapm` package from being installed.

Use the following method to downgrade the `VRTSaslapm` package.

To downgrade the ASL/APM package

1. Create a response file to the `pkgadd` command that specifies instance=overwrite. The following example shows a response file:

```
#
# Copyright 2004 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
#ident  "@(#)default    1.7     04/12/21 SMI"
#
mail=
instance=overwrite
partial=ask
runlevel=ask
idepend=ask
rdepend=ask
space=ask
setuid=ask
conflict=ask
action=ask
networktimeout=60
networkretries=3
authentication=quit
keystore=/var/sadm/security
proxy=
basedir=default
```

2. To downgrade the package, use the following command:

```
pkgadd -a <response_file> -d ./VRTSaslapm.pkg
```

Section 4

Troubleshooting Veritas Storage Foundation Cluster File System High Availability

Chapter 13

Troubleshooting Veritas Storage Foundation Cluster File System High Availability

This chapter includes the following topics:

- About troubleshooting Veritas Storage Foundation Cluster File System High Availability
- Troubleshooting CFS
- Troubleshooting fenced configurations
- Troubleshooting Cluster Volume Manager in SFHA Solutions clusters

About troubleshooting Veritas Storage Foundation Cluster File System High Availability

Use the information in this chapter to diagnose setup or configuration problems that you might encounter. For issues that arise from the component products, it may be necessary to refer to the appropriate documentation to resolve it.

Troubleshooting information for I/O fencing also applies to troubleshooting Veritas Storage Foundation Cluster File System High Availability (SFCFSHA).

See “Troubleshooting I/O fencing” on page 192.

Troubleshooting CFS

This section discusses troubleshooting CFS problems.

Incorrect order in root user's <library> path

An incorrect order in the root user’s <library> path can cause the system to hang while changing the primary node in the Cluster File System or the RAC cluster.

If the <library> path of the root user contains an entry pointing to a Cluster File System (CFS) file system before the /usr/lib entry, the system may hang when trying to perform one of the following tasks:

- Changing the primary node for the CFS file system
- Unmounting the CFS files system on the primary node
- Stopping the cluster or the service group on the primary node

This configuration issue occurs primarily in a RAC environment with Oracle binaries installed on a shared CFS file system.

The following is an example of a <library path> that may cause the system to hang:

```
LD_LIBRARY_PATH=/opt/SUNWspro/lib:\
/app/oracle/orahome/lib:/usr/lib:/usr/ccs/lib
```

In the above example, /app/oracle is a CFS file system, and if the user tries to change the primary node for this file system, the system will hang. The user is still able to ping and telnet to the system, but simple commands such as `ls` will not respond. One of the first steps required during the changing of the primary node is freezing the file system cluster wide, followed by a quick issuing of the `fsck` command to replay the intent log.

Since the initial entry in <library> path is pointing to the frozen file system itself, the `fsck` command goes into a deadlock situation. In fact, all commands (including `ls`) which rely on the <library> path will hang from now on.

The recommended procedure to correct for this problem is as follows: Move any entries pointing to a CFS file system in any user's (especially root) <library> path towards the end of the list after the entry for /usr/lib

Therefore, the above example of a <library path> would be changed to the following:

```
LD_LIBRARY_PATH=/opt/SUNWspro/lib:\
/usr/lib:/usr/ccs/lib:/app/oracle/orahome/lib
```

CFS commands might hang when run by non-root (2403263)

The CFS commands might hang when run by non-root.

Workaround

To resolve this issue

◆ Use `halogin` command to save the authentication information before running any CFS commands on a non-root sessions.

When you run the `halogin` command, VCS stores encrypted authentication information in the user's home directory.

Troubleshooting fenced configurations

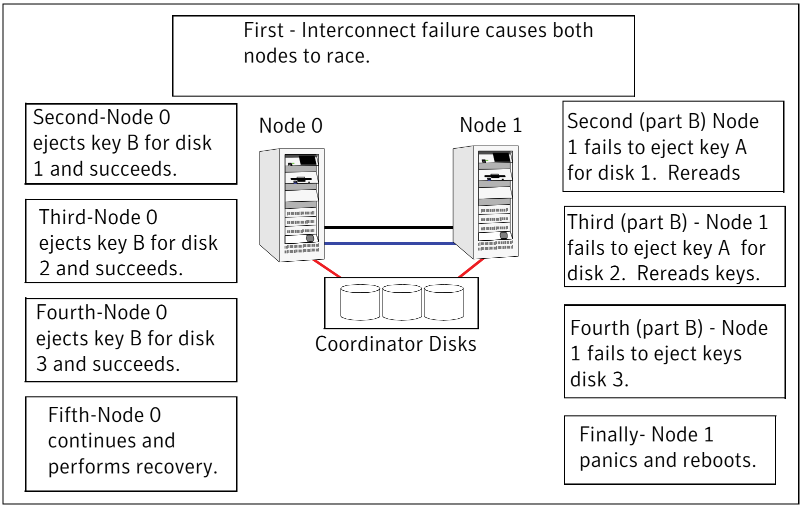
The following information describes network partitioning in a fenced environment.

See the *Veritas Cluster Server Administrator's Guide*.

Example of a preexisting network partition (split-brain)

Figure 13-1 shows a two-node cluster in which the severed cluster interconnect poses a potential split-brain condition.

Figure 13-1 Preexisting network partition (split-brain)



First - Interconnect failure causes both nodes to race.
Second-Node 0 ejects key B for disk 1 and succeeds.
Third-Node 0 ejects key B for disk 2 and succeeds.
Fourth-Node 0 ejects key B for disk 3 and succeeds.
Fifth-Node 0 continues and performs recovery.
Node 0
Node 1
Coordinator Disks
Second (part B) Node 1 fails to eject key A for disk 1. Rereads
Third (part B) - Node 1 fails to eject key A for disk 2. Rereads keys.
Fourth (part B) - Node 1 fails to eject keys disk 3.
Finally- Node 1 panics and reboots.

Because the fencing module operates identically on each system, both nodes assume the other is failed, and carry out fencing operations to insure the other node is ejected. The VCS GAB module on each node determines the peer has failed due to loss of heartbeats and passes the membership change to the fencing module.

Each side “races” to gain control of the coordinator disks. Only a registered node can eject the registration of another node, so only one side successfully completes the command on each disk.

The side that successfully ejects the peer from a majority of the coordinator disks wins. The fencing module on the winning side then passes the membership change up to VCS and other higher-level packages registered with the fencing module, allowing VCS to invoke recovery actions. The losing side forces a kernel panic and reboots.

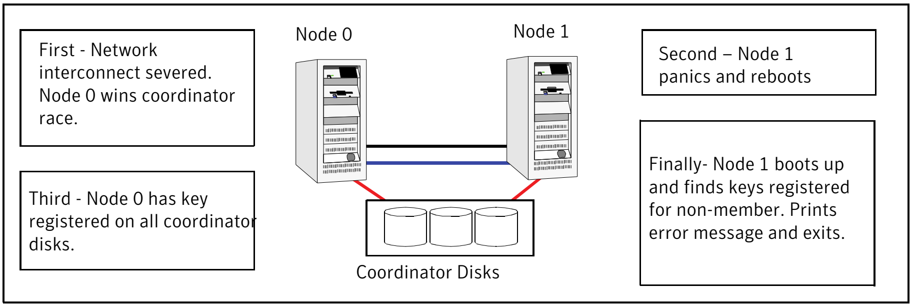
Recovering from a preexisting network partition (split-brain)

The fencing module `vxfen` prevents a node from starting up after a network partition and subsequent panic and reboot of a node.

Example Scenario I

Figure 13-2 scenario could cause similar symptoms on a two-node cluster with one node shut down for maintenance. During the outage, the private interconnect cables are disconnected.

Figure 13-2 Example scenario I


First - Network interconnect severed. Node 0 wins coordinator race.
Third - Node 0 has key registered on all coordinator disks.
Node 0
Node 1
Coordinator Disks
Second – Node 1 panics and reboots
Finally- Node 1 boots up and finds keys registered for non-member. Prints error message and exits.

In example scenario I, the following occurs:

- Node 0 wins a coordinator race following to a network failure.
- Node 1 panics and reboots.
- Node 0 has keys registered on the coordinator disks. When Node 1 boots up, it sees the Node 0 keys, but cannot see Node 0 in the current GAB membership. It senses a potential preexisting split brain and causes the vxfen module to

print an error message to the console. The vxfen module prevents fencing from starting, which, in turn, prevents VCS from coming online.

Suggested solution: Shut down Node 1, reconnect the cables, and restart Node 1.

Example Scenario II

Similar to example scenario I, if private interconnect cables are disconnected in a two-node cluster, Node 1 is fenced out of the cluster, panics, and reboots. If before the private interconnect cables are fixed and Node 1 rejoins the cluster, Node 0 reboots and remote (or just reboots). No node can write to the data disks until the private networks are fixed. This is because GAB membership cannot be formed, therefore the cluster cannot be formed.

Similar to example scenario I, if private interconnect cables are disconnected in a two-node cluster, Node 1 is fenced out of the cluster, panics, and reboots. No node can write to the data disks until the private networks are fixed. This is because GAB membership cannot be formed, therefore the cluster cannot be formed.

Suggested solution: Shut down both nodes, reconnect the cables, restart the nodes.

Example Scenario III

Similar to example scenario II, if private interconnect cables are disconnected in a two-node cluster, Node 1 is fenced out of the cluster, panics, and reboots. If before the private interconnect cables are fixed and Node 1 rejoins the cluster, Node 0 panics due to hardware failure and cannot come back up, Node 1 cannot rejoin.

Suggested solution: Shut down Node 1, reconnect the cables, restart the node. You must then clear the registration of Node 0 from the coordinator disks.

To fix scenario III

1. On Node 1, type the following command:

   ```
   # /opt/VRTSvcs/vxfen/bin/vxfenclearpre
   ```

2. Restart the node.
3. Run:

   ```
   # gabconfig -cx
   ```

Troubleshooting Cluster Volume Manager in SFHA Solutions clusters

This section discusses troubleshooting CVM problems.

CVM group is not online after adding a node to the SFHA Solutions cluster

The possible causes for the CVM group being offline after adding a node to the cluster are as follows:

- The cssd resource is configured as a critical resource in the cvm group.
- Other resources configured in the cvm group as critical resources are not online.

To resolve the issue if cssd is configured as a critical resource

1. Log onto one of the nodes in the existing cluster as the root user.
2. Configure the cssd resource as a non-critical resource in the cvm group:

```
# haconf -makerw
# hares -modify cssd Critical 0
# haconf -dump -makero
```

To resolve the issue if other resources in the group are not online

1. Log onto one of the nodes in the existing cluster as the root user.
2. Bring the resource online:

```
# hares -online resource_name -sys system_name
```

3. Verify the status of the resource:

```
# hastatus -resource  resource_name
```

4. If the resource is not online, configure the resource as a non-critical resource:

```
# haconf -makerw
# hares -modify resource_name Critical 0
# haconf -dump -makero
```

Shared disk group cannot be imported in SFHA Solutions cluster

If you see a message resembling:

```
vxvm:vxconfigd:ERROR:vold_pgr_register(/dev/vx/rdmp/disk_name):
local_node_id<0
Please make sure that CVM and vxfen are configured
and operating correctly
```

First, make sure that CVM is running. You can see the CVM nodes in the cluster by running the vxclustadm nidmap command.

```
# vxclustadm nidmap
Name          CVM Nid    CM Nid     State
system01         1          0          Joined: Master
system02         0          1          Joined: Slave
```

This above ouput shows that CVM is healthy, with system system01 as the CVM master. If CVM is functioning correctly, then the output above is displayed when CVM cannot retrieve the node ID of the local system from the `vxfen` driver. This usually happens when port b is not configured.

To verify vxfen driver is configured

◆ Check the GAB ports with the command:

```
# gabconfig -a
```

Port b must exist on the local system.

Error importing shared disk groups in SFHA Solutions cluster

The following message may appear when importing shared disk group:

```
VxVM vxdg ERROR V-5-1-587 Disk group disk group name: import
failed: No valid disk found containing disk group
```

You may need to remove keys written to the disk.

For information about removing keys written to the disk:

Unable to start CVM in SFHA Solutions cluster

If you cannot start CVM, check the consistency between the `/etc/llthosts` and `main.cf` files for node IDs.

You may need to remove keys written to the disk.

For information about removing keys written to the disk:

CVMVolDg not online even though CVMCluster is online in SFHA Solutions cluster

When the CVMCluster resource goes online, then all shared disk groups that have the auto-import flag set are automatically imported. If the disk group import fails for some reason, the CVMVolDg resources fault. Clearing and taking the CVMVolDg type resources offline does not resolve the problem.

To resolve the resource issue

1. Fix the problem causing the import of the shared disk group to fail.
2. Offline the cvm group containing the resource of type CVMVolDg as well as the service group containing the CVMCluster resource type.
3. Bring the cvm group containing the CVMCluster resource online.
4. Bring the cvm group containing the CVMVolDg resource online.

Shared disks not visible in SFHA Solutions cluster

If the shared disks in `/dev/rdsk` are not visible, perform the following tasks:

Make sure that all shared LUNs are discovered by the HBA and SCSI layer. This can be verified by running the `ls -ltr` command on any of the disks under `/dev/rdsk/*`.

For example:

```
# ls -ltr /dev/rdsk/disk_name
lrwxrwxrwx   1 root     root          81 Aug 18 11:58
c2t5006016141E02D28d4s2
-> ../../devices/pci@7c0/pci@0/pci@8/SUNW,qlc@0/fp@0,
0/ssd@w5006016141e02d28,4:c,raw
lrwxrwxrwx   1 root     root          81 Aug 18 11:58
c2t5006016141E02D28d3s2
-> ../../devices/pci@7c0/pci@0/pci@8/SUNW,qlc@0/fp@0,
0/ssd@w5006016141e02d28,3:c,raw
lrwxrwxrwx   1 root     root          81 Aug 18 11:58
c2t5006016141E02D28d2s2
-> ../../devices/pci@7c0/pci@0/pci@8/SUNW,qlc@0/fp@0,
0/ssd@w5006016141e02d28,2:c,raw
lrwxrwxrwx   1 root     root          81 Aug 18 11:58
c2t5006016141E02D28d1s2
```

```
-> ../../devices/pci@7c0/pci@0/pci@8/SUNW,qlc@0/fp@0,
0/ssd@w5006016141e02d28,1:c,raw
```

If all LUNs are not discovered by SCSI, the problem might be corrected by specifying `dev_flags` or `default_dev_flags` and *max_luns* parameters for the SCSI driver.

If the LUNs are not visible in `/dev/rdsk/*` files, it may indicate a problem with SAN configuration or zoning.

For a system running Solaris OS, perform the following additional steps:

Perform the following additional steps:

- Check the file `/kernel/drv/sd.conf` to see if the new LUNs were added.
- Check the format to see if the LUNs have been labeled in the server.
- Check to see is the disk is seen, using the following command:

  ```
  # prtvtoc /dev/rdsk/disk_name
  ```

Section 5

Troubleshooting Veritas Cluster Server

Chapter 14

Troubleshooting and recovery for VCS

This chapter includes the following topics:

VCS message logging

VCS generates two types of logs: the engine log and the agent log. Log file names are appended by letters. Letter A indicates the first log file, B the second, C the third, and so on.

The engine log is located at /var/VRTSvcs/log/engine_A.log. The format of engine log messages is:

Timestamp (Year/MM/DD) | Mnemonic | Severity | UMI | Message Text

- *Timestamp*: the date and time the message was generated.
- *Mnemonic*: the string ID that represents the product (for example, VCS).
- *Severity*: levels include CRITICAL, ERROR, WARNING, NOTICE, and INFO (most to least severe, respectively).
- *UMI*: a unique message ID.
- *Message Text*: the actual message generated by VCS.

A typical engine log resembles:

```
2011/07/10 16:08:09 VCS INFO V-16-1-10077 Received new
cluster membership
```

The agent log is located at `/var/VRTSvcs/log/<agent>.log`. The format of agent log messages resembles:

Timestamp (Year/MM/DD) | Mnemonic | Severity | UMI | Agent Type | Resource Name | Entry Point | Message Text

A typical agent log resembles:

```
2011/07/10 10:38:23 VCS WARNING V-16-2-23331
Oracle:VRT:monitor:Open for ora_lgwr failed, setting
cookie to null.
```

Note that the logs on all nodes may not be identical because

- VCS logs local events on the local nodes.
- All nodes may not be running when an event occurs.

VCS prints the warning and error messages to STDERR.

If the VCS engine, Command Server, or any of the VCS agents encounter some problem, then First Failure Data Capture (FFDC) logs are generated and dumped along with other core dumps and stack traces to the following location:

- For VCS engine: `$VCS_DIAG/diag/had`

- For Command Server: `$VCS_DIAG/diag/CmdServer`
- For VCS agents: `$VCS_DIAG/diag/agents/`*`type`*, where *type* represents the specific agent type.

The default value for variable $VCS_DIAG is `/var/VRTSvcs/`.

If the debug logging is not turned on, these FFDC logs are useful to analyze the issues that require professional support.

GAB message logging

If GAB encounters some problem, then First Failure Data Capture (FFDC) logs are also generated and dumped.

When you have configured GAB, GAB also starts a GAB logging daemon (`/opt/VRTSgab/gablogd`). GAB logging daemon is enabled by default. You can change the value of the GAB tunable parameter `gab_ibuf_count` to disable the GAB logging daemon.

This GAB logging daemon collects the GAB related logs when a critical events such as an iofence or failure of the master of any GAB port occur, and stores the data in a compact binary form. You can use the `gabread_ffdc` utility as follows to read the GAB binary log files:

```
/opt/VRTSgab/gabread_ffdc  binary_logs_files_location
```

You can change the values of the following environment variables that control the GAB binary log files:

- GAB_FFDC_MAX_INDX: Defines the maximum number of GAB binary log files
  The GAB logging daemon collects the defined number of log files each of eight MB size. The default value is 20, and the files are named `gablog.1` through `gablog.20`. At any point in time, the most recent file is the `gablog.1` file.
- GAB_FFDC_LOGDIR: Defines the log directory location for GAB binary log files
  The default location is:
  `/var/adm/gab_ffdc`
  Note that the gablog daemon writes its log to the `glgd_A.log` and `glgd_B.log` files in the same directory.

You can either define these variables in the following GAB startup file or use the `export` command. You must restart GAB for the changes to take effect.

```
/etc/default/gab
```

Enabling debug logs for agents

This section describes how to enable debug logs for VCS agents.

To enable debug logs for agents

1. Set the configuration to read-write:

   ```
   # haconf -makerw
   ```

2. Enable logging and set the desired log levels. The following example depicts the command for the IPMultiNIC resource type.

   ```
   # hatype -modify IPMultiNIC LogDbg DBG_1 DBG_2 DBG_4 DBG_21
   ```

   See the description of the LogDbg attribute for more information.

3. For script-based agents, run the `halog` command to add the messages to the engine log:

   ```
   # halog -addtags DBG_1 DBG_2 DBG_4 DBG_21
   ```

4. Save the configuration.

   ```
   # haconf -dump -makero
   ```

If DBG_AGDEBUG is set, the agent framework logs for an instance of the agent appear in the agent log on the node on which the agent is running.

Enabling debug logs for IMF

Run the following commands to enable additional debug logs for Intelligent Monitoring Framework (IMF). The messages get logged in the agent-specific log file `/var/VRTSvcs/log/`*`agentname`*`_A.log`.

See "Troubleshooting Intelligent Monitoring Framework (IMF)" on page 185.

To enable additional debug logs

1 For Process, Mount, and Application agents:

```
# hatype -modify agentname LogDbg
DBG_AGDEBUG DBG_AGTRACE DBG_AGINFO DBG_1 DBG_2
DBG_3 DBG_4 DBG_5 DBG_6 DBG_7
```

2 For Oracle and Netlsnr agents:

```
# hatype -modify agentname LogDbg
DBG_AGDEBUG DBG_AGTRACE DBG_AGINFO DBG_1 DBG_2
DBG_3 DBG_4 DBG_5 DBG_6 DBG_7
DBG_8 DBG_9 DBG_10
```

3 For CFSMount agent:

```
# hatype -modify agentname LogDbg
DBG_AGDEBUG DBG_AGTRACE DBG_AGINFO DBG_1 DBG_2
DBG_3 DBG_4 DBG_5 DBG_6 DBG_7
DBG_8 DBG_9 DBG_10 DBG_11 DBG_12
DBG_13 DBG_14 DBG_15 DBG_16
DBG_17 DBG_18 DBG_19 DBG_20 DBG_21
```

4 For CVMvxconfigd agent, you do not have to enable any additional debug logs.

5 For AMF driver in-memory trace buffer:

```
# amfconfig -S errlevel all all
```

If you had enabled AMF driver in-memory trace buffer, you can view the additional logs using the `amfconfig -p dbglog` command.

Enabling debug logs for the VCS engine

You can enable debug logs for the VCS engine, VCS agents, and HA commands in two ways:

- To enable debug logs at run-time, use the `halog -addtags` command.
- To enable debug logs at startup, use the VCS_DEBUG_LOG_TAGS environment variable. You must set the VCS_DEBUG_LOG_TAGS before you start HAD or before you run HA commands.
  Examples:

```
# export VCS_DEBUG_LOG_TAGS="DBG_TRACE DBG_POLICY"
# hastart

# export VCS_DEBUG_LOG_TAGS="DBG_AGINFO DBG_AGDEBUG DBG_AGTRACE"
# hastart

# export VCS_DEBUG_LOG_TAGS="DBG_IPM"
# hagrp -list
```

Note: Debug log messages are verbose. If you enable debug logs, log files might fill up quickly.

About debug log tags usage

The following table illustrates the use of debug tags:

Entity	Debug logs used
Agent functions	DBG_1 to DBG_21
Agent framework	DBG_AGTRACE DBG_AGDEBUG DBG_AGINFO
Icmp agent	DBG_HBFW_TRACE DBG_HBFW_DEBUG DBG_HBFW_INFO

Entity	Debug logs used
HAD	DBG_AGENT (for agent-related debug logs) DBG_ALERTS (for alert debug logs) DBG_CTEAM (for GCO debug logs) DBG_GAB, DBG_GABIO (for GAB debug messages) DBG_GC (for displaying global counter with each log message) DBG_INTERNAL (for internal messages) DBG_IPM (for Inter Process Messaging) DBG_JOIN (for Join logic) DBG_LIC (for licensing-related messages) DBG_NTEVENT (for NT Event logs) DBG_POLICY (for engine policy) DBG_RSM (for RSM debug messages) DBG_TRACE (for trace messages) DBG_SECURITY (for security-related messages) DBG_LOCK (for debugging lock primitives) DBG_THREAD (for debugging thread primitives) DBG_HOSTMON (for HostMonitor debug logs)

Gathering VCS information for support analysis

You must run the `hagetcf` command to gather information when you encounter issues with VCS. Symantec Technical Support uses the output of these scripts to assist with analyzing and solving any VCS problems. The `hagetcf` command gathers information about the installed software, cluster configuration, systems, logs, and related information and creates a gzip file.

See the `hagetcf`(1M) manual page for more information.

To gather VCS information for support analysis

◆ Run the following command on each node:

```
# /opt/VRTSvcs/bin/hagetcf
```

The command prompts you to specify an output directory for the gzip file. You may save the gzip file to either the default /tmp directory or a different directory.

Troubleshoot and fix the issue.

See "Troubleshooting the VCS engine" on page 176.

See "Troubleshooting VCS startup" on page 184.

See "Troubleshooting service groups" on page 187.

See "Troubleshooting resources" on page 190.

See "Troubleshooting notification" on page 210.

See "Troubleshooting and recovery for global clusters" on page 210.

See "Troubleshooting the steward process" on page 214.

If the issue cannot be fixed, then contact Symantec technical support with the file that the hagetcf command generates.

Gathering LLT and GAB information for support analysis

You must run the getcomms script to gather LLT and GAB information when you encounter issues with LLT and GAB. The getcomms script also collects core dump and stack traces along with the LLT and GAB information.

To gather LLT and GAB information for support analysis

1 If you had changed the default value of the GAB_FFDC_LOGDIR parameter, you must again export the same variable before you run the `getcomms` script.

  See "GAB message logging" on page 169.

2 Run the following command to gather information:

```
# /opt/VRTSgab/getcomms
```

  The script uses rsh by default. Make sure that you have configured passwordless rsh. If you have passwordless ssh between the cluster nodes, you can use the `-ssh` option. To gather information on the node that you run the command, use the `-local` option.

  Troubleshoot and fix the issue.

  See "Troubleshooting Low Latency Transport (LLT)" on page 179.

  See "Troubleshooting Group Membership Services/Atomic Broadcast (GAB)" on page 183.

  If the issue cannot be fixed, then contact Symantec technical support with the file `/tmp/commslog.`*`time_stamp`*`.tar` that the `getcomms` script generates.

Gathering IMF information for support analysis

You must run the `getimf` script to gather information when you encounter issues with IMF (Intelligent Monitoring Framework).

To gather IMF information for support analysis

◆ Run the following command on each node:

```
# /opt/VRTSamf/bin/getimf
```

  Troubleshoot and fix the issue.

  See "Troubleshooting Intelligent Monitoring Framework (IMF)" on page 185.

  If the issue cannot be fixed, then contact Symantec technical support with the file that the `getimf` script generates.

Message catalogs

VCS includes multilingual support for message catalogs. These binary message catalogs (BMCs), are stored in the following default locations. The variable *language* represents a two-letter abbreviation.

`/opt/VRTS/messages/`*`language`*`/`*`module_name`*

The VCS command-line interface displays error and success messages in VCS-supported languages. The `hamsg` command displays the VCS engine logs in VCS-supported languages.

The BMCs are:

gcoconfig.bmc	gcoconfig messages
VRTSvcsHbfw.bmc	Heartbeat framework messages
VRTSvcsTriggers.bmc	VCS trigger messages
VRTSvcsWac.bmc	Wide-area connector process messages
vxfen*.bmc	Fencing messages
gab.bmc	GAB command-line interface messages
hagetcf.bmc	hagetcf messages
llt.bmc	LLT command-line interface messages
VRTSvcsAgfw.bmc	Agent framework messages
VRTSvcsAlerts.bmc	VCS alert messages
VRTSvcsApi.bmc	VCS API messages
VRTSvcsCommon.bmc	Common modules messages
VRTSvcsHad.bmc	VCS engine (HAD) messages
VRTSvcs*platform*Agent.bmc	VCS bundled agent messages
VRTSvcs*platformagent_name*.bmc	VCS enterprise agent messages

Troubleshooting the VCS engine

This topic includes information on troubleshooting the VCS engine.

See “Preonline IP check” on page 178.

HAD diagnostics

When the VCS engine HAD dumps core, the core is written to the directory `$VCS_DIAG/diag/had`. The default value for variable *$VCS_DIAG* is `/var/VRTSvcs/`.

When HAD core dumps, review the contents of the `$VCS_DIAG/diag/had` directory. See the following logs for more information:

- Operating system console log
- Engine log
- hashadow log

VCS runs the script `/opt/VRTSvcs/bin/vcs_diag` to collect diagnostic information when HAD and GAB encounter heartbeat problems. The diagnostic information is stored in the `$VCS_DIAG/diag/had` directory.

When HAD starts, it renames the directory to `had.timestamp`, where *timestamp* represents the time at which the directory was renamed.

HAD is not running

On Solaris 10, if you find that HAD does not work, check to make sure that the VCS service is online.

Run the SMF `svcs -l vcs` command:

```
# svcs -l vcs
```

Output should resemble the following:

```
fmri          svc:/system/vcs:default
name          Veritas Cluster Server (VCS) Init service
enabled       true
state         online
next_state    none
state_time    Tue Feb 10 11:27:30 2009
restarter     svc:/system/svc/restarter:default
dependency    require_all/none svc:/system/gab (online)
dependency    require_all/none svc:/system/filesystem/local (online)
```

If the output does not resemble the previous output, refer to Solaris documentation for Service Management Facility (SMF).

HAD restarts continuously

When you start HAD by using the `hastart` command, HAD might restart continuously. The system is unable to come to a RUNNING state and loses its port h membership.

Recommended action: Check the `engine_A.log` file for a message with the identifier V-16-1-10125. The following message is an example:

VCS INFO V-16-1-10125 GAB timeout set to 30000 ms

The value indicates the timeout that is set for HAD to heartbeat with GAB. If system is heavily loaded, a timeout of 30 seconds might be insufficient for HAD to heartbeat with GAB. If required, set the timeout to an appropriate higher value.

DNS configuration issues cause GAB to kill HAD

If HAD is periodically killed by GAB for no apparent reason, review the HAD core files for any DNS resolver functions (res_send(), res_query(), res_search() etc) in the stack trace. The presence of DNS resolver functions may indicate DNS configuration issues.

The VCS High Availability Daemon (HAD) uses the gethostbyname() function. On UNIX platforms, if the file `/etc/nsswitch.conf` has DNS in the hosts entry, a call to the gethostbyname() function may lead to calls to DNS resolver methods.

If the name servers specified in the `/etc/resolve.conf` are not reachable or if there are any DNS configuration issues, the DNS resolver methods called may block HAD, leading to HAD not sending heartbeats to GAB in a timely manner.

Seeding and I/O fencing

When I/O fencing starts up, a check is done to make sure the systems that have keys on the coordinator points are also in the GAB membership. If the `gabconfig` command in `/etc/gabtab` allows the cluster to seed with less than the full number of systems in the cluster, or the cluster is forced to seed with the `gabconfig -x` command, it is likely that this check will not match. In this case, the fencing module will detect a possible split-brain condition, print an error, and HAD will not start.

It is recommended to let the cluster automatically seed when all members of the cluster can exchange heartbeat signals to each other. In this case, all systems perform the I/O fencing key placement after they are already in the GAB membership.

Preonline IP check

You can enable a preonline check of a failover IP address to protect against network partitioning. The check pings a service group's configured IP address to verify that it is not already in use. If it is, the service group is not brought online.

A second check verifies that the system is connected to its public network and private network. If the system receives no response from a broadcast ping to the public network and a check of the private networks, it determines the system is isolated and does not bring the service group online.

To enable the preonline IP check, do one of the following:

- If preonline trigger script is not already present, copy the preonline trigger script from the sample triggers directory into the triggers directory:

  ```
  # cp /opt/VRTSvcs/bin/sample_triggers/VRTSvcs/preonline_ipc
  /opt/VRTSvcs/bin/triggers/preonline
  ```

  Change the file permissions to make it executable.
- If preonline trigger script is already present, create a directory such as `/preonline` and move the existing preonline trigger as T0preonline to that directory. Copy the preonline_ipc trigger as T1preonline to the same directory.
- If you already use multiple triggers, copy the preonline_ipc trigger as TNpreonline, where TN is the next higher TNumber.

Troubleshooting Low Latency Transport (LLT)

This section includes error messages associated with Low Latency Transport (LLT) and provides descriptions and the recommended action.

LLT startup script displays errors

If more than one system on the network has the same clusterid-nodeid pair and the same Ethernet sap/UDP port, then the LLT startup script displays error messages similar to the following:

```
LLT lltconfig ERROR V-14-2-15238 node 1 already exists
in cluster 8383 and has the address - 00:18:8B:E4:DE:27
LLT lltconfig ERROR V-14-2-15241 LLT not configured,
use -o to override this warning
LLT lltconfig ERROR V-14-2-15664 LLT could not
configure any link
LLT lltconfig ERROR V-14-2-15245 cluster id 1 is
already being used by nid 0 and has the
address - 00:04:23:AC:24:2D
LLT lltconfig ERROR V-14-2-15664 LLT could not
configure any link
```

Check the log files that get generated in the `/var/svc/log` directory for any errors.

Recommended action: Ensure that all systems on the network have unique clusterid-nodeid pair. You can use the `lltdump -f` *`device`* `-D` command to get the list of unique clusterid-nodeid pairs connected to the network. This utility is available only for LLT-over-ethernet.

LLT detects cross links usage

If LLT detects more than one link of a system that is connected to the same network, then LLT logs a warning message similar to the following in the syslog:

```
LLT WARNING V-14-1-10498 recvarpack cross links? links 0 and 2 saw
the same peer link number 1 for node 1
```

Recommended Action: This is an informational message. LLT supports cross links. However, if this cross links is not an intentional network setup, then make sure that no two links from the same system go to the same network. That is, the LLT links need to be on separate networks.

LLT link status messages

Table 14-1 describes the LLT logs messages such as trouble, active, inactive, or expired in the syslog for the links.

Table 14-1 LLT link status messages

Message	Description and Recommended action

`LLT INFO V-14-1-10205`
`link 1 (`*`link_name`*`) node 1 in trouble`
	This message implies that LLT did not receive any heartbeats on the indicated link from the indicated peer node for LLT peertrouble time. The default LLT peertrouble time is 2s for hipri links and 4s for lo-pri links.
	Recommended action: If these messages sporadically appear in the syslog, you can ignore them. If these messages flood the syslog, then perform one of the following:
	■ Increase the peertrouble time to a higher value (but significantly lower than the peerinact value). Run the following command:
	`lltconfig -T peertrouble:<value>`
	`for hipri link`
	`lltconfig -T peertroublelo:<value>`
	`for lopri links.`
	See the `lltconfig`(1m) manual page for details.
	■ Replace the LLT link.

`LLT INFO V-14-1-10024`
`link 0 (`*`link_name`*`) node 1 active`
	This message implies that LLT started seeing heartbeats on this link from that node.
	Recommended action: No action is required. This message is informational.

Table 14-1 LLT link status messages *(continued)*

Message	Description and Recommended action
`LLT INFO V-14-1-10032 link 1 (`*`link_name`*`) node 1 inactive 5 sec (510)`	This message implies that LLT did not receive any heartbeats on the indicated link from the indicated peer node for the indicated amount of time. If the peer node has not actually gone down, check for the following: ■ Check if the link has got physically disconnected from the system or switch. ■ Check for the link health and replace the link if necessary.

`LLT INFO V-14-1-10510 sent hbreq (NULL) on link 1 (`*`link_name`*`) node 1. 4 more to go.`
`LLT INFO V-14-1-10510 sent hbreq (NULL) on link 1 (`*`link_name`*`) node 1. 3 more to go.`
`LLT INFO V-14-1-10510 sent hbreq (NULL) on link 1 (`*`link_name`*`) node 1. 2 more to go.`
`LLT INFO V-14-1-10032 link 1 (`*`link_name`*`) node 1 inactive 6 sec (510)`
`LLT INFO V-14-1-10510 sent hbreq (NULL) on link 1 (`*`link_name`*`) node 1. 1 more to go.`
`LLT INFO V-14-1-10510 sent hbreq (NULL) on link 1 (`*`link_name`*`) node 1. 0 more to go.`
`LLT INFO V-14-1-10032 link 1 (`*`link_name`*`) node 1 inactive 7 sec (510)`
`LLT INFO V-14-1-10509 link 1 (`*`link_name`*`) node 1 expired`
	This message implies that LLT did not receive any heartbeats on the indicated link from the indicated peer node for more than LLT peerinact time. LLT attempts to request heartbeat mechanism (sends 5 hbreqs to the peer node) and if the peer node does not respond, LLT marks this link as “expired” for that peer node.
	Recommended action: If the peer node has not actually gone down, check for the following:
	■ Check if the link has got physically disconnected from the system or switch.
	■ Check for the link health and replace the link if necessary.

`LLT INFO V-14-1-10499 recvarpreq link 0 for node 1 addr change from 00:00:00:00:00:00 to 00:18:8B:E4:DE:27`	This message is logged when LLT learns the peer node’s address. Recommended action: No action is required. This message is informational.

Table 14-1 LLT link status messages *(continued)*

Message	Description and Recommended action

On local node that detects the link failure:
`LLT INFO V-14-1-10519 link 0 down`
`LLT INFO V-14-1-10585 local link 0 down for 1 sec`
`LLT INFO V-14-1-10586 send linkdown_ntf on link 1 for local link 0`
`LLT INFO V-14-1-10590 recv linkdown_ack from node 1 on link 1 for local link 0`
`LLT INFO V-14-1-10592 received ack from all the connected nodes`
On peer nodes:
`LLT INFO V-14-1-10589 recv linkdown_ntf from node 0 on link 1 for peer link 0`
`LLT INFO V-14-1-10587 send linkdown_ack to node 0 on link 1 for peer link 0`
	These messages are printed when you have enabled LLT to detect faster failure of links. When a link fails or is disconnected from a node (cable pull, switch failure, and so on), LLT on the local node detects this event and propagates this information to all the peer nodes over the LLT hidden link. LLT marks this link as disconnected when LLT on the local node receives the acknowledgment from all the nodes.

Unexpected db_type warning while stopping LLT that is configured over UDP

When LLT is configured over UDP, sometimes you can see large delays between packets sent from one node to packets received on the receiver. This may occur due to various network issues. In the meanwhile, if the receiver is unconfigured for LLT, the receiver's network stack replies to the sender with the ICMP response that the destination port (LLT port) is no longer available. LLT does not recognize this kind of response as a valid LLT packet and displays a warning message similar to the following:

```
llt: [ID 718506 kern.notice] LLT WARNING V-14-1-10027
llt_lrput: unexpected db_type is 0xd, link 0
llt: [ID 351264 kern.notice] pkt content (first 64
bytes) 45 00 00 68 79 8F 40 00 FF 01 7E B1 C0 A8 01
01 C0 A8 01 02 03 03 80 9A 00 00 00 00 45 00 00 4C
8B 67 40 00 FF 11 6C E5 C0 A8 01 02 C0 A8 01 01 23
41 23 41 00 38 35 A8 05 00 1A 07 00 DD 00 00
```

You can ignore this kind of warning if the packet received is ICMP. In this example, the 10th byte 0x01 indicates that this is an ICMP packet. The 21st and 22nd bytes indicate the ICMP type and code for the packet.

Troubleshooting Group Membership Services/Atomic Broadcast (GAB)

This section includes error messages associated with Group Membership Services/Atomic Broadcast (GAB) and provides descriptions and the recommended action.

Delay in port reopen

If a GAB port was closed and reopened before LLT state cleanup action is completed, then GAB logs a message similar to the following:

```
GAB INFO V-15-1-20102 Port v: delayed reopen
```

Recommended Action: If this issue occurs during a GAB reconfiguration, and does not recur, the issue is benign. If the issue persists, collect commslog from each node, and contact Symantec support.

Node panics due to client process failure

If VCS daemon does not heartbeat with GAB within the configured timeout specified in VCS_GAB_TIMEOUT (default 30sec) environment variable, the node panics with a message similar to the following:

```
GAB Port h halting node due to client process failure at 3:109
```

GABs attempt (five retries) to kill the VCS daemon fails if VCS daemon is stuck in the kernel in an uninterruptible state or the system is heavily loaded that the VCS daemon cannot die with a SIGKILL.

Recommended Action:

- In case of performance issues, increase the value of the VCS_GAB_TIMEOUT environment variable to allow VCS more time to heartbeat.
- In case of a kernel problem, configure GAB to not panic but continue to attempt killing the VCS daemon.

  Do the following:

  - Run the following command on each node:

```
gabconfig -k
```

  - Add the “-k” option to the gabconfig command in the `/etc/gabtab` file:

```
gabconfig -c -k -n 6
```

- In case the problem persists, collect sar or similar output and crash dumps and contact Symantec support.

Troubleshooting VCS startup

This topic includes error messages associated with starting VCS (shown in bold text), and provides descriptions of each error and the recommended action.

"VCS:10622 local configuration missing"

The local configuration is missing.

Recommended Action: Start the VCS engine, HAD, on another system that has a valid configuration file. The system with the configuration error "pulls" the valid configuration from the other system.

Another method is to install the configuration file on the local system and force VCS to reread the configuration file. If the file appears valid, verify that is not an earlier version.

Type the following commands to verify the configuration:

```
# cd /etc/VRTSvcs/conf/config
# hacf -verify .
```

"VCS:10623 local configuration invalid"

The local configuration is invalid.

Recommended Action: Start the VCS engine, HAD, on another system that has a valid configuration file. The system with the configuration error "pulls" the valid configuration from the other system.

Another method is to correct the configuration file on the local system and force VCS to re-read the configuration file. If the file appears valid, verify that is not an earlier version.

Type the following commands to verify the configuration:

```
# cd /etc/VRTSvcs/conf/config
 # hacf -verify .
```

"VCS:11032 registration failed. Exiting"

GAB was not registered or has become unregistered.

Recommended Action: GAB is registered by the `gabconfig` command in the file /etc/gabtab. Verify that the file exists and that it contains the command `gabconfig -c`.

GAB can become unregistered if LLT is set up incorrectly. Verify that the configuration is correct in /etc/llttab. If the LLT configuration is incorrect, make the appropriate changes and reboot.

"Waiting for cluster membership."

This indicates that GAB may not be seeded. If this is the case, the command `gabconfig -a` does not show any members, and the following messages may appear on the console or in the event log.

```
GAB: Port a registration waiting for seed port membership
 GAB: Port h registration waiting for seed port membership
```

Troubleshooting Intelligent Monitoring Framework (IMF)

Review the following logs to isolate and troubleshoot Intelligent Monitoring Framework (IMF) related issues:

- System console log for the given operating system
- VCS engine Log : /var/VRTSvcs/log/engine_A.log
- Agent specific log : /var/VRTSvcs/log/*agentname*_A.log
- AMF in-memory trace buffer : View the contents using the `amfconfig -p dbglog` command

See "Enabling debug logs for IMF" on page 170.

See "Gathering IMF information for support analysis" on page 175.

Table 14-2 lists the most common issues for intelligent resource monitoring and provides instructions to troubleshoot and fix the issues.

Table 14-2 IMF-related issues and recommended actions

Issue	Description and recommended action
Intelligent resource monitoring has not reduced system utilization	If the system is busy even after intelligent resource monitoring is enabled, troubleshoot as follows: ■ Check the agent log file to see whether the `imf_init` agent function has failed. If the `imf_init` agent function has failed, then do the following: ■ Make sure that the AMF_START environment variable value is set to 1. ■ Make sure that the AMF module is loaded. ■ Make sure that the IMF attribute values are set correctly for the following attribute keys: ■ The value of the Mode key of the IMF attribute must be set to 1, 2, or 3. ■ The value of the MonitorFreq key of the IMF attribute must be be set to either 0 or a value greater than 0. For example, the value of the MonitorFreq key can be set to 0 for the Process agent. Refer to the appropriate agent documentation for configuration recommendations corresponding to the IMF-aware agent. Note that the IMF attribute can be overridden. So, if the attribute is set for individual resource, then check the value for individual resource. ■ Verify that the resources are registered with the AMF driver. Check the `amfstat` command output. ■ Check the LevelTwoMonitorFreq attribute settings for the agent. For example, Process agent must have this attribute value set to 0. Refer to the appropriate agent documentation for configuration recommendations corresponding to the IMF-aware agent.
Enabling the agent's intelligent monitoring does not provide immediate performance results	The actual intelligent monitoring for a resource starts only after a steady state is achieved. So, it takes some time before you can see positive performance effect after you enable IMF. This behavior is expected. For more information on when a steady state is reached, see the following topic:
Agent does not perform intelligent monitoring despite setting the IMF mode to 3	For the agents that use AMF driver for IMF notification, if intelligent resource monitoring has not taken effect, do the following: ■ Make sure that IMF attribute's Mode key value is set to three (3). ■ Review the the agent log to confirm that `imf_init()` agent registration with AMF has succeeded. AMF driver must be loaded before the agent starts because the agent registers with AMF at the agent startup time. If this was not the case, start the AMF module and restart the agent.

Table 14-2 IMF-related issues and recommended actions *(continued)*

Issue	Description and recommended action
AMF module fails to unload despite changing the IMF mode to 0	Even after you change the value of the Mode key to zero, the agent still continues to have a hold on the AMF driver until you kill the agent. To unload the AMF module, all holds on it must get released. If the AMF module fails to unload after changing the IMF mode value to zero, do the following: ■ Run the `amfconfig -Uof` command. This command forcefully removes all holds on the module and unconfigures it. ■ Then, unload AMF.
When you try to enable IMF for an agent, the `haimfconfig -enable -agent <agent_name>` command returns a message that IMF is enabled for the agent. However, when VCS and the respective agent is running, the `haimfconfig -display` command shows the status for agent_name as DISABLED.	A few possible reasons for this behavior are as follows: ■ The agent might require some manual steps to make it IMF-aware. Refer to the agent documentation for these manual steps. ■ The agent is a custom agent and is not IMF-aware. For information on how to make a custom agent IMF-aware, see the Veritas Cluster Server Agent Developer's Guide. ■ If the preceding steps do not resolve the issue, contact Symantec technical support.

Troubleshooting service groups

This topic cites the most common problems associated with bringing service groups online and taking them offline. Bold text provides a description of the problem. Recommended action is also included, where applicable.

VCS does not automatically start service group

VCS does not automatically start a failover service group if the VCS engine (HAD) in the cluster was restarted by the hashadow process.

This behavior prevents service groups from coming online automatically due to events such as GAB killing HAD because to high load, or HAD committing suicide to rectify unexpected error conditions.

System is not in RUNNING state

Recommended Action: Type `hasys -display` *`system`* to check the system status. When the system is not in running state, we may start VCS if there are no issues found.

Service group not configured to run on the system

The SystemList attribute of the group may not contain the name of the system.

Recommended Action: Use the output of the command `hagrp -display` *`service_group`* to verify the system name.

Service group not configured to autostart

If the service group is not starting automatically on the system, the group may not be configured to AutoStart, or may not be configured to AutoStart on that particular system.

Recommended Action: Use the output of the command `hagrp -display` *`service_group AutoStartList node_list`* to verify the values of the AutoStart and AutoStartList attributes.

Service group is frozen

Recommended Action: Use the output of the command `hagrp -display` *`service_group`* to verify the value of the Frozen and TFrozen attributes. Use the command `hagrp -unfreeze` to unfreeze the group. Note that VCS will not take a frozen service group offline.

Failover service group is online on another system

The group is a failover group and is online or partially online on another system.

Recommended Action: Use the output of the command `hagrp -display` *`service_group`* to verify the value of the State attribute. Use the command `hagrp -offline` to offline the group on another system.

A critical resource faulted

Output of the command `hagrp -display` *`service_group`* indicates that the service group has faulted.

Recommended Action: Use the command `hares -clear` to clear the fault.

Service group autodisabled

When VCS does not know the status of a service group on a particular system, it autodisables the service group on that system. Autodisabling occurs under the following conditions:

- When the VCS engine, HAD, is not running on the system.
  Under these conditions, all service groups that include the system in their SystemList attribute are autodisabled. This does not apply to systems that are powered off.
- When all resources within the service group are not probed on the system.

Recommended Action: Use the output of the command `hagrp -display` *`service_group`* to verify the value of the AutoDisabled attribute.

Warning: To bring a group online manually after VCS has autodisabled the group, make sure that the group is not fully or partially active on any system that has the AutoDisabled attribute set to 1 by VCS. Specifically, verify that all resources that may be corrupted by being active on multiple systems are brought down on the designated systems. Then, clear the AutoDisabled attribute for each system:

```
# hagrp -autoenable service_group -sys system
```

Service group is waiting for the resource to be brought online/taken offline.

Recommended Action: Review the IState attribute of all resources in the service group to locate which resource is waiting to go online (or which is waiting to be taken offline). Use the `hastatus` command to help identify the resource. See the engine and agent logs in /var/VRTSvcs/log for information on why the resource is unable to be brought online or be taken offline.

To clear this state, make sure all resources waiting to go online/offline do not bring themselves online/offline. Use the `hagrp -flush` command or the `hagrp -flush -force` command to clear the internal state of VCS. You can then bring the service group online or take it offline on another system.

For more information on the `hagrp -flush` and `hagrp -flush -force` commands.

Warning: Exercise caution when you use the -force option. It can lead to situations where a resource status is unintentionally returned as FAULTED. In the time interval that a resource transitions from 'waiting to go offline' to 'not waiting', if the agent has not completed the offline agent function, the agent may return the state of the resource as OFFLINE. VCS considers such unexpected offline of the resource as FAULT and starts recovery action that was not intended.

Service group is waiting for a dependency to be met.

Recommended Action: To see which dependencies have not been met, type `hagrp -dep` *`service_group`* to view service group dependencies, or `hares -dep` *`resource`* to view resource dependencies.

Service group not fully probed.

This occurs if the agent processes have not monitored each resource in the service group. When the VCS engine, HAD, starts, it immediately "probes" to find the initial state of all of resources. (It cannot probe if the agent is not returning a value.) A service group must be probed on all systems included in the SystemList attribute before VCS attempts to bring the group online as part of AutoStart. This ensures that even if the service group was online prior to VCS being brought up, VCS will not inadvertently bring the service group online on another system.

Recommended Action: Use the output of `hagrp -display` *`service_group`* to see the value of the ProbesPending attribute for the system's service group. (It should be zero.) To determine which resources are not probed, verify the local "Probed" attribute for each resource on the specified system. Zero means waiting for probe result, 1 means probed, and 2 means VCS not booted. See the engine and agent logs for information.

Troubleshooting resources

This topic cites the most common problems associated with bringing resources online and taking them offline. Bold text provides a description of the problem. Recommended action is also included, where applicable.

Service group brought online due to failover

VCS attempts to bring resources online that were already online on the failed system, or were in the process of going online. Each parent resource must wait for its child resources to be brought online before starting.

Recommended Action: Verify that the child resources are online.

Waiting for service group states

The state of the service group prevents VCS from bringing the resource online.

Recommended Action: Review the state of the service group.

Waiting for child resources

One or more child resources of parent resource are offline.

Recommended Action: Bring the child resources online first.

Waiting for parent resources

One or more parent resources are online.

Recommended Action: Take the parent resources offline first.

See "Troubleshooting resources" on page 190.

Waiting for resource to respond

The resource is waiting to come online or go offline, as indicated. VCS directed the agent to run an online entry point for the resource.

Recommended Action: Verify the resource's IState attribute. See the engine and agent logs in /var/VRTSvcs/engine_A.log and /var/VRTSvcs/*agent*_A.log for information on why the resource cannot be brought online.

Agent not running

The resource's agent process is not running.

Recommended Action: Use `hastatus -summary` to see if the agent is listed as faulted. Restart the agent:

```
# haagent -start resource_type -sys system
```

Invalid agent argument list.

The scripts are receiving incorrect arguments.

Recommended Action: Verify that the arguments to the scripts are correct. Use the output of `hares -display resource` to see the value of the ArgListValues attribute. If the ArgList attribute was dynamically changed, stop the agent and restart it.

To stop the agent:

◆
```
# haagent -stop resource_type -sys system
```

To restart the agent

◆
```
# haagent -start resource_type -sys system
```

The Monitor entry point of the disk group agent returns ONLINE even if the disk group is disabled

This is expected agent behavior. VCS assumes that data is being read from or written to the volumes and does not declare the resource as offline. This prevents potential data corruption that could be caused by the disk group being imported on two hosts.

You can deport a disabled disk group when all I/O operations are completed or when all volumes are closed. You can then reimport the disk group to the same system. Reimporting a disabled disk group may require a system reboot.

Note: A disk group is disabled if data including the kernel log, configuration copies, or headers in the private region of a significant number of disks is invalid or inaccessible. Volumes can perform read-write operations if no changes are required to the private regions of the disks.

Troubleshooting I/O fencing

The following sections discuss troubleshooting the I/O fencing problems. Review the symptoms and recommended solutions.

Node is unable to join cluster while another node is being ejected

A cluster that is currently fencing out (ejecting) a node from the cluster prevents a new node from joining the cluster until the fencing operation is completed. The following are example messages that appear on the console for the new node:

```
...VxFEN ERROR V-11-1-25 ... Unable to join running cluster
since cluster is currently fencing
a node out of the cluster.
```

If you see these messages when the new node is booting, the vxfen startup script on the node makes up to five attempts to join the cluster.

To manually join the node to the cluster when I/O fencing attempts fail

- If the vxfen script fails in the attempts to allow the node to join the cluster, restart vxfen driver with the command:

  ```
  # svcadm disable -t vxfen
  # svcadm enable vxfen
  ```

  If the command fails, restart the new node.

The vxfentsthdw utility fails when SCSI TEST UNIT READY command fails

While running the vxfentsthdw utility, you may see a message that resembles as follows:

```
Issuing SCSI TEST UNIT READY to disk reserved by other node
FAILED.
Contact the storage provider to have the hardware configuration
fixed.
```

The disk array does not support returning success for a `SCSI TEST UNIT READY` command when another host has the disk reserved using SCSI-3 persistent reservations. This happens with the Hitachi Data Systems 99XX arrays if bit 186 of the system mode option is not enabled.

Manually removing existing keys from SCSI-3 disks

Review the following procedure to remove specific registration and reservation keys created by another node from a disk.

Note: If you want to clear all the pre-existing keys, use the vxfenclearpre utility.

To remove the registration and reservation keys from disk

1. Create a file to contain the access names of the disks:

   ```
   # vi /tmp/disklist
   ```

   For example:

   ```
   /dev/rdsk/c1t0d11s2
   ```

2. Read the existing keys:

   ```
   # vxfenadm -s all -f /tmp/disklist
   ```

   The output from this command displays the key:

   ```
   Device Name: /dev/rdsk/c1t0d11s2

   Total Number Of Keys: 1
   key[0]:
   [Numeric Format]: 86,70,66,69,65,68,48,50
   [Character Format]: VFBEAD02
   [Node Format]: Cluster ID: 48813 Node ID: 2
     Node Name: unknown
   ```

3. If you know on which node the key (say A1) was created, log in to that node and enter the following command:

   ```
   # vxfenadm -x -kA1 -f /tmp/disklist
   ```

   The key A1 is removed.

4. If you do not know on which node the key was created, follow 5 through 7 to remove the key.

5. Register a second key “A2” temporarily with the disk:

   ```
   # vxfenadm -m -k A2 -f /tmp/disklist
   ```

   ```
   Registration completed for disk path /dev/rdsk/c1t0d11s2
   ```

6 Remove the first key from the disk by preempting it with the second key:

```
# vxfenadm -p -kA2 -f /tmp/disklist -vA1

key: A2------ prempted the key: A1------ on disk

/dev/rdsk/c1t0d11s2
```

7 Remove the temporary key registered in 5.

```
# vxfenadm -x -kA2 -f /tmp/disklist

Deleted the key : [A2------] from device /dev/rdsk/c1t0d11s2
```

No registration keys exist for the disk.

System panics to prevent potential data corruption

When a node experiences a split-brain condition and is ejected from the cluster, it panics and displays the following console message:

```
VXFEN:vxfen_plat_panic: Local cluster node ejected from cluster to
prevent potential data corruption.
```

A node experiences the split-brain condition when it loses the heartbeat with its peer nodes due to failure of all private interconnects or node hang. Review the behavior of I/O fencing under different scenarios and the corrective measures to be taken.

See "How I/O fencing works in different event scenarios" on page 195.

How I/O fencing works in different event scenarios

Table 14-3 describes how I/O fencing works to prevent data corruption in different failure event scenarios. For each event, review the corrective operator actions.

Table 14-3 I/O fencing scenarios

Event	Node A: What happens?	Node B: What happens?	Operator action
Both private networks fail.	Node A races for majority of coordination points. If Node A wins race for coordination points, Node A ejects Node B from the shared disks and continues.	Node B races for majority of coordination points. If Node B loses the race for the coordination points, Node B panics and removes itself from the cluster.	When Node B is ejected from cluster, repair the private networks before attempting to bring Node B back.
Both private networks function again after event above.	Node A continues to work.	Node B has crashed. It cannot start the database since it is unable to write to the data disks.	Restart Node B after private networks are restored.
One private network fails.	Node A prints message about an IOFENCE on the console but continues.	Node B prints message about an IOFENCE on the console but continues.	Repair private network. After network is repaired, both nodes automatically use it.
Node A hangs.	Node A is extremely busy for some reason or is in the kernel debugger. When Node A is no longer hung or in the kernel debugger, any queued writes to the data disks fail because Node A is ejected. When Node A receives message from GAB about being ejected, it panics and removes itself from the cluster.	Node B loses heartbeats with Node A, and races for a majority of coordination points. Node B wins race for coordination points and ejects Node A from shared data disks.	Repair or debug the node that hangs and reboot the node to rejoin the cluster.

Table 14-3 I/O fencing scenarios *(continued)*

Event	Node A: What happens?	Node B: What happens?	Operator action
Nodes A and B and private networks lose power. Coordination points and data disks retain power. Power returns to nodes and they restart, but private networks still have no power.	Node A restarts and I/O fencing driver (vxfen) detects Node B is registered with coordination points. The driver does not see Node B listed as member of cluster because private networks are down. This causes the I/O fencing device driver to prevent Node A from joining the cluster. Node A console displays: `Potentially a preexisting split brain. Dropping out of the cluster. Refer to the user documentation for steps required to clear preexisting split brain.`	Node B restarts and I/O fencing driver (vxfen) detects Node A is registered with coordination points. The driver does not see Node A listed as member of cluster because private networks are down. This causes the I/O fencing device driver to prevent Node B from joining the cluster. Node B console displays: `Potentially a preexisting split brain. Dropping out of the cluster. Refer to the user documentation for steps required to clear preexisting split brain.`	Resolve preexisting split-brain condition. See “Fencing startup reports preexisting split-brain” on page 200.

Table 14-3 I/O fencing scenarios *(continued)*

Event	Node A: What happens?	Node B: What happens?	Operator action
Node A crashes while Node B is down. Node B comes up and Node A is still down.	Node A is crashed.	Node B restarts and detects Node A is registered with the coordination points. The driver does not see Node A listed as member of the cluster. The I/O fencing device driver prints message on console: `Potentially a preexisting split brain. Dropping out of the cluster. Refer to the user documentation for steps required to clear preexisting split brain.`	Resolve preexisting split-brain condition. See "Fencing startup reports preexisting split-brain" on page 200.
The disk array containing two of the three coordination points is powered off. No node leaves the cluster membership	Node A continues to operate as long as no nodes leave the cluster.	Node B continues to operate as long as no nodes leave the cluster.	Power on the failed disk array so that subsequent network partition does not cause cluster shutdown, or replace coordination points.

Table 14-3 I/O fencing scenarios *(continued)*

Event	Node A: What happens?	Node B: What happens?	Operator action
The disk array containing two of the three coordination points is powered off. Node B gracefully leaves the cluster and the disk array is still powered off. Leaving gracefully implies a clean shutdown so that vxfen is properly unconfigured.	Node A continues to operate in the cluster.	Node B has left the cluster.	Power on the failed disk array so that subsequent network partition does not cause cluster shutdown, or replace coordination points.
The disk array containing two of the three coordination points is powered off. Node B abruptly crashes or a network partition occurs between node A and node B, and the disk array is still powered off.	Node A races for a majority of coordination points. Node A fails because only one of the three coordination points is available. Node A panics and removes itself from the cluster.	Node B has left cluster due to crash or network partition.	Power on the failed disk array and restart I/O fencing driver to enable Node A to register with all coordination points, or replace coordination points. See "Replacing defective disks when the cluster is offline" on page 204.

Cluster ID on the I/O fencing key of coordinator disk does not match the local cluster's ID

If you accidentally assign coordinator disks of a cluster to another cluster, then the fencing driver displays an error message similar to the following when you start I/O fencing:

```
000068 06:37:33 2bdd5845  0 ... 3066 0 VXFEN WARNING V-11-1-56
 Coordinator disk has key with cluster id 48813
 which does not match local cluster id 57069
```

The warning implies that the local cluster with the cluster ID 57069 has keys. However, the disk also has keys for cluster with ID 48813 which indicates that nodes from the cluster with cluster id 48813 potentially use the same coordinator disk.

You can run the following commands to verify whether these disks are used by another cluster. Run the following commands on one of the nodes in the local cluster. For example, on system01:

```
system01> # lltstat -C
57069

system01> # cat /etc/vxfentab
/dev/vx/rdmp/disk_7
/dev/vx/rdmp/disk_8
/dev/vx/rdmp/disk_9

system01> # vxfenadm -s /dev/vx/rdmp/disk_7
Reading SCSI Registration Keys...
Device Name: /dev/vx/rdmp/disk_7
Total Number Of Keys: 1
key[0]:
 [Numeric Format]:  86,70,48,49,52,66,48,48
 [Character Format]: VFBEAD00
 [Node Format]: Cluster ID: 48813 Node ID: 0 Node Name: unknown
```

Where *disk_7*, *disk_8*, and *disk_9* represent the disk names in your setup.

Recommended action: You must use a unique set of coordinator disks for each cluster. If the other cluster does not use these coordinator disks, then clear the keys using the vxfenclearpre command before you use them as coordinator disks in the local cluster.

Fencing startup reports preexisting split-brain

The vxfen driver functions to prevent an ejected node from rejoining the cluster after the failure of the private network links and before the private network links are repaired.

For example, suppose the cluster of system 1 and system 2 is functioning normally when the private network links are broken. Also suppose system 1 is the ejected system. When system 1 restarts before the private network links are restored, its membership configuration does not show system 2; however, when it attempts to register with the coordinator disks, it discovers system 2 is registered with them. Given this conflicting information about system 2, system 1 does not join the cluster and returns an error from vxfenconfig that resembles:

```
vxfenconfig: ERROR: There exists the potential for a preexisting
split-brain. The coordinator disks list no nodes which are in
the current membership. However, they also list nodes which are
not in the current membership.

I/O Fencing Disabled!
```

Also, the following information is displayed on the console:

```
<date> <system name> vxfen: WARNING: Potentially a preexisting
<date> <system name> split-brain.
<date> <system name> Dropping out of cluster.
<date> <system name> Refer to user documentation for steps
<date> <system name> required to clear preexisting split-brain.
<date> <system name>
<date> <system name> I/O Fencing DISABLED!
<date> <system name>
<date> <system name> gab: GAB:20032: Port b closed
```

However, the same error can occur when the private network links are working and both systems go down, system 1 restarts, and system 2 fails to come back up. From the view of the cluster from system 1, system 2 may still have the registrations on the coordination points.

Assume the following situations to understand preexisting split-brain in server-based fencing:

- There are three CP servers acting as coordination points. One of the three CP servers then becomes inaccessible. While in this state, one client node leaves the cluster, whose registration cannot be removed from the inaccessible CP server. When the inaccessible CP server restarts, it has a stale registration from the node which left the VCS. In this case, no new nodes can join the cluster. Each node that attempts to join the cluster gets a list of registrations from the CP server. One CP server includes an extra registration (of the node which left earlier). This makes the joiner node conclude that there exists a preexisting split-brain between the joiner node and the node which is represented by the stale registration.
- All the client nodes have crashed simultaneously, due to which fencing keys are not cleared from the CP servers. Consequently, when the nodes restart, the vxfen configuration fails reporting preexisting split brain.

These situations are similar to that of preexisting split-brain with coordinator disks, where you can solve the problem running the `vxfenclearpre` command. A similar solution is required in server-based fencing using the `cpsadm` command.

See “Clearing preexisting split-brain condition” on page 202.

Clearing preexisting split-brain condition

Review the information on how the VxFEN driver checks for preexisting split-brain condition.

See "Fencing startup reports preexisting split-brain" on page 200.

Table 14-4 describes how to resolve a preexisting split-brain condition depending on the scenario you have encountered:

Table 14-4 Recommened solution to clear pre-existing split-brain condition

Scenario	Solution
Actual potential split-brain condition—system 2 is up and system 1 is ejected	**1** Determine if system1 is up or not. **2** If system 1 is up and running, shut it down and repair the private network links to remove the split-brain condition. **3** Restart system 1.
Apparent potential split-brain condition—system 2 is down and system 1 is ejected (Disk-based fencing is configured)	**1** Physically verify that system 2 is down. Verify the systems currently registered with the coordination points. Use the following command for coordinator disks: `# vxfenadm -s all -f /etc/vxfentab` The output of this command identifies the keys registered with the coordinator disks. **2** Clear the keys on the coordinator disks as well as the data disks in all shared disk groups using the `vxfenclearpre` command. The command removes SCSI-3 registrations and reservations. **3** Make any necessary repairs to system 2. **4** Restart system 2.

Table 14-4 Recommened solution to clear pre-existing split-brain condition *(continued)*

Scenario	Solution

Apparent potential split-brain condition—system 2 is down and system 1 is ejected
(Server-based fencing is configured)
	1 Physically verify that system 2 is down.
	Verify the systems currently registered with the coordination points.
	Use the following command for CP servers:
	`# cpsadm -s cp_server -a list_membership`
	`-c cluster_name`
	where *cp_server* is the virtual IP address or virtual hostname on which CP server is configured, and *cluster_name* is the VCS name for the SFHA Solutions cluster (application cluster).
	The command lists the systems registered with the CP server.
	2 Clear the keys on the CP servers using the `cpsadm` command. The `cpsadm` command clears a registration on a CP server:
	`# cpsadm -s cp_server -a unreg_node`
	`-c cluster_name -n nodeid`
	where *cp_server* is the virtual IP address or virtual hostname on which the CP server is listening, *cluster_name* is the VCS name for the VCS, and *nodeid* specifies the node id of VCS node. Ensure that fencing is not already running on a node before clearing its registration on the CP server.
	After removing all stale registrations, the joiner node will be able to join the cluster.
	3 Make any necessary repairs to system 2.
	4 Restart system 2.

Registered keys are lost on the coordinator disks

If the coordinator disks lose the keys that are registered, the cluster might panic when a cluster reconfiguration occurs.

To refresh the missing keys

◆ Use the vxfenswap utility to replace the coordinator disks with the same disks. The vxfenswap utility registers the missing keys during the disk replacement.

Replacing defective disks when the cluster is offline

If the disk becomes defective or inoperable and you want to switch to a new diskgroup in a cluster that is offline, then perform the following procedure.

In a cluster that is online, you can replace the disks using the vxfenswap utility.

Review the following information to replace coordinator disk in the coordinator disk group, or to destroy a coordinator disk group.

Note the following about the procedure:

- When you add a disk, add the disk to the disk group vxfencoorddg and retest the group for support of SCSI-3 persistent reservations.
- You can destroy the coordinator disk group such that no registration keys remain on the disks. The disks can then be used elsewhere.

To replace a disk in the coordinator disk group when the cluster is offline

1 Log in as superuser on one of the cluster nodes.

2 If VCS is running, shut it down:

```
# hastop -all
```

Make sure that the port h is closed on all the nodes. Run the following command to verify that the port h is closed:

```
# gabconfig -a
```

3 Stop the VCSMM driver on each node:

```
# svcadm disable -t vcsmm
```

4 Stop I/O fencing on each node:

```
# svcadm disable -t vxfen
```

This removes any registration keys on the disks.

5 Import the coordinator disk group. The file /etc/vxfendg includes the name of the disk group (typically, vxfencoorddg) that contains the coordinator disks, so use the command:

```
# vxdg -tfC import `cat /etc/vxfendg`
```

where:

-t specifies that the disk group is imported only until the node restarts.

-f specifies that the import is to be done forcibly, which is necessary if one or more disks is not accessible.

-C specifies that any import locks are removed.

6 To remove disks from the disk group, use the VxVM disk administrator utility, vxdiskadm.

You may also destroy the existing coordinator disk group. For example:

- Verify whether the coordinator attribute is set to on.

  ```
  # vxdg list vxfencoorddg |  grep flags: | grep coordinator
  ```

- Destroy the coordinator disk group.

  ```
  # vxdg -o coordinator destroy vxfencoorddg
  ```

7 Add the new disk to the node and initialize it as a VxVM disk.

Then, add the new disk to the vxfencoorddg disk group:

- If you destroyed the disk group in step 6, then create the disk group again and add the new disk to it.
- If the diskgroup already exists, then add the new disk to it.

  ```
  # vxdg -g vxfencoorddg -o coordinator adddisk disk_name
  ```

8 Test the recreated disk group for SCSI-3 persistent reservations compliance.

9 After replacing disks in a coordinator disk group, deport the disk group:

```
# vxdg deport `cat /etc/vxfendg`
```

10 On each node, start the I/O fencing driver:

```
# svcadm enable vxfen
```

11 On each node, start the VCSMM driver:

```
# svcadm enable vcsmm
```

12 Verify that the I/O fencing module has started and is enabled.

```
# gabconfig -a
```

Make sure that port b membership exists in the output for all nodes in the cluster.

Make sure that port b and port o memberships exist in the output for all nodes in the cluster.

```
# vxfenadm -d
```

Make sure that I/O fencing mode is not disabled in the output.

13 If necessary, restart VCS on each node:

```
# hastart
```

The vxfenswap utility exits if rcp or scp commands are not functional

The vxfenswap utility displays an error message if rcp or scp commands are not functional.

To recover the vxfenswap utility fault

◆ Verify whether the rcp or scp functions properly.

Make sure that you do not use echo or cat to print messages in the .bashrc file for the nodes.

If the vxfenswap operation is unsuccessful, use the `vxfenswap -a cancel` command if required to roll back any changes that the utility made.

Troubleshooting CP server

All CP server operations and messages are logged in the /var/VRTScps/log directory in a detailed and easy to read format. The entries are sorted by date and time. The logs can be used for troubleshooting purposes or to review for any possible security issue on the system that hosts the CP server.

The following files contain logs and text files that may be useful in understanding and troubleshooting a CP server:

- /var/VRTScps/log/cpserver_[ABC].log

- /var/VRTSvcs/log/vcsauthserver.log (Security related)
- If the vxcpserv process fails on the CP server, then review the following diagnostic files:
  - /var/VRTScps/diag/FFDC_CPS_*pid*_vxcpserv.log
  - /var/VRTScps/diag/stack_*pid*_vxcpserv.txt

Note: If the vxcpserv process fails on the CP server, these files are present in addition to a core file. VCS restarts vxcpserv process automatically in such situations.

The file /var/VRTSvcs/log/vxfen/vxfend_[ABC].log contains logs that may be useful in understanding and troubleshooting fencing-related issues on a VCS (client cluster) node.

See "Troubleshooting issues related to the CP server service group" on page 207.

See "Checking the connectivity of CP server" on page 207.

See "Issues during fencing startup on VCS nodes set up for server-based fencing" on page 208.

See "Issues during online migration of coordination points" on page 209.

Troubleshooting issues related to the CP server service group

If you cannot bring up the CPSSG service group after the CP server configuration, perform the following steps:

- Verify that the CPSSG service group and its resources are valid and properly configured in the VCS configuration.
- Check the VCS engine log (`/var/VRTSvcs/log/engine_[ABC].log`) to see if any of the CPSSG service group resources are FAULTED.
- Review the sample dependency graphs to make sure the required resources are configured correctly.

Checking the connectivity of CP server

You can test the connectivity of CP server using the `cpsadm` command.

You must have set the environment variables CPS_USERNAME and CPS_DOMAINTYPE to run the `cpsadm` command on the VCS (client cluster) nodes.

To check the connectivity of CP server

- Run the following command to check whether a CP server is up and running at a process level:

  ```
  # cpsadm -s cp_server -a ping_cps
  ```

  where *cp_server* is the virtual IP address or virtual hostname on which the CP server is listening.

Troubleshooting server-based fencing on the SFHA Solutions cluster nodes

The file /var/VRTSvcs/log/vxfen/vxfend_[ABC].log contains logs files that may be useful in understanding and troubleshooting fencing-related issues on a SFHA Solutions cluster (application cluster) node.

Issues during fencing startup on VCS nodes set up for server-based fencing

Table 14-5 Fencing startup issues on VCS (client cluster) nodes

Issue	Description and resolution
`cpsadm` command on the VCS gives connection error	If you receive a connection error message after issuing the `cpsadm` command on the VCS, perform the following actions: ■ Ensure that the CP server is reachable from all the VCS nodes. ■ Check that the VCS nodes use the correct CP server virtual IP or virtual hostname and the correct port number. Check the /etc/vxfenmode file. ■ Ensure that the running CP server is using the same virtual IP/virtual hostname and port number.
Authorization failure	Authorization failure occurs when the CP server's nodes or users are not added in the CP server configuration. Therefore, fencing on the VCS (client cluster) node is not allowed to access the CP server and register itself on the CP server. Fencing fails to come up if it fails to register with a majority of the coordination points. To resolve this issue, add the CP server node and user in the CP server configuration and restart fencing.

Table 14-5 Fencing startup issues on VCS (client cluster) nodes *(continued)*

Issue	Description and resolution
Authentication failure	If you had configured secure communication between the CP server and the VCS (client cluster) nodes, authentication failure can occur due to the following causes: ■ Symantec Product Authentication Services (AT) is not properly configured on the CP server and/or the VCS. ■ The CP server and the VCS nodes use different root brokers, and trust is not established between the authentication brokers:

Issues during online migration of coordination points

During online migration of coordination points using the vxfenswap utility, the operation is automatically rolled back if a failure is encountered during validation of coordination points from any of the cluster nodes.

Validation failure of the new set of coordination points can occur in the following circumstances:

- The /etc/vxfenmode.test file is not updated on all the VCS nodes, because new coordination points on the node were being picked up from an old /etc/vxfenmode.test file. The /etc/vxfenmode.test file must be updated with the current details. If the /etc/vxfenmode.test file is not present, vxfenswap copies configuration for new coordination points from the /etc/vxfenmode file.
- The coordination points listed in the /etc/vxfenmode file on the different VCS nodes are not the same. If different coordination points are listed in the /etc/vxfenmode file on the cluster nodes, then the operation fails due to failure during the coordination point snapshot check.
- There is no network connectivity from one or more VCS nodes to the CP server(s).
- Cluster, nodes, or users for the VCS nodes have not been added on the new CP servers, thereby causing authorization failure.

Vxfen service group activity after issuing the vxfenswap command

The Coordination Point agent reads the details of coordination points from the `vxfenconfig -l` output and starts monitoring the registrations on them.

Thus, during vxfenswap, when the vxfenmode file is being changed by the user, the Coordination Point agent does not move to FAULTED state but continues monitoring the old set of coordination points.

As long as the changes to vxfenmode file are not committed or the new set of coordination points are not reflected in `vxfenconfig -l` output, the Coordination Point agent continues monitoring the old set of coordination points it read from `vxfenconfig -l` output in every monitor cycle.

The status of the Coordination Point agent (either ONLINE or FAULTED) depends upon the accessibility of the coordination points, the registrations on these coordination points, and the fault tolerance value.

When the changes to vxfenmode file are committed and reflected in the `vxfenconfig -l` output, then the Coordination Point agent reads the new set of coordination points and proceeds to monitor them in its new monitor cycle.

Troubleshooting notification

Occasionally you may encounter problems when using VCS notification. This section cites the most common problems and the recommended actions. Bold text provides a description of the problem.

Notifier is configured but traps are not seen on SNMP console.

Recommended Action: Verify the version of SNMP traps supported by the console: VCS notifier sends SNMP v2.0 traps. If you are using HP OpenView Network Node Manager as the SNMP, verify events for VCS are configured using xnmevents. You may also try restarting the OpenView daemon (ovw) if, after merging VCS events in vcs_trapd, the events are not listed in the OpenView Network Node Manager Event configuration.

By default, notifier assumes the community string is public. If your SNMP console was configured with a different community, reconfigure it according to the notifier configuration. See the *Veritas Cluster Server Bundled Agents Reference Guide* for more information on NotifierMngr.

Troubleshooting and recovery for global clusters

This topic describes the concept of disaster declaration and provides troubleshooting tips for configurations using global clusters.

Disaster declaration

When a cluster in a global cluster transitions to the FAULTED state because it can no longer be contacted, failover executions depend on whether the cause was due to a split-brain, temporary outage, or a permanent disaster at the remote cluster.

If you choose to take action on the failure of a cluster in a global cluster, VCS prompts you to declare the type of failure.

- *Disaster*, implying permanent loss of the primary data center
- *Outage*, implying the primary may return to its current form in some time
- *Disconnect*, implying a split-brain condition; both clusters are up, but the link between them is broken
- *Replica*, implying that data on the takeover target has been made consistent from a backup source and that the RVGPrimary can initiate a takeover when the service group is brought online. This option applies to VVR environments only.

You can select the groups to be failed over to the local cluster, in which case VCS brings the selected groups online on a node based on the group's FailOverPolicy attribute. It also marks the groups as being offline in the other cluster. If you do not select any service groups to fail over, VCS takes no action except implicitly marking the service groups as offline on the downed cluster.

Lost heartbeats and the inquiry mechanism

The loss of internal and all external heartbeats between any two clusters indicates that the remote cluster is faulted, or that all communication links between the two clusters are broken (a wide-area split-brain).

VCS queries clusters to confirm the remote cluster to which heartbeats have been lost is truly down. This mechanism is referred to as inquiry. If in a two-cluster configuration a connector loses all heartbeats to the other connector, it must consider the remote cluster faulted. If there are more than two clusters and a connector loses all heartbeats to a second cluster, it queries the remaining connectors before declaring the cluster faulted. If the other connectors view the cluster as running, the querying connector transitions the cluster to the UNKNOWN state, a process that minimizes false cluster faults. If all connectors report that the cluster is faulted, the querying connector also considers it faulted and transitions the remote cluster state to FAULTED.

VCS alerts

VCS alerts are identified by the alert ID, which is comprised of the following elements:

- `alert_type`—The type of the alert

- `cluster`—The cluster on which the alert was generated
- `system`—The system on which this alert was generated
- `object`—The name of the VCS object for which this alert was generated. This could be a cluster or a service group.

Alerts are generated in the following format:

```
alert_type-cluster-system-object
```

For example:

```
GNOFAILA-Cluster1-oracle_grp
```

This is an alert of type GNOFAILA generated on cluster Cluster1 for the service group oracle_grp.

Types of alerts

VCS generates the following types of alerts.

- CFAULT—Indicates that a cluster has faulted
- GNOFAILA—Indicates that a global group is unable to fail over within the cluster where it was online. This alert is displayed if the ClusterFailOverPolicy attribute is set to Manual and the wide-area connector (wac) is properly configured and running at the time of the fault.
- GNOFAIL—Indicates that a global group is unable to fail over to any system within the cluster or in a remote cluster.

  Some reasons why a global group may not be able to fail over to a remote cluster:

  - The ClusterFailOverPolicy is set to either Auto or Connected and VCS is unable to determine a valid remote cluster to which to automatically fail the group over.
  - The ClusterFailOverPolicy attribute is set to Connected and the cluster in which the group has faulted cannot communicate with one ore more remote clusters in the group's ClusterList.
  - The wide-area connector (wac) is not online or is incorrectly configured in the cluster in which the group has faulted

Managing alerts

Alerts require user intervention. You can respond to an alert in the following ways:

- If the reason for the alert can be ignored, use the Alerts dialog box in the Java console or the `haalert` command to delete the alert. You must provide a comment as to why you are deleting the alert; VCS logs the comment to engine log.
- Take an action on administrative alerts that have actions associated with them.
- VCS deletes or negates some alerts when a negating event for the alert occurs.

An administrative alert will continue to live if none of the above actions are performed and the VCS engine (HAD) is running on at least one node in the cluster. If HAD is not running on any node in the cluster, the administrative alert is lost.

Actions associated with alerts

This section describes the actions you can perform on the following types of alerts:

- CFAULT—When the alert is presented, clicking **Take Action** guides you through the process of failing over the global groups that were online in the cluster before the cluster faulted.
- GNOFAILA—When the alert is presented, clicking **Take Action** guides you through the process of failing over the global group to a remote cluster on which the group is configured to run.
- GNOFAIL—There are no associated actions provided by the consoles for this alert

Negating events

VCS deletes a CFAULT alert when the faulted cluster goes back to the running state

VCS deletes the GNOFAILA and GNOFAIL alerts in response to the following events:

- The faulted group's state changes from FAULTED to ONLINE.
- The group's fault is cleared.
- The group is deleted from the cluster where alert was generated.

Concurrency violation at startup

VCS may report a concurrency violation when you add a cluster to the ClusterList of the service group. A concurrency violation means that the service group is online on two nodes simultaneously.

Recommended Action: Verify the state of the service group in each cluster before making the service group global.

Troubleshooting the steward process

When you start the steward, it blocks the command prompt and prints messages to the standard output. To stop the steward, run the following command from a different command prompt of the same system:

If the steward is running in secure mode: `steward -stop - secure`

If the steward is not running in secure mode: `steward -stop`

In addition to the standard output, the steward can log to its own log files:

- steward_A.log
- steward-err_A.log

Use the tststew utility to verify that:

- The steward process is running
- The steward process is sending the right response

Troubleshooting licensing

This section cites problems you may encounter with VCS licensing. It provides instructions on how to validate license keys and lists the error messages associated with licensing.

Validating license keys

The `installvcs` script handles most license key validations. However, if you install a VCS key outside of `installvcs` (using `vxlicinst`, for example), you can validate the key using the procedure described below.

1 The `vxlicinst` command handles some of the basic validations:

node lock: Ensures that you are installing a node-locked key on the correct system

demo hard end date: Ensures that you are not installing an expired demo key

2 Run the `vxlicrep` command to make sure a VCS key is installed on the system. The output of the command resembles:

```
License Key      = XXXX-XXXX-XXXX-XXXX-XXXX-XXXX
Product Name     = Veritas Cluster Server
License Type     = PERMANENT
OEM ID           = 4095
Features :

Platform                    = Solaris

Version                     = 6.0
Tier                        = Unused
Reserved                    = 0
Mode                        = VCS
Global Cluster Option       = Enabled
```

3 Look for the following in the command output:

Make sure the Product Name lists the name of your purchased component, for example, Veritas Cluster Server. If the command output does not return the product name, you do not have a VCS key installed.

If the output shows the License Type for a VCS key as DEMO, ensure that the Demo End Date does not display a past date.

Make sure the *Mode* attribute displays the correct value.

If you have purchased a license key for the Global Cluster Option, make sure its status is Enabled.

4 Start VCS. If HAD rejects a license key, see the licensing error message at the end of the engine_A log file.

Licensing error messages

This section lists the error messages associated with licensing. These messages are logged to the file /var/VRTSvcs/log/engine_A.log.

[Licensing] Insufficient memory to perform operation

The system does not have adequate resources to perform licensing operations.

[Licensing] No valid VCS license keys were found

No valid VCS keys were found on the system.

[Licensing] Unable to find a valid base VCS license key

No valid base VCS key was found on the system.

[Licensing] License key cannot be used on this OS platform

This message indicates that the license key was meant for a different platform. For example, a license key meant for Windows is used on Solaris platform.

[Licensing] VCS evaluation period has expired

The VCS base demo key has expired

[Licensing] License key can not be used on this system

Indicates that you have installed a key that was meant for a different system (i.e. node-locked keys)

[Licensing] Unable to initialize the licensing framework

This is a VCS internal message. Call Veritas Technical Support.

[Licensing] QuickStart is not supported in this release

VCS QuickStart is not supported in this version of VCS.

[Licensing] Your evaluation period for the feature has expired. This feature will not be enabled the next time VCS starts

The evaluation period for the specified VCS feature has expired.

Index