

Veritas Storage Foundation™ for Oracle® RAC Administrator's Guide

Solaris

5.0 Maintenance Pack 3



Veritas Storage Foundation for Oracle RAC Administrator's 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 5.0MP3

Documentation version 5.0MP3.0

Legal Notice

Copyright © 2008 Symantec Corporation. All rights reserved.

Symantec, the Symantec Logo, Veritas, Veritas Storage Foundation, and FlashSnap are trademarks or registered trademarks of Symantec Corporation or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.

This Symantec product may contain third party software for which Symantec is required to provide attribution to the third party ("Third Party Programs"). Some of the Third Party Programs are available under open source or free software licenses. The License Agreement accompanying the Software does not alter any rights or obligations you may have under those open source or free software licenses. Please see the Third Party Legal Notice Appendix to this Documentation or TPIP ReadMe File accompanying this Symantec product for more information on the Third Party Programs.

The product described in this document is distributed under licenses restricting its use, copying, distribution, and decompilation/reverse engineering. No part of this document may be reproduced in any form by any means without prior written authorization of Symantec Corporation and its licensors, if any.

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
20330 Stevens Creek Blvd.
Cupertino, CA 95014

<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 maintenance 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 Web-based support that provides rapid response and up-to-the-minute information
- Upgrade assurance that delivers automatic software upgrade protection
- Global support that is available 24 hours a day, 7 days a week
- Advanced features, including Account Management Services

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

www.symantec.com/techsupp/

Contacting Technical Support

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

www.symantec.com/business/support/assistance_care.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:

<https://licensing.symantec.com>

Customer service

Customer service information is available at the following URL:

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

Customer Service is available to assist with 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 maintenance 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

Maintenance agreement resources

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

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

Additional enterprise services

Symantec offers a comprehensive set of services that allow you to maximize your investment in Symantec products and to develop your knowledge, expertise, and global insight, which enable you to manage your business risks proactively.

Enterprise services that are available include the following:

Symantec Early Warning Solutions	These solutions provide early warning of cyber attacks, comprehensive threat analysis, and countermeasures to prevent attacks before they occur.
Managed Security Services	These services remove the burden of managing and monitoring security devices and events, ensuring rapid response to real threats.
Consulting Services	Symantec Consulting Services provide on-site technical expertise from Symantec and its trusted partners. Symantec Consulting Services offer a variety of prepackaged and customizable options that include assessment, design, implementation, monitoring, and management capabilities. Each is focused on establishing and maintaining the integrity and availability of your IT resources.
Educational Services	Educational Services provide a full array of technical training, security education, security certification, and awareness communication programs.

To access more information about Enterprise services, please visit our Web site at the following URL:

www.symantec.com

Select your country or language from the site index.

Contents

Technical Support	4
Section 1 SF Oracle RAC concepts and administration	17
Chapter 1 Storage Foundation for Oracle RAC overview	19
About SF Oracle RAC	19
How SF Oracle RAC works (high-level perspective)	20
Component products and processes of SF Oracle RAC	23
Communication infrastructure	24
Cluster interconnect communication channel	26
Low-level communication: port relationship between GAB and processes	28
Cluster Volume Manager (CVM)	29
Cluster File System (CFS)	31
Oracle Disk Manager	34
Veritas Cluster Server	35
Oracle RAC	36
RAC extensions	38
About preventing data corruption with I/O fencing	40
About SCSI-3 Persistent Reservations	41
About I/O fencing operations	42
About I/O fencing communication	42
Supported SF Oracle RAC options	42
Managing databases using SF Oracle RAC	43
Securing a cluster using SF Oracle RAC	43
Managing multiple SF Oracle RAC clusters using a common Web console server	44
Replicating data to a remote site using SF Oracle RAC	44
Chapter 2 Configuration best practices	47
About configuration best practices	47
Cluster configuration	48
Basic clustered shared storage cluster	48

Clusters running in secure mode	49
Campus, or metropolitan, shared storage cluster	49
Global cluster	49
Oracle configuration best practices	50
Database storage configuration best practices	50
Database snapshot and backup options	51
Database FlashSnap for cloning	52
Storage Checkpoints for recovery	52
Database storage optimization options	53

Chapter 3

Administering SF Oracle RAC and its components

Administering SF Oracle RAC	55
Setting the PATH variable	56
Setting the MANPATH variable	57
Stopping SF Oracle RAC on a single node	57
Stopping and starting LLT and GAB	60
Adding LLT links to increase capacity	61
Turning on the debug messages for the kernel drivers	61
Administering VCS	62
Viewing available Veritas devices and drivers	62
Loading Veritas drivers into memory	63
Configuring VCS to start Oracle with a specified Pfile	63
Verifying VCS configuration	64
Starting and stopping VCS	64
Rebooting nodes in a cluster	64
Administering I/O fencing	64
About I/O fencing	65
How I/O fencing works in different event scenarios	67
About I/O fencing utilities	71
About vxfststhdw utility	71
About vxfenadm utility	80
About vxfenclearpre utility	82
About vxfenswap utility	84
Administering CFS	92
Adding CFS file systems to VCS configuration	92
Using cfsmount to mount CFS file systems	92
Resizing CFS file systems	93
Verifying the status of CFS file systems	93
Verifying CFS port	94
CFS agent log files	94
Administering CVM	94

Listing all the CVM shared disks	95
Establishing CVM cluster membership manually	95
Manually importing a shared disk group	95
Manually deporting a shared disk group	95
Manually starting shared volumes	95
Evaluating the state of CVM ports	96
Verifying if CVM is running in an SF Oracle RAC cluster	96
Verifying CVM membership state	96
Verifying state of CVM shared disk groups	97
Verifying activation mode	97
CVM log files	97
Administering Oracle	98
Creating a database	98
Increasing swap space for Oracle	99
Stopping CRS	99
Determining CRS object status	99
Unregistering Oracle database instances	100
Configuring virtual IP addresses for CRS	102
Configuring Oracle group to start and stop CRS objects	102
Configuring listeners	103
Starting or stopping Oracle listener	103
Using srvctl to manage node applications	103
Starting Oracle Enterprise Manager agent	104
Clearing Oracle group faults	104
Starting and stopping Oracle service groups	104
Administering ODM	105
Verifying the ODM port	105
Starting ODM	105

Section 2 Managing a database using SF Oracle RAC 107

Chapter 4 Configuring and managing the repository database for Oracle	109
About the repository database	109
Setting up the repository database using the sfua_db_config script	110
Creating and configuring the repository database	110
Checking the SFDB repository with sfua_db_config	113
Setting administrative permissions	113
Runtime management tasks for the repository	114

Backing up and restoring the repository with sfua_rept_adm	114
Monitoring free space for the SFDB repository	117
Adding a new system to a SF Oracle RAC configuration	118

Chapter 5

Using Storage Checkpoints and Storage Rollback	121
About Storage Checkpoints and Storage Rollback in SF Oracle RAC	121
Using Storage Checkpoints and Storage Rollback for backup and restore	122
Storage Checkpoints	122
Storage Rollbacks	122
Storage Checkpoints and Storage Rollback process	122
Determining space requirements for Storage Checkpoints	124
Storage Checkpoint Performance	125
Backing up and recovering the database using Storage Checkpoints	126
Specify the Storage Checkpoint option	127
Verifying a Storage Checkpoint	128
Backing up using a Storage Checkpoint	131
Recovering a database using a Storage Checkpoint	132
Guidelines for Oracle recovery	134
Back up all control files before Storage Rollback	134
Ensure that the control files are not rolled back	135
Ensure that all archived redo logs are available	135
Media recovery procedures	136
Using the Storage Checkpoint Command Line Interface (CLI)	137
Commands Overview	137
Command Line Interface examples	139
Prerequisites	140
Creating or updating the repository using dbed_update	140
Creating Storage Checkpoints using dbed_ckptcreate	140
Displaying Storage Checkpoints using dbed_ckptdisplay	142
Mounting Storage Checkpoints using dbed_ckptmount	145
Unmounting Storage Checkpoints using dbed_ckptumount	146
Creating and working with Storage Checkpoint allocation policies using dbed_ckptpolicy	147
Performing Storage Rollback using dbed_ckptrollback	150
Removing Storage Checkpoints using dbed_ckptremove	152
Cloning the Oracle instance using dbed_clonedb	152

Chapter 6	Using Database FlashSnap for backup and off-host processing	157
	About Veritas Database FlashSnap	157
	Solving typical database problems with Database FlashSnap	158
	Database FlashSnap applications	158
	Using Database FlashSnap	159
	Using Database FlashSnap commands	161
	Using the Database FlashSnap online option	161
	Planning to use Database FlashSnap	162
	Selecting the snapshot mode	162
	Preparing hosts and storage for Database FlashSnap	162
	Setting up hosts	162
	Creating a snapshot mirror of a volume or volume set used by the database	163
	Upgrading existing volumes to use Veritas Volume Manager 5.0	170
	About creating database snapshots	177
	Online database snapshots	177
	Tasks before creating a snapshot	178
	Creating a snapshot	179
	Tasks after creating a snapshot	180
	FlashSnap commands	182
	Creating a snapplan (dbed_vmchecksnap)	182
	Validating a snapplan (dbed_vmchecksnap)	188
	Displaying, copying, and removing a snapplan (dbed_vmchecksnap)	191
	Creating a snapshot (dbed_vmsnap)	194
	Backing up the database from snapshot volumes (dbed_vmclonedb)	197
	Cloning a database (dbed_vmclonedb)	201
	Resynchronizing the snapshot to your database	211
	Removing a snapshot volume	212
Chapter 7	Using Database Dynamic Storage Tiering	215
	About Database Dynamic Storage Tiering	215
	Database Dynamic Storage Tiering building blocks	216
	Database Dynamic Storage Tiering in a High Availability (HA) environment	218
	Dynamic Storage Tiering policy management	218
	Relocating files	219
	Relocating tablespaces	219
	Relocating table partitions	220

Using preset policies	221
Configuring Database Dynamic Storage Tiering	222
Database Dynamic Storage Tiering command requirements	222
Defining database parameters	223
Setting up storage classes	225
Converting a VxFS file system to a VxFS multi-volume file system	226
Classifying volumes into a storage class	227
Displaying free space on your storage class	228
Adding new volumes to a storage class	229
Removing volumes from a storage class	229
Extent balancing in a database environment	230
Extent balancing file system	230
Creating an extent balanced file system	231
Running Database Dynamic Storage Tiering reports	232
Viewing modified allocation policies	233
Viewing audit reports	233
Oracle Database Dynamic Storage Tiering use cases	233
Migrating partitioned data and tablespaces	233
Scheduling the relocation of archive and Flashback logs	236

Section 3 Performance and troubleshooting 239

Chapter 8 Investigating I/O performance using storage mapping	241
About Storage Mapping in SF Oracle RAC	241
Understanding Storage Mapping	241
Verifying Veritas Storage Mapping set up	243
Using the vxstorage_stats command	243
Displaying Storage Mapping information	244
Displaying I/O statistics information	245
Using the dbed_analyzer command	246
Obtaining Storage Mapping information for a list of tablespaces	247
About arrays for Storage Mapping and statistics	249
Oracle File Mapping (ORAMAP)	250
Mapping components	250
Storage Mapping views	251
Verifying Oracle file mapping set up	252
Enabling Oracle file mapping	252
Accessing dynamic performance views	253

Chapter 9

Troubleshooting SF Oracle RAC	257
About troubleshooting SF Oracle RAC	257
Running scripts for engineering support analysis	258
Troubleshooting tips	258
Troubleshooting I/O fencing	261
SCSI reservation errors during bootup	261
The vxfststdhw utility fails when SCSI TEST UNIT READY command fails	261
Node is unable to join cluster while another node is being ejected	262
System panics to prevent potential data corruption	262
Clearing keys after split brain using vxfencearp command	264
Registered keys are lost on the coordinator disks	264
Replacing defective disks when the cluster is offline	264
The vxfseswap utility faults when echo or cat is used in .bashrc file	267
Troubleshooting CVM	267
Shared disk group cannot be imported	267
Error importing shared disk groups	268
Unable to start CVM	268
CVMVolDg not online even though CVMCluster is online	268
Shared disks not visible	268
Troubleshooting the repository database	269
sfua_db_config script command options	269
Switching the repository database from one node to another	270
Troubleshooting Database Dynamic Storage Tiering commands	271
Troubleshooting CFS	272
Incorrect order in root user's <library> path	272
Troubleshooting VCSIPC	273
VCSIPC errors in Oracle trace/log files	273
Troubleshooting interconnects	273
Restoring communication between host and disks after cable disconnection	274
Troubleshooting Oracle	274
Oracle user must be able to read /etc/llttab File	274
Error when starting an Oracle instance	274
Oracle log files show shutdown called even when not shutdown manually	275
DBCA fails while creating database	275
CRS processes fail to startup	275
CRS fails after restart	275
Removing Oracle CRS if installation fails	276

	Troubleshooting the Virtual IP (VIP) Configuration	278
	OCR and Vote disk related issues	278
	Troubleshooting ODM	279
	File System configured incorrectly for ODM shuts down Oracle	279
Chapter 10	Prevention and recovery strategies	281
	Prevention and recovery strategies	281
	Verification of GAB ports in SF Oracle RAC cluster	281
	Examining GAB seed membership	282
	Manual GAB membership seeding	283
	Evaluating VCS I/O fencing ports	283
	Verifying normal functioning of VCS I/O fencing	284
	Managing SCSI-3 PGR keys in SF Oracle RAC cluster	285
	Identifying a faulty coordinator LUN	286
	Replacing a faulty coordinator disk without stopping I/O Fencing	287
	Collecting I/O Fencing kernel logs	288
Chapter 11	Tunable parameters	289
	About SF Oracle RAC tunable parameters	289
	About LMX tunable parameters	290
	Solaris OS LMX tunable parameters	290
	About VXFEN tunable parameters	292
	Configuring the VXFEN module parameters	293
	To reconfigure the VXFEN module	293
Section 4	Reference	297
Appendix A	Database FlashSnap status information	299
	About Database FlashSnap status information	299
	Database FlashSnap status information from the GUI	299
	Snapshot status information from the GUI	300
	Snapshot database status information from the GUI	302
	Database FlashSnap Snapshot status information from the CLI	303
	Snapshot status information from the CLI	303
	Snapshot database status information from the CLI	306

Appendix B	Using third party software to back up files	307
	About using third party software to back up files	307
	Using third party software to back up files	307
	Backing up and restoring Oracle Disk Manager files using Oracle RMAN	307
Appendix C	Error messages	309
	About error messages	309
	LMX error messages	309
	LMX critical error messages	309
	LMX non-critical error messages	311
	VxVM error messages	312
	VXFEN driver error messages	312
	VXFEN driver informational message	313
	Node ejection informational messages	313
Glossary		315
Index		319

SF Oracle RAC concepts and administration

- [Storage Foundation for Oracle RAC overview](#)
- [Configuration best practices](#)
- [Administering SF Oracle RAC and its components](#)

Storage Foundation for Oracle RAC overview

This chapter includes the following topics:

- [About SF Oracle RAC](#)
- [How SF Oracle RAC works \(high-level perspective\)](#)
- [Component products and processes of SF Oracle RAC](#)
- [About preventing data corruption with I/O fencing](#)
- [Supported SF Oracle RAC options](#)

About SF Oracle RAC

Veritas Storage Foundation™ for Oracle® RAC (SF Oracle RAC) by Symantec provides a robust infrastructure for Oracle Real Application Clusters (RAC) that simplifies management of RAC databases. SF Oracle RAC integrates existing Veritas storage management and clustering technologies into a flexible solution for administrators.

SF Oracle RAC enables you to perform the following tasks:

- Create a standard approach toward application and database management in data centers. While other clusterware can only work with an Oracle database, SF Oracle RAC incorporates existing Veritas storage management and clustering technologies that provide flexible support for many types of applications and databases. Administrators can apply existing expertise of Veritas technologies toward this product.

- Set up an infrastructure for Oracle RAC that simplifies database management while fully integrating with Oracle Clusterware, formerly known as Oracle Cluster Ready Services (CRS).
- Enhance scalability and availability with access to eight RAC instances per database in a cluster.
- Transition from a local high availability site to a wide-area disaster recovery environment with primary and secondary sites.
- Back up and recover databases using volume-level and file system-level snapshot technologies. SF Oracle RAC enables full volume-level snapshots for off-host processing that reduce load on production systems, and file system-level snapshots that involve efficient backup and rollback processes.
- Match data storage with data usage requirements through Database Dynamic Storage Tiering (DST). Data can then be relocated based upon data usage and other requirements determined by the database administrator (DBA).
- Prevent data corruption at the storage layer with robust split-brain protection.
- Increase scalability with high throughput and low latency technology for use by Oracle Cache Fusion.
- Share all types of files, in addition to Oracle database files, across nodes.
- Increase availability and performance with dynamic multipathing (DMP), which provides wide storage array support for protection from failures and performance bottlenecks in the HBAs and SAN switches.
- Model and test cluster configurations without affecting production systems using the VCS simulator clustering technologies.
- Optimize I/O performance through storage mapping technologies and tunable attributes.

How SF Oracle RAC works (high-level perspective)

Real Application Clusters (RAC) is a parallel database environment that takes advantage of the processing power of multiple computers. The Oracle database is the physical data stored in tablespaces on disk, while the Oracle instance is a set of processes and shared memory that provide access to the physical database. Specifically, the instance involves server processes acting on behalf of clients to read data into shared memory and make modifications to it, and background processes to write changed data to disk.

In traditional environments, only one instance accesses a database at a specific time. SF Oracle RAC enables all nodes to concurrently run Oracle instances and execute transactions against the same database. This software coordinates access

to the shared data for each node to provide consistency and integrity. Each node adds its processing power to the cluster as a whole and can increase overall throughput or performance.

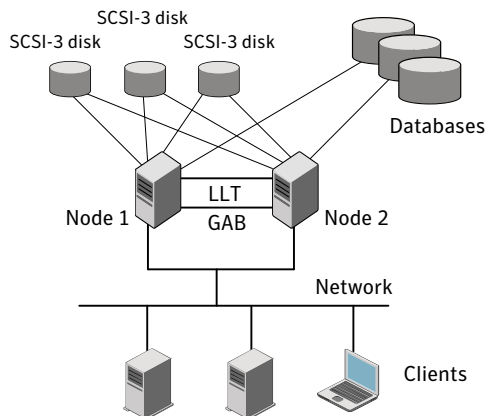
At a conceptual level, SF Oracle RAC is a cluster that manages applications (instances), networking, and storage components using resources contained in service groups. SF Oracle RAC clusters have many of the same properties as Veritas Cluster Server (VCS) clusters:

- Each node runs its own operating system.
- A cluster interconnect enables cluster communications.
- A public network connects each node to a LAN for client access.
- Shared storage is accessible by each node that needs to run the application.

[Figure 1-1](#) below displays the basic layout and individual components required for a SF Oracle RAC installation. This basic layout includes the following components:

- SCSI-3 Coordinator disks used for I/O fencing
- Nodes that form an application cluster and are connected to both the coordinator disks and databases
- Database(s) for storage and backup

Figure 1-1 SF Oracle RAC basic layout and components



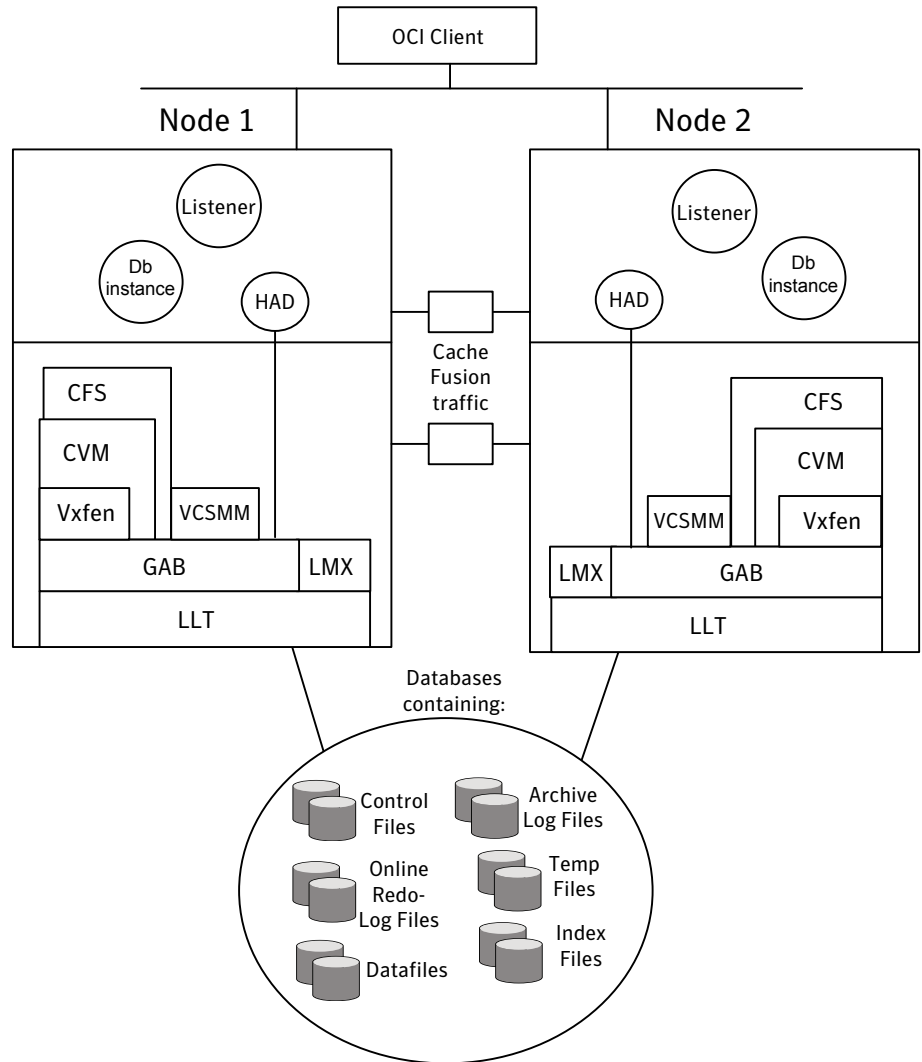
SF Oracle RAC adds the following technologies to a failover cluster environment, which are engineered specifically to improve performance, availability, and manageability of Oracle RAC environments:

- Cluster File System (CFS) and Cluster Volume Manager (CVM) technologies to manage multi-instance database access to shared storage.
- An Oracle Disk Manager (ODM) library to maximize Oracle disk I/O performance.
- Interfaces to Oracle Clusterware (referred to as CRS—formerly Cluster Ready Services) and RAC for managing cluster membership and communication.

Figure 1-2 displays the technologies that make up the SF Oracle RAC internal architecture.

SF Oracle RAC provides an environment that can tolerate failures with minimal downtime and interruption to users. If a node fails as clients access the same database on multiple nodes, clients attached to the failed node can reconnect to a surviving node and resume access. Recovery after failure in the SF Oracle RAC environment is far quicker than recovery for a failover database because another Oracle instance is already up and running. The recovery process involves applying outstanding redo log entries from the failed node.

Figure 1-2 SF Oracle RAC architecture



Component products and processes of SF Oracle RAC

To understand how SF Oracle RAC manages database instances running in parallel on multiple nodes, review the architecture and communication mechanisms that provide the infrastructure for Oracle RAC.

Table 1-1 SF Oracle RAC component products

Component product	Description
Cluster Volume Manager (CVM)	Enables simultaneous access to shared volumes based on technology from Veritas Volume Manager (VxVM). See “Cluster Volume Manager (CVM)” on page 29.
Cluster File System (CFS)	Enables simultaneous access to shared file systems based on technology from Veritas File System (VxFS). See “Cluster File System (CFS)” on page 31.
Database Accelerator	Provides the interface with the Oracle Disk Manager (ODM) API. See “Oracle Disk Manager” on page 34.
Cluster Server (VCS)	Uses technology from Veritas Cluster Server to manage Oracle RAC databases and infrastructure components. See “Veritas Cluster Server” on page 35.
RAC Extensions	Manages cluster membership and communications between cluster nodes. See “RAC extensions” on page 38.

Communication infrastructure

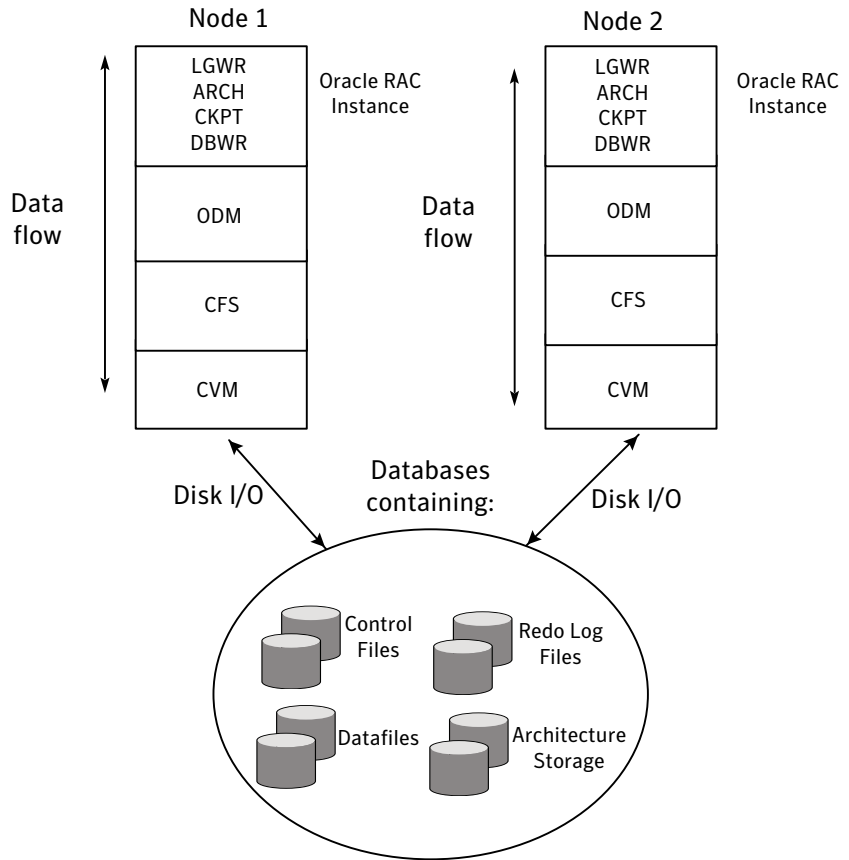
To understand the communication infrastructure, review the data flow and communication requirements.

Data flow

The CVM, CFS, ODM, and Oracle RAC elements reflect the overall data flow, or data stack, from an instance running on a server to the shared storage. The various Oracle processes composing an instance -- such as DB Writers, Log Writer, Checkpoint, and Archiver -- read and write data to the storage through the I/O stack. Oracle communicates through the ODM interface to CFS, which in turn accesses the storage through the CVM.

Figure 1-3 represents the overall data flow.

Figure 1-3 Data stack

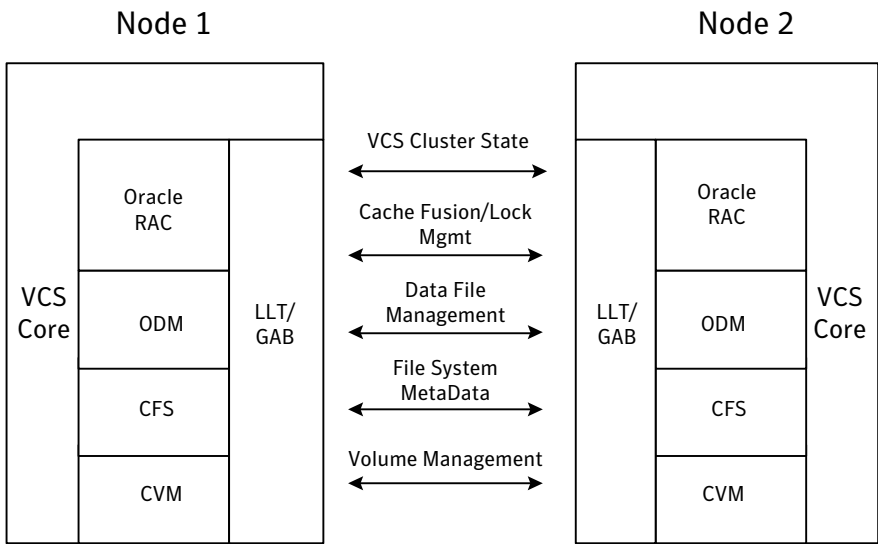


Communication requirements

End-users on a client system are unaware that they are accessing a database hosted by multiple instances. The key to performing I/O to a database accessed by multiple instances is communication between the processes. Each layer or component in the data stack must reliably communicate with its peer on other nodes to function properly. RAC instances must communicate to coordinate protection of data blocks in the database. ODM processes must communicate to coordinate data file protection and access across the cluster. CFS coordinates metadata updates for file systems, while CVM coordinates the status of logical volumes and maps.

Figure 1-4 represents the communication stack.

Figure 1-4 Communication stack



Cluster interconnect communication channel

The cluster interconnect provides the communication channel for all system-to-system communication, in addition to one-node communication between modules. Low Latency Transport (LLT) and Group Membership Services/Atomic Broadcast (GAB) make up the VCS communications package central to the operation of SF Oracle RAC.

In a standard operational state, significant traffic through LLT and GAB results from Lock Management, while traffic for other data is relatively sparse.

Low Latency Transport

LLT provides fast, kernel-to-kernel communications and monitors network connections. LLT functions as a high performance replacement for the IP stack and runs directly on top of the Data Link Protocol Interface (DLPI) layer. The use of LLT rather than IP removes latency and overhead associated with the IP stack.

The major functions of LLT are traffic distribution, heartbeats, and support for RAC Inter-Process Communications (VCSIPC):

- Traffic distribution
LLT distributes (load-balances) internode communication across all available cluster interconnect links. All cluster communications are evenly distributed

across as many as eight network links for performance and fault resilience. If a link fails, LLT redirects traffic to the remaining links.

- Heartbeats

LLT is responsible for sending and receiving heartbeat traffic over network links. The Group Membership Services function of GAB uses heartbeats to determine cluster membership.

- VCSIPC

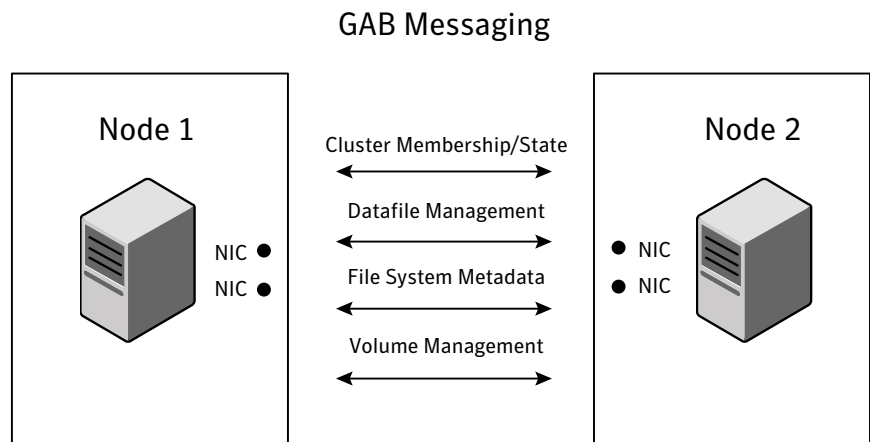
RAC Inter-Process Communications (VCSIPC) uses the VCSIPC shared library for these communications. VCSIPC leverages all features of LLT and uses LMX, an LLT multiplexer, to provide fast data transfer between Oracle processes on different nodes.

Group membership services/Atomic Broadcast

The GAB protocol is responsible for cluster membership and cluster communications.

Figure 1-5 shows the cluster communication using GAB messaging.

Figure 1-5 Cluster communication



Review the following information on cluster membership and cluster communication:

- Cluster membership

At a high level, all nodes configured by the installer can operate as a cluster; these nodes form a cluster membership. In SF Oracle RAC, a cluster membership specifically refers to all systems configured with the same cluster ID communicating by way of a redundant cluster interconnect.

All nodes in a distributed system, such as SF Oracle RAC, must remain constantly alert to the nodes currently participating in the cluster. Nodes can leave or join the cluster at any time because of shutting down, starting up, rebooting, powering off, or faulting processes. SF Oracle RAC uses its cluster membership capability to dynamically track the overall cluster topology.

SF Oracle RAC uses LLT heartbeats to determine cluster membership:

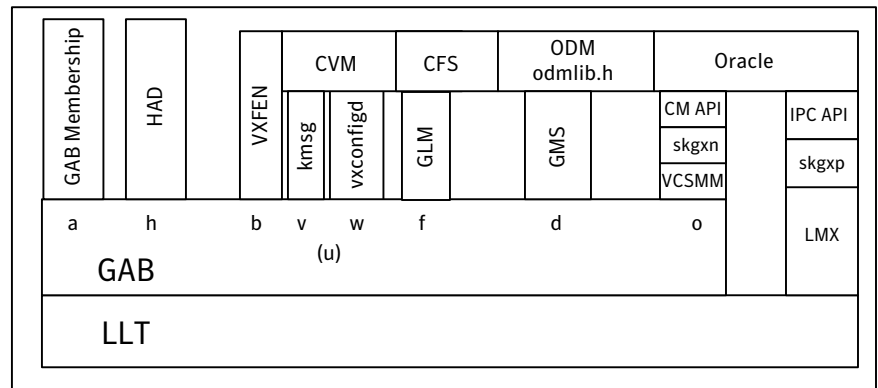
- When systems no longer receive heartbeat messages from a peer for a predetermined interval, a protocol excludes the peer from the current membership.
- GAB informs processes on the remaining nodes that the cluster membership has changed; this action initiates recovery actions specific to each module. For example, CVM must initiate volume recovery and CFS must perform a fast parallel file system check.
- When systems start receiving heartbeats from a peer outside of the current membership, a protocol enables the peer to join the membership.
- Cluster communications
GAB provides reliable cluster communication between SF Oracle RAC modules. GAB provides guaranteed delivery of point-to-point messages and broadcast messages to all nodes. Point-to-point messaging involves sending and acknowledging the message. Atomic-broadcast messaging ensures all systems within the cluster receive all messages. If a failure occurs while transmitting a broadcast message, GAB ensures all systems have the same information after recovery.

Low-level communication: port relationship between GAB and processes

All components in SF Oracle RAC use GAB for communication. Each process wanting to communicate with a peer process on other nodes registers with GAB on a specific port. This registration enables communication and notification of membership changes. For example, the VCS engine (HAD) registers on port h. HAD receives messages from peer had processes on port h. HAD also receives notification when a node fails or when a peer process on port h becomes unregistered.

Some processes use multiple ports for specific communications requirements. For example, CVM uses multiple ports to allow communications by kernel and user-level functions in CVM independently.

Figure 1-6 Low-level communication



For additional information about the different GAB ports:

See [“GAB port membership”](#) on page 260.

Cluster Volume Manager (CVM)

CVM is an extension of Veritas Volume Manager, the industry-standard storage virtualization platform. CVM extends the concepts of VxVM across multiple nodes. Each node recognizes the same logical volume layout, and more importantly, the same state of all volume resources.

CVM supports performance-enhancing capabilities, such as striping, mirroring, and mirror break-off (snapshot) for off-host backup. You can use standard VxVM commands from one node in the cluster to manage all storage. All other nodes immediately recognize any changes in disk group and volume configuration with no interaction.

CVM architecture

CVM is designed with a "master and slave" architecture. One node in the cluster acts as the configuration master for logical volume management, and all other nodes are slaves. Any node can take over as master if the existing master fails. The CVM master exists on a per-cluster basis and uses GAB and LLT to transport its configuration data.

Just as with VxVM, the Volume Manager configuration daemon, `vxconfigd`, maintains the configuration of logical volumes. This daemon handles changes to the volumes by updating the operating system at the kernel level. For example,

if a mirror of a volume fails, the mirror detaches from the volume and `vxconfigd` determines the proper course of action, updates the new volume layout, and informs the kernel of a new volume layout. CVM extends this behavior across multiple nodes and propagates volume changes to the master `vxconfigd`.

Note: You must perform operator-initiated changes on the master node.

The `vxconfigd` process on the master pushes these changes out to slave `vxconfigd` processes, each of which updates the local kernel.

CVM does not impose any write locking between nodes. Each node is free to update any area of the storage. All data integrity is the responsibility of the upper application. From an application perspective, standalone systems access logical volumes in the same way as CVM systems.

CVM imposes a "Uniform Shared Storage" model. All nodes must connect to the same disk sets for a given disk group. Any node unable to detect the entire set of physical disks for a given disk group cannot import the group. If a node loses contact with a specific disk, CVM excludes the node from participating in the use of that disk.

CVM communication

CVM communication involves various GAB ports for different types of communication. For an illustration of these ports, see [Figure 1-6](#).

CVM communication involves the following GAB ports:

- Port `w`

Most CVM communication uses port `w` for `vxconfigd` communications. During any change in volume configuration, such as volume creation, plex attachment or detachment, and volume resizing, `vxconfigd` on the master node uses port `w` to share this information with slave nodes.

When all slaves use port `w` to acknowledge the new configuration as the next active configuration, the master updates this record to the disk headers in the VxVM private region for the disk group as the next configuration.

- Port `v`

CVM uses port `v` for kernel-to-kernel communication. During specific configuration events, certain actions require coordination across all nodes. An example of synchronizing events is a resize operation. CVM must ensure all nodes see the new or old size, but never a mix of size among members. CVM also uses this port to obtain cluster membership from GAB and determine the status of other CVM members in the cluster.

CVM recovery

When a node leaves a cluster, the new membership is delivered by GAB, to CVM on existing cluster nodes. Fencing driver (VXFEN) ensures that split-brain scenarios are taken care of before CVM is notified. CVM then initiates recovery of mirrors of shared volumes that might have been in an inconsistent state following the exit of the node.

For database files, when ODM is enabled with SmartSync option, Oracle Resilvering handles recovery of mirrored volumes. For non-database files, this recovery is optimized using Dirty Region Logging (DRL). The DRL is a map stored in a special purpose VxVM sub-disk and attached as an additional plex to the mirrored volume. When a DRL subdisk is created for a shared volume, the length of the sub-disk is automatically evaluated so as to cater to the number of cluster nodes. If the shared volume has Fast Mirror Resync (FlashSnap) enabled, the DCO (Data Change Object) log volume created automatically has DRL embedded in it. In the absence of DRL or DCO, CVM does a full mirror resynchronization.

Configuration differences with VxVM

CVM configuration differs from VxVM configuration in the following areas:

- Configuration commands occur on the master node.
- Disk groups are created (could be private) and imported as shared disk groups.
- Disk groups are activated per node.
- Shared disk groups are automatically imported when CVM starts.

Cluster File System (CFS)

CFS enables you to simultaneously mount the same file system on multiple nodes and is an extension of the industry-standard Veritas File System. Unlike other file systems which send data through another node to the storage, CFS is a true SAN file system. All data traffic takes place over the storage area network (SAN), and only the metadata traverses the cluster interconnect.

In addition to using the SAN fabric for reading and writing data, CFS offers storage checkpoints and rollback for backup and recovery.

Access to cluster storage in typical SF Oracle RAC configurations use CFS. Raw access to CVM volumes is also possible but not part of a common configuration.

CFS architecture

SF Oracle RAC uses CFS to manage a file system in a large database environment. Since CFS is an extension of VxFS, it operates in a similar fashion and caches

metadata and data in memory (typically called buffer cache or vnode cache). CFS uses a distributed locking mechanism called Global Lock Manager (GLM) to ensure all nodes have a consistent view of the file system. GLM provides metadata and cache coherency across multiple nodes by coordinating access to file system metadata, such as inodes and free lists. The role of GLM is set on a per-file system basis to enable load balancing.

CFS involves a primary/secondary architecture. One of the nodes in the cluster is the primary node for a file system. Though any node can initiate an operation to create, delete, or resize data, the GLM master node carries out the actual operation. After creating a file, the GLM master node grants locks for data coherency across nodes. For example, if a node tries to modify a block in a file, it must obtain an exclusive lock to ensure other nodes that may have the same file cached have this cached copy invalidated.

SF Oracle RAC configurations minimize the use of GLM locking. Oracle RAC accesses the file system through the ODM interface and handles its own locking; only Oracle (and not GLM) buffers data and coordinates write operations to files. A single point of locking and buffering ensures maximum performance. GLM locking is only involved when metadata for a file changes, such as during create and resize operations.

CFS communication

CFS communication involves various GAB ports for different types of communication. For an illustration of these ports, see [Figure 1-6](#).

CFS uses port f for GLM lock and metadata communication. SF Oracle RAC configurations minimize the use of GLM locking except when metadata for a file changes.

CFS file system benefits

Many features available in VxFS do not come into play in an SF Oracle RAC environment because ODM handles such features. CFS adds such features as high availability, consistency and scalability, and centralized management to VxFS. Using CFS in an SF Oracle RAC environment provides the following benefits:

- Increased manageability, including easy creation and expansion of files without a file system, you must provide Oracle with fixed-size partitions. With CFS, you can grow file systems dynamically to meet future requirements. Use the `vxresize` command from CVM master and CFS primary to dynamically change the size of a CFS filesystem. For more information on `vxresize`, refer to the `vxresize(1)`, `fsadm_vxfs(1)` and `chfs(1)` manual pages.
- Less prone to user error

Raw partitions are not visible and administrators can compromise them by mistakenly putting file systems over the partitions. Nothing exists in Oracle to prevent you from making such a mistake.

- Data center consistency

If you have raw partitions, you are limited to a RAC-specific backup strategy. CFS enables you to implement your backup strategy across the data center.

CFS configuration differences

The first node to mount a CFS file system as shared becomes the primary node for that file system. All other nodes are "secondaries" for that file system.

Use the `fsclustadm` command from any node to view which node is primary and set the CFS primary node for a specific file system.

Mount the cluster file system individually from each node. The `-o cluster` option of the `mount` command mounts the file system in shared mode, which means you can mount the file system simultaneously on mount points on multiple nodes.

When using the `fsadm` utility for online administration functions on VxFS file systems, including file system resizing, defragmentation, directory reorganization, and querying or changing the largefiles flag, run `fsadm` from the primary node. This command fails from secondaries.

CFS recovery

The `vxfsckd` daemon is responsible for ensuring file system consistency when a node crashes that was a primary node for a shared file system. If the local node is a secondary node for a given file system and a reconfiguration occurs in which this node becomes the primary node, the kernel requests `vxfsckd` on the new primary node to initiate a replay of the intent log of the underlying volume. The `vxfsckd` daemon forks a special call to `fsck` that ignores the volume reservation protection normally respected by `fsck` and other VxFS utilities. The `vxfsckd` can check several volumes at once if the node takes on the primary role for multiple file systems.

After a secondary node crash, no action is required to recover file system integrity. As with any crash on a file system, internal consistency of application data for applications running at the time of the crash is the responsibility of the applications.

Comparing raw volumes and CFS for data files

Keep these points in mind about raw volumes and CFS for data files:

- If you use file-system-based data files, the file systems containing these files must be located on shared disks. Create the same file system mount point on each node.
- If you use raw devices, such as VxVM volumes, set the permissions for the volumes to be owned permanently by the database account.

For example, type:

```
# vxedit -g dgname set group=oracle owner=oracle mode 660 \  
/dev/vx/rdsk/dgname/volume_name
```

VxVM sets volume permissions on import. The VxVM volume, and any file system that is created in it, must be owned by the Oracle database account.

Oracle Disk Manager

SF Oracle RAC requires Oracle Disk Manager (ODM), a standard API published by Oracle for support of database I/O. Veritas provides a library for Oracle to use as its I/O library.

ODM architecture

When the Veritas ODM library is linked, Oracle is able to bypass all caching and locks at the file system layer and to communicate directly with raw volumes. The SF Oracle RAC implementation of ODM generates performance equivalent to performance with raw devices while the storage uses easy-to-manage file systems.

All ODM features can operate in a cluster environment. Nodes communicate with each other before performing any operation that could potentially affect another node. For example, before creating a new data file with a specific name, ODM checks with other nodes to see if the file name is already in use.

Veritas ODM performance enhancements

Veritas ODM enables the following performance benefits provided by Oracle Disk Manager:

- Locking for data integrity.
- Few system calls and context switches.
- Increased I/O parallelism.
- Efficient file creation and disk allocation.

Databases using file systems typically incur additional overhead:

- Extra CPU and memory usage to read data from underlying disks to the file system cache. This scenario requires copying data from the file system cache to the Oracle cache.
- File locking that allows for only a single writer at a time. Allowing Oracle to perform locking allows for finer granularity of locking at the row level.
- File systems generally go through a standard Sync I/O library when performing I/O. Oracle can make use of Kernel Async I/O libraries (KAIO) with raw devices to improve performance.

ODM communication

ODM uses port d to communicate with other ODM instances to support the file management features of Oracle Managed Files (OMF). OMF enables DBAs to set `init.ora` parameters for db datafile, controlfile, and logfile names and for those structures to be named automatically. OMF allows for the automatic deletion of physical data files when DBAs remove tablespaces.

For an illustration of the ODM and port d, see [Figure 1-6](#).

Veritas Cluster Server

Veritas Cluster Server (VCS) directs SF Oracle RAC operations by controlling the startup and shutdown of components layers and providing monitoring and notification of failure.

In a typical SF Oracle RAC configuration, the RAC service groups for VCS run as "parallel" service groups rather than "failover" service groups; in the event of a failure, VCS does not attempt to migrate a failed service group. Instead, the software enables you to configure the group to restart on failure.

VCS architecture

The High Availability Daemon (HAD) is the main VCS daemon running on each node. HAD tracks changes in the cluster configuration and monitors resource status by communicating over GAB and LLT. HAD manages all application services using agents, which are installed programs to manage resources (specific hardware or software entities).

The VCS architecture is modular for extensibility and efficiency; HAD does not need to know how to start up Oracle or any other application under VCS control. Instead, you can add agents to manage different resources with no effect on the engine (HAD). Agents only communicate with HAD on the local node, and HAD communicates status with HAD processes on other nodes. Because agents do not

need to communicate across systems, VCS is able to minimize traffic on the cluster interconnect.

SF Oracle RAC provides specific agents for VCS to manage CVM, CFS, and Oracle agents.

VCS communication

SF Oracle RAC uses port `h` for HAD communication. Agents communicate with HAD on the local node about resources, and HAD distributes its view of resources on that node to other nodes through port `h`. HAD also receives information from other cluster members to update its own view of the cluster.

Cluster configuration files

VCS uses two configuration files in a default configuration:

- The `main.cf` file defines the entire cluster, including the cluster name, systems in the cluster, and definitions of service groups and resources, in addition to service group and resource dependencies.
- The `types.cf` file defines the resource types. Each resource in a cluster is identified by a unique name and classified according to its type. VCS includes a set of pre-defined resource types for storage, networking, and application services.

Additional files similar to `types.cf` may be present if you add agents. For example, SF Oracle RAC includes additional resource types files, such as `OracleTypes.cf`, `PrivNIC.cf`, and `MultiPrivNIC.cf`.

Oracle RAC

Review the following information on Cluster Ready Services, the Oracle Cluster Registry, application resources, and the voting disk.

Note: Refer to the Oracle RAC documentation for additional information.

Oracle Clusterware

Oracle Clusterware, also known as Cluster Ready Services (CRS), manages Oracle cluster-related functions including membership, group services, global resource management, and databases.

CRS is required for every Oracle RAC instance and is only applicable to Oracle 10g and Oracle 11g.

CRS requires the following major components:

- A cluster interconnect that allows for cluster communications
- A private virtual IP address for cluster communications over the interconnect.
- A public virtual IP address for client connections.
- Shared storage accessible by each node.

Oracle Cluster Registry

The Oracle Cluster Registry (OCR) contains cluster and database configuration and state information for Oracle RAC and CRS.

This is roughly analogous to the `main.cf` file and in-memory configuration in VCS. However, only one process performs I/O to the OCR file on disk.

The information maintained in the OCR includes:

- The list of nodes
- The mapping of database instances to nodes
- CRS application resource profiles
- Resource profiles that define the properties of resources under CRS control
- Rules that define dependencies between the CRS resources
- The current state of the cluster

The OCR data exists on a shared raw volume or a cluster file system that is accessible to each node. For Oracle RAC 10g, this requires a minimum of 100 MB of disk space. For Oracle RAC 11g, this requires a minimum of 256 MB of disk space.

You can specify redundant storage for the OCR to protect against failures. CRS faults nodes if the OCR is not accessible because of corruption or disk failure. Oracle automatically backs up OCR data. You can also export the OCR contents before making configuration changes in CRS. This way, if you encounter configuration problems and are unable to restart CRS, you can restore the original contents.

Application Resources

CRS application resources are similar to VCS resources. Each component controlled by CRS is defined by an application resource, including databases, instances, services, and node applications.

Unlike VCS, CRS uses separate resources for components that run in parallel on multiple nodes.

Resource Profiles

Resources are defined by profiles, which are similar to the attributes that define VCS resources. The OCR contains application resource profiles, dependencies, and status information.

CRS Node Applications

CRS uses these node application resources to manage Oracle components, such as databases, listeners, and virtual IP addresses. Node application resources are created during CRS installation.

Voting Disk

The voting disk is a heartbeat mechanism used by CRS to maintain cluster node membership. Voting disk data exists on a shared raw volume or a cluster file system that is accessible to each node.

The ocssd processes of CRS provides cluster node membership and group membership information to RAC instances. On each node, ocssd processes write a heartbeat to the voting disk every second. If a node is unable to access the voting disk, CRS determines the cluster is in a split brain condition and panics the node.

RAC extensions

Oracle RAC relies on several support services provided by VCS. Key features include Veritas Cluster Server Membership Manager (VCSMM) and Veritas Cluster Server Inter-Process Communication (VCSIPC), and LLT Multiplexer (LMX).

Veritas Cluster Server membership manager

To protect data integrity by coordinating locking between RAC instances, Oracle must know which instances actively access a database. Oracle provides an API called skgxn (system kernel generic interface node membership) to obtain information on membership. SF Oracle RAC implements this API as a library linked to Oracle after you install Oracle RAC. Oracle uses the linked skgxn library to make ioctl calls to VCSMM, which in turn obtains membership information for clusters and instances by communicating with GAB on port o.

For an illustration of the connection between VCSMM, the skgxn library, and port o, see [Figure 1-6](#).

LLT multiplexer

Oracle instances use the `skgxp` library for interprocess communication. This interface enables Oracle to send communications between processes on instances.

SF Oracle RAC provides a library dynamically linked to Oracle at installation time to implement the `skgxp` functionality. This module communicates with the LLT Multiplexer (LMX) using `ioctl` calls.

The LMX module is a kernel module designed to receive communications from the `skgxp` module and pass them on to the correct process on the correct instance on other nodes. The LMX module "multiplexes" communications between multiple processes on other nodes. LMX leverages all features of LLT, including load balancing and fault resilience.

Note: The LLT multiplexer is no longer supported with Oracle 11g.

Veritas Cluster Server inter-process communication

To coordinate access to a single database by multiple instances, Oracle uses extensive communications between nodes and instances. Oracle uses Inter-Process Communications (VCSIPC) for Global Enqueue Service locking traffic and Global Cache Service cache fusion. SF Oracle RAC uses LLT to support VCSIPC in a cluster and leverages its high-performance and fault-resilient capabilities.

Oracle has an API for VCSIPC, System Kernel Generic Interface Inter-Process Communications (`skgxp`), that isolates Oracle from the underlying transport mechanism. As Oracle conducts communication between processes, it does not need to know how data moves between systems; the cluster implementer can create the highest performance for internode communications without Oracle reconfiguration.

Oracle and cache fusion traffic

Private IP address types are required by Oracle for cache fusion traffic.

- For Oracle 10g, Veritas Cluster Inter-Process Communication (VCSIPC) supports Oracle cache fusion and provides an API for Oracle cache fusion. The API is provided through the `libskgxp` library. The `libskgxp` library makes calls into the LLT Multiplexer (LMX). LMX uses Low Latency Transport LLT to send data across the private interconnects. LMX also provides link aggregation and redundancy of private links.
- For Oracle 11g, Symantec provides a MultiPrivNIC agent and UDP/IP support for Oracle's cache fusion.

About preventing data corruption with I/O fencing

I/O fencing is a feature that prevents data corruption in the event of a communication breakdown in a cluster.

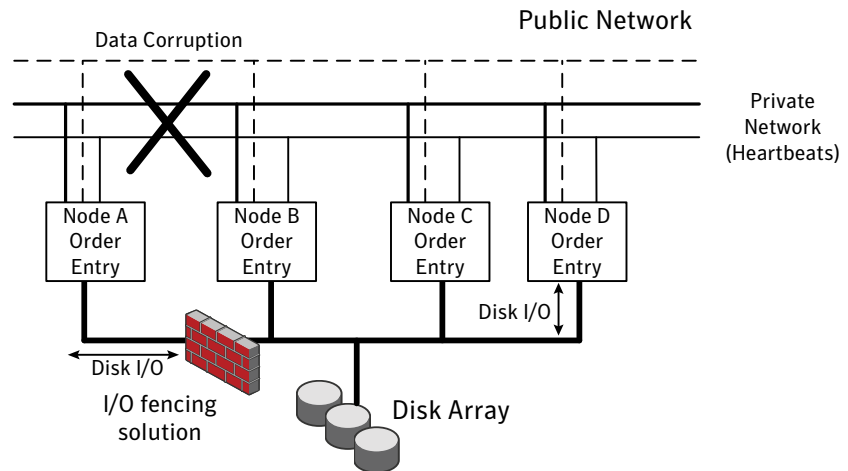
To provide high availability, the cluster must be capable of taking corrective action when a node fails. In this situation, SF Oracle RAC configures its components to reflect the altered membership.

Problems arise when the mechanism that detects the failure breaks down because symptoms appear identical to those of a failed node. For example, if a system in a two-node cluster fails, the system stops sending heartbeats over the private interconnects. The remaining node then takes corrective action. The failure of the private interconnects, instead of the actual nodes, presents identical symptoms and causes each node to determine its peer has departed. This situation typically results in data corruption because both nodes try to take control of data storage in an uncoordinated manner.

In addition to a broken set of private networks, other scenarios can generate this situation. If a system is so busy that it appears to stop responding or "hang," the other nodes could declare it as dead. This declaration may also occur for the nodes that use the hardware that supports a "break" and "resume" function. When a node drops to PROM level with a break and subsequently resumes operations, the other nodes may declare the system dead. They can declare it dead even if the system later returns and begins write operations.

SF Oracle RAC uses I/O fencing to remove the risk that is associated with split brain. I/O fencing allows write access for members of the active cluster. It blocks access to storage from non-members. It even blocks a node that is alive is unable to cause damage.

[Figure 1-7](#) displays a schematic of a four node cluster, each node writing order entries to the connected disk array. When the private network connection between the four nodes is disrupted (between Node A and the other 3 nodes in the figure below), a split brain situation occurs with the possibility of data corruption to the disk array. The I/O fencing process prevents split-brain and any data corruption by fencing off Node A from the cluster.

Figure 1-7 Private network disruption and I/O fencing solution

About SCSI-3 Persistent Reservations

SCSI-3 Persistent Reservations (SCSI-3 PR) are required for I/O fencing and resolve the issues of using SCSI reservations in a clustered SAN environment. SCSI-3 PR enables access for multiple nodes to a device and simultaneously blocks access for other nodes.

SCSI-3 reservations are persistent across SCSI bus resets and support multiple paths from a host to a disk. In contrast, only one host can use SCSI-2 reservations with one path. If the need arises to block access to a device because of data integrity concerns, only one host and one path remain active. The requirements for larger clusters, with multiple nodes reading and writing to storage in a controlled manner, make SCSI-2 reservations obsolete.

SCSI-3 PR uses a concept of registration and reservation. Each system registers its own "key" with a SCSI-3 device. Multiple systems registering keys form a membership and establish a reservation, typically set to "Write Exclusive Registrants Only." The WERO setting enables only registered systems to perform write operations. For a given disk, only one reservation can exist amidst numerous registrations.

With SCSI-3 PR technology, blocking write access is as easy as removing a registration from a device. Only registered members can "eject" the registration of another member. A member wishing to eject another member issues a "preempt and abort" command. Ejecting a node is final and atomic; an ejected node cannot eject another node. In SF Oracle RAC, a node registers the same key for all paths

to the device. A single preempt and abort command ejects a node from all paths to the storage device.

About I/O fencing operations

I/O fencing, provided by the kernel-based fencing module (vxfen), performs identically on node failures and communications failures. When the fencing module on a node is informed of a change in cluster membership by the GAB module, it immediately begins the fencing operation. The node tries to eject the key for departed nodes from the coordinator disks using the preempt and abort command. When the node successfully ejects the departed nodes from the coordinator disks, it ejects the departed nodes from the data disks. In a split brain scenario, both sides of the split would race for control of the coordinator disks. The side winning the majority of the coordinator disks wins the race and fences the loser. The loser then panics and restarts the system.

About I/O fencing communication

The vxfen driver connects to GAB port b to intercept cluster membership changes (reconfiguration messages). During a membership change, the fencing driver determines which systems are members of the cluster to allow access to shared disks.

See [“Low-level communication: port relationship between GAB and processes”](#) on page 28.

After completing fencing operations, the driver passes reconfiguration messages to higher modules. CVM handles fencing of data drives for shared disk groups. After a node successfully joins the GAB cluster and the driver determines that a preexisting split brain does not exist, CVM can import all shared disk groups. The CVM master coordinates the order of import and the key for each disk group. As each slave joins the cluster, it accepts the CVM list of disk groups and keys, and adds its proper digit to the first byte of the key. Each slave then registers the keys with all drives in the disk groups.

Supported SF Oracle RAC options

SF Oracle RAC supports the following activities using optional product features:

- Managing databases
- Securing a cluster
- Managing multiple clusters
- Managing storage

- Replicating data to remote sites

Managing databases using SF Oracle RAC

SF Oracle RAC database management options include:

- The ability to back up and recover data at the volume and file system levels using Veritas Storage Checkpoints and Veritas Database FlashSnap.
- The ability to manage tiered storage using Database Dynamic Tiered Storage. You can set policies and move files to maximize the economic efficiency of your storage.
- The ability to evaluate or troubleshoot I/O performance with Veritas Storage Mapping. You can access mapping information that allows for a detailed understanding of the storage hierarchy in which files reside.

Securing a cluster using SF Oracle RAC

The Symantec Product Authentication Service is an optional feature common to Symantec products that validates identities based on existing network operating system domains (such as NIS and NT) or private domains. The authentication service protects communication channels among Symantec application clients and services through message integrity and confidentiality services.

Before you install the authentication service, refer to the *Symantec Product Authentication Service Installation Guide* at the following location on the Veritas software disc:

`authentication_service/docs/vxat_install.pdf`

Symantec Product Authentication Service secures communication using digital certificates for authentication and SSL to encrypt communication over the public network. You can configure SF Oracle RAC to use the Authentication Service to secure communication between the following:

- Cluster nodes and clients, including the VCS Java and the Web consoles. You can set up Authentication Service for the cluster during the SF Oracle RAC installation and configuration process. If you want to enable Authentication Service after installation, refer to the *Veritas Cluster Server User's Guide*.
- Veritas Cluster Server Management Console management server and the centrally managed SF Oracle RAC clusters.

To configure the cluster in secure mode, SF Oracle RAC requires you to configure a system in your enterprise as root broker and all nodes in the cluster as authentication brokers.

- **Root broker**
A root broker serves as the main registration and certification authority; it has a self-signed certificate and can authenticate other brokers. The root broker is only used during initial creation of an authentication broker.
- **Authentication brokers**
Authentication brokers serve as intermediate registration and certification authorities. Authentication brokers have certificates that are signed by the root. Each node in SF Oracle RAC cluster serves as an authentication broker.

Managing multiple SF Oracle RAC clusters using a common Web console server

Veritas Cluster Server Management Console is a high availability management solution. You can use the cluster management console option with SF Oracle RAC to monitor and administer multiple clusters from a single web console. SF Oracle RAC nodes must be discovered by the cluster management console server before you can manage the nodes using the server.

For more information, refer to the *Veritas Cluster Server Management Console Implementation Guide*.

Replicating data to a remote site using SF Oracle RAC

The global clustering feature of VCS enables wide-area failover and disaster recovery, also referred to as high availability and disaster recovery, or HA/DR. Local clustering provides local failover for each site or building. Local clustering does not provide protection against outages caused by large-scale disasters such as major floods, hurricanes, and earthquakes that affect an entire city or region. An entire cluster could be affected by such outages. Global clustering ensures data availability during large-scale disasters. This type of clustering involves migrating applications between clusters over a considerable distance.

VCS enables you to link clusters running at separate locations and switch service groups across clusters. Global clustering provides complete protection against failure of a cluster.

To understand how global clusters work, review the example of an Oracle RAC database configured using global clustering. Oracle RAC is installed and configured in cluster A and cluster B. Oracle data is located on shared disks within each cluster and is replicated across clusters to ensure data concurrency. The VCS service groups for Oracle are online on a node in cluster A and are configured to fail over on cluster A and cluster B.

VCS continuously monitors and communicates events between clusters. Cluster-to-cluster communications ensure that the global cluster environment is

aware of the state of global service group at all times. If cluster A fails, the service groups can fail over to a remote cluster B. VCS also redirects clients when the application is online at the new location.

Supported replication technologies

SF Oracle RAC supports Veritas Volume Replicator (VVR) as well as several hardware-based replication technologies. VVR is a fully integrated component of Veritas Volume Manager that replicates data to remote locations over any standard IP network to provide continuous data availability.

SF Oracle RAC supports the following hardware-based replication technologies:

- EMC SRDF
- Hitachi TrueCopy
- IBM Metro Mirror
- EMC MirrorView
- IBM SAN Volume Controller (SVC)
- Oracle Data Guard

Configuration best practices

This chapter includes the following topics:

- [About configuration best practices](#)
- [Cluster configuration](#)
- [Oracle configuration best practices](#)
- [Database storage configuration best practices](#)
- [Database snapshot and backup options](#)
- [Database storage optimization options](#)

About configuration best practices

SF Oracle RAC provides configuration options to support the requirements of diverse environments. You can optimize your clustering configuration, Oracle configuration, or storage configuration depending on your specific requirements.

Note: In this document, all command examples with '\$' are oracle user commands and all command examples with '#' are root user commands. The document does not always provide the full path to the files or directories used in these examples. Make sure that you set the PATH and MANPATH variables.

To set the PATH and MANPATH variables:

See [“Setting the PATH variable”](#) on page 56.

See [“Setting the MANPATH variable”](#) on page 57.

Cluster configuration

Cluster configuration options include:

- Basic shared storage cluster
- Secure shared storage cluster
- Campus, or Metropolitan, shared storage cluster
- Global cluster

Table 2-1

Cluster configuration	Requirements	Advantage
Basic shared storage cluster	Minimal requirements For information about minimal requirements, see the <i>Veritas Storage Foundation for Oracle RAC Installation and Configuration Guide</i> .	Simple, scalable
Secure shared storage cluster	Additional set up for root broker and authentication For setup requirements, see the <i>Veritas Storage Foundation for Oracle RAC Installation and Configuration Guide</i> .	Additional security
Campus, or Metropolitan, shared storage cluster	Cluster spans two sets of storage within a distance of up to 50 miles of each other	Extended failover option protects against local disaster
Global cluster	Cluster spans two remote sites, SFRAC HA/DR license is required, replication technology is required	Long distance failover option protects against site disaster

Basic clustered shared storage cluster

In this configuration, nodes in a single cluster share access to a storage device clustered with Cluster Volume Manager (CVM), typically over a SAN. For example, in a multi-node cluster, any node designated to access the Oracle RAC database must have access to the storage where the database’s tablespaces, redo logs and control files are stored. Clustering storage with CVM supports the multiple Oracle

instances required for Oracle RAC databases. On VCS basic shared storage clusters, when a node or application fails, all data required to start on another node is stored on the shared disk. On SF Oracle RAC clustered shared storage clusters, multiple nodes with Oracle instances are already up and accessing the same clustered shared storage database simultaneously.

Clusters running in secure mode

SF Oracle RAC uses Symantec Product Authentication Service (AT) to provide secure communication between cluster nodes and clients. AT uses digital certificates for authentication and SSL to encrypt communication over the public network to secure communications.

For more information about AT, refer to the Symantec Product Authentication service documentation.

Campus, or metropolitan, shared storage cluster

In a campus environment, VCS and Cluster Volume Manager (CVM) are used to create a cluster that spans multiple data centers or buildings. Instead of a single storage array, data is mirrored between arrays using CVM. This provides synchronized copies of data at both sites. This procedure is identical to mirroring between two arrays in a data center, only now it is spread over a distance.

SF Oracle RAC addresses the following basic challenges in campus cluster configurations:

- Latency challenges: SF Oracle RAC handles latency challenges in keeping mirrors synchronized while ensuring efficient recovery in case of site failures for both data and VxVM metadata.
- Read performance: The read performance is enhanced as data is read from local mirrors.
- Site awareness: SF Oracle RAC makes sure that all mirrors on a site are detached proactively when a site goes down.

Global cluster

A global cluster links clusters at separate locations and enables wide-area failover and disaster recovery. Local clustering provides local failover for each site or building. Campus and replicated cluster configurations offer protection against disasters affecting limited geographic regions. Large scale disasters such as major floods, hurricanes, and earthquakes can cause outages for an entire city or region. In such situations, you can ensure data availability by migrating applications to sites located considerable distances apart.

For more on replication in an SF Oracle RAC environment:

See [“Replicating data to a remote site using SF Oracle RAC”](#) on page 44.

Oracle configuration best practices

The following recommendations provide best practices for configuring Oracle in an SF Oracle RAC environment.

Placement of Oracle binaries	Install Oracle binaries on local disk to enable phased upgrade of the cluster.
Placement of Oracle data files and logs	Create separate clustered file systems for Oracle data files. Place redo and archived logs on CFS rather than on local file systems.
Placement of OCR and Voting disk	Place the OCR and VOTE devices on CVM volumes or CFS file systems to provide high availability to VOTE disks with dynamic multipathing. Mirror the CVM volumes that contain OCR and VOTE devices for redundancy.
Configuration of Oracle private IPs	Configure Oracle private IP addresses under VCS PrivNIC or MultiPrivNIC to protect against link failures. This requires you to configure CRS private IP addresses and UDP cache fusion IP addresses on LLT interconnects.
ASM considerations	■ Use VxVM mirrored volumes with dynamic multi-pathing with external redundancy to ensure high availability. ■ Do not use VxVM volumes, which are used by ASM, for any other purpose such as creation of file systems. ■ Do not enable ODM when databases are installed on ASM.

Database storage configuration best practices

The following recommendations provide best practices for improving performance, protection, and high availability of data in your SF Oracle RAC environment.

Storage	■ Use multiple storage arrays to ensure protection against array failures. ■ Design the storage layout keeping in mind performance and high availability requirements. Use technologies such as striping and mirroring. ■ Use appropriate stripe width and depth to optimize I/O performance. ■ Use SCSI-3 PGR compliant storage. ■ Use dynamic multi-pathing with multiple access paths to disks.
Volume design	■ Mirror the volumes across two or more storage arrays, especially with host-based mirrors. ■ Separate Oracle recovery structures from database files to ensure high availability when you design placement policies. ■ Separate redo logs and place them on the fastest storage (for example, RAID 1+0) for better performance. ■ Implement zoning to control access to shared storage. Be aware that physical disks may be shared by multiple servers or applications and must therefore be protected from accidental access.
File design	■ Use ODM with Cluster File System for better performance. ODM with SmartSync enables faster recovery of mirrored volumes using Oracle resilvering. ■ Create separate file systems for Oracle binaries, data, redo logs, and archive logs. ■ Always place redo and archived logs on CFS file systems rather than local file systems.

Database snapshot and backup options

You can configure the following database components with SF Oracle RAC for cloning and recovery of databases.

- Storage Checkpoints
- Database FlashSnap

The following sections provide a brief overview of these features.

Database FlashSnap for cloning

Veritas Database FlashSnap helps to create a point-in-time copy of a database for backup and off-host processing. Database FlashSnap allows you to make backup copies of your volumes online and with minimal interruption to users.

Database FlashSnap lets you capture an online image of an actively changing database at a given instant, known as a snapshot. A snapshot copy of the database is referred to as a database snapshot. You can use a database snapshot on the same host as the production database or on a secondary host sharing the same storage. A database snapshot can be used for off-host processing applications, such as backup, data warehousing, and decision-support queries. When the snapshot is no longer needed, the database administrator can import the original snapshot back to the primary host and resynchronize the snapshot to the original database volumes. Database FlashSnap commands are executed from the command line interface.

Database FlashSnap advantages

Database FlashSnap provides the following advantages:

- The database snapshot can be used on the same host as the production database or on a secondary host sharing the same storage.
- In many companies, there is a clear separation between the roles of system administrators and database administrators. Creating database snapshots typically requires superuser (root) privileges, privileges that database administrators do not usually have. Because root privileges are not required, Database FlashSnap overcomes these obstacles by enabling database administrators to easily create consistent snapshots of the database.

For additional Database Flashsnap information: See [“About Veritas Database FlashSnap”](#) on page 157.

Storage Checkpoints for recovery

A Storage Checkpoint creates an exact image of a database instantly and provides a consistent image of the database from the point in time the Storage Checkpoint was created. The Storage Checkpoint image is managed and available through the command line interface (CLI).

Because each Storage Checkpoint is a consistent, point-in-time image of a file system, Storage Rollback is the restore facility for these on-disk backups. Storage Rollback rolls back changed blocks contained in a Storage Checkpoint into the primary file system for faster database restoration.

The combination of data redundancy (disk mirroring) and Storage Checkpoints is recommended for highly critical data to protect them from both physical media failure and logical errors.

Advantages and limitations of Storage Checkpoints

Storage Checkpoints and rollback provides the following advantages:

- Initially, a Storage Checkpoint contains no data—it contains only the inode list and the block map of the primary fileset. This block map points to the actual data on the primary file system.
- Because only the inode list and block map are needed and no data is copied, creating a Storage Checkpoint takes only a few seconds and very little space.
- A Storage Checkpoint keeps track of block change information and thereby enables incremental database backup at the block level.
- A Storage Checkpoint helps recover data from incorrectly modified files.
- A Storage Checkpoint can be mounted, allowing regular file system operations to be performed. Mountable Storage Checkpoints can be used for a wide range of application solutions, including backup, investigations into data integrity, staging upgrades or database modifications, and data replication solutions.

The limitations of Storage Checkpoints are as follows:

- Storage Checkpoints can only be used to restore from logical errors (for example, a human error).
- Because all the data blocks are on the same physical device, Storage Checkpoints cannot be used to restore files due to a media failure. A media failure requires a database restore from a tape backup or a copy of the database files kept on a separate medium.

For additional detailed Storage Checkpoint and Storage Rollback information: See [“About Storage Checkpoints and Storage Rollback in SF Oracle RAC”](#) on page 121.

Database storage optimization options

Database Dynamic Storage Tiering (DST) matches data storage with data usage requirements. Data can then be relocated based upon data usage and other requirements determined by the database administrator (DBA).

As more and more data is retained over a period of time, eventually, some of that data is needed less frequently. The data that is needed less frequently still requires a large amount of disk space. DST enables the database administrator to manage data so that less frequently used data can be moved to slower, less expensive disks.

This also permits the frequently accessed data to be stored on faster disks for quicker retrieval.

For additional Dynamic Storage Tiering information:

See [“About Database Dynamic Storage Tiering”](#) on page 215.

Administering SF Oracle RAC and its components

This chapter includes the following topics:

- [Administering SF Oracle RAC](#)
- [Administering VCS](#)
- [Administering I/O fencing](#)
- [Administering CFS](#)
- [Administering CVM](#)
- [Administering Oracle](#)
- [Administering ODM](#)

Administering SF Oracle RAC

This section provides instructions for the following SF Oracle RAC administration tasks:

- [Setting the PATH variable](#)
- [Setting the MANPATH variable](#)
- [Turning on the debug messages for the kernel drivers](#)
- [Stopping and starting LLT and GAB](#)
- [Stopping SF Oracle RAC on a single node](#)
- [Adding LLT links to increase capacity](#)

If you encounter issues while administering SF Oracle RAC, refer to the troubleshooting section for assistance.

See [“About troubleshooting SF Oracle RAC”](#) on page 257.

Setting the PATH variable

To set the PATH variable for the root user:

For sh, ksh, bash:

```
PATH=/opt/VRTSob/bin:/usr/ccs/bin:/usr/local/bin:/usr/bin/X11:/sbin:/usr/bin:  
/usr/sbin:/usr/lib/vxvm/bin:/opt/VRTSvxfs/sbin:/opt/VRTSvxfs/cfs/bin:  
/opt/VRTSvcs/bin:/opt/VRTS/bin:/etc/vx/bin:/usr/ucb:/opt/VRTSvcs/vxfen/bin/  
/opt/VRTSdbcom/bin:/opt/VRTSgab export PATH
```

For csh:

```
setenv PATH  
/opt/VRTSob/bin:/usr/ccs/bin:/usr/local/bin:/usr/bin/X11:/sbin:/usr/bin:/usr/sbin:  
/usr/lib/vxvm/bin:/opt/VRTSvxfs/sbin:/opt/VRTSvxfs/cfs/bin:/opt/VRTSvcs/bin:  
/opt/VRTS/bin:/etc/vx/bin:/usr/ucb:/opt/VRTSvcs/vxfen/bin:/opt/VRTSdbcom/bin/  
/opt/VRTSgab
```

To set the PATH variable for the oracle user:

For sh, ksh, bash:

```
PATH=/usr/bin:/etc:/usr/sbin:/usr/ucb:/usr/oracle/bin:/usr/bin/X11:  
/sbin:./oracle/orahome/bin:/crshome/crs/bin:/opt/VRTS/bin:  
/opt/VRTSdbed/bin export PATH
```

For csh:

```
setenv PATH  
/usr/bin:/etc:/usr/sbin:/usr/ucb:/usr/oracle/bin:  
/usr/bin/X11:/sbin:./oracle/orahome/bin:  
/crshome/crs/bin:/opt/VRTS/bin:/opt/VRTSdbed/bin
```

In this example:

- `/oracle/orahome/bin` is the location of the Oracle Home bin directory.
- `/crshome/crs/bin` is the location of the Oracle CRS bin directory.
- `/opt/VRTSdbed/bin` is the location of the dbed bin directory.

Setting the MANPATH variable

To set the MANPATH variable for the root user:

For sh, ksh, bash:

```
MANPATH=/usr/man:/usr/share/man:/opt/VRTS/man
export MANPATH
```

For csh:

```
setenv MANPATH /usr/man:/usr/share/man:/opt/VRTS/man
```

Stopping SF Oracle RAC on a single node

This section describes the procedure for gracefully stopping SF Oracle RAC on a single node within a cluster. This procedure may be required for cluster or node maintenance, cluster or node testing, or for any other user specific requirement.

This procedure involves the following tasks:

- Stopping SF Oracle RAC on the single node
- Stopping Oracle in the cluster
- Maintaining cluster membership for that single node

Stopping SF Oracle RAC on a single node within a cluster

- 1 Log in as superuser to the node.

Note: In this example procedure, the node is galaxy.

- 2 For Oracle9i, stop all Oracle resources including the database on galaxy.

- Stop gsd. On galaxy, log in as oracle user and enter the following command:

```
$ $ORACLE_HOME/bin/gsdctl stop
```

- Offline all VCS service groups that contain resources for managing Oracle listener process or the database instances. As root user, enter the following command:

```
# hagrps -offline group_name -sys galaxy
```

- 3 For Oracle RAC 10g and Oracle RAC 11g, stop all resources.

- If the database instances are under VCS control, then take the corresponding VCS service groups offline. As superuser, enter the following command:

```
# hagrps -offline group_name -sys galaxy
```

- If the database instances are under CRS control, then run the following command on the one node (galaxy):

```
$ srvctl stop instance -d database_name -i instance_name
```

- If the resources are under CRS control, then stop all resources configured under CRS control. As oracle user, enter the following command:

```
$ srvctl stop nodeapps -n galaxy
```

Repeat the above command for each node in the cluster.

Note: If the Oracle listener or database instances are managed by CRS, they are taken offline by the `srvctl` command as shown above.

- 4 For Oracle RAC 10g and Oracle RAC 11g, stop CRS on each node in the cluster. Symantec highly recommends putting CRS under VCS control.

- If CRS is controlled by VCS, then log in as superuser on any system in the cluster and enter the following command:

```
# hares -offline cssd-resource -sys galaxy
```

- If CRS is not controlled by VCS, then log in as superuser on each system in the cluster and enter the following command:

```
# /etc/init.d/init.crs stop
```

- 5 Take offline all other VCS groups that depend on VxFS file systems or VxVM disk groups, whether local or CFS.

```
# hagrps -offline group_name -sys galaxy
```

- 6 Stop all applications that use CFS or VxVM, but are not under VCS control.

- 7** Make sure that no processes are running which make use of mounted shared file system or shared volumes.

```
# mount -v |grep vxfs  
  
# fuser -c /mount_point
```

- 8** Unmount all the VxFS file system which is not under VCS control.

```
# umount /mount_point
```

- 9** Deport any CVM disk groups that are not under VCS control.

- 10** With SF Oracle RAC 5.0 installed and configured, run the following command on the single node (galaxy) in the SF Oracle RAC cluster:

```
# hastop -local
```

- 11** For Oracle RAC 10g and Oracle RAC 11g, comment out the CRS-specific lines in the /etc/inittab file to prevent the system from starting the CRS daemons during system startup.

On each system, prefix the CRS lines with a colon (":") to comment them out.

For example:

```
h1:3:respawn:/etc/init.d/init.evmd run >/dev/null 2>&1 /null  
h2:3:respawn:/etc/init.d/init.cssd fatal >/dev/null 2>&1 /null  
h3:3:respawn:/etc/init.d/init.crsd run >/dev/null 2>&1 /null
```

- 12** For an optional port verification step, make sure only ports a, b, d, and o are open on galaxy by running the `gabconfig -a` command:

```
# gabconfig -a
```

```
GAB Port Memberships
```

```
=====
```

```
Port a gen 6b5901 membership 01
```

```
Port b gen 6b5904 membership 01
```

```
Port d gen 6b5907 membership 01
```

```
Port o gen 6b5905 membership 01
```

Run the `gabconfig -a` command on the other node(s) in the system . The following command output should be displayed:

```
# gabconfig -a
```

```
GAB Port Memberships
```

```
=====
```

```
Port a gen 6b5901 membership 01
```

```
Port b gen 6b5904 membership 01
```

```
Port d gen 6b5907 membership 01
```

```
Port f gen 6b5905 membership ;1
```

```
Port h gen 6b5908 membership ;1
```

```
Port o gen 6b5903 membership 01
```

```
Port v gen 6b5907 membership ;1
```

```
Port w gen 6b5909 membership ;1
```

Stopping and starting LLT and GAB

You can use the following procedures to stop and restart LLT and GAB modules.

See the *Veritas Storage Foundation Cluster File System Administrator's Guide* for procedures to start and stop the CFS drivers.

To stop LLT and GAB

- ◆ Run the following commands:

```
# /etc/init.d/odm stop
```

```
# /etc/init.d/gab stop
```

```
# /etc/init.d/llt stop
```

To start LLT and GAB

- ◆ Run the following commands:

```
# /etc/init.d/llt start
# /etc/init.d/gab start
# /etc/init.d/odm start
```

Adding LLT links to increase capacity

If you need additional capacity for Oracle communication on your private interconnects, you can add LLT links.

LLT links can be added or removed while clients are connected.

Refer to the `lltconfig` manual page for more information.

Note: When adding or removing LLT links, shutting down GAB or the high-availability daemon, HAD, is not required.

When adding or removing LLT links, your changes take effect immediately but will be lost on the next restart. For changes to persist you must also update `/etc/llttab`.

To add a new LLT link

- ◆ Enter the following command:

```
# lltnconfig -d device -t tag
```

To remove an LLT link

- ◆ Enter the following command:

```
# lltnconfig -u tag
```

Turning on the debug messages for the kernel drivers

To debug any issues that you encounter with the Veritas kernel drivers such as LLT, GAB, VxFEN, VxIO, or VxFS, you must turn on the debug messages.

To turn on the debug messages for the kernel drivers

- 1 On all the nodes, do the following:

```
# bosboot -aDd /dev/ipldevice
```

Refer to the `bosboot` manual page for more information.

- 2 Reboot the nodes.

```
# shutdown -r
```

Administering VCS

This section provides instructions for the following VCS administration tasks:

- [Viewing available Veritas devices and drivers](#)
- [Loading Veritas drivers into memory](#)
- [Configuring VCS to start Oracle with a specified Pfile](#)
- [Verifying VCS configuration](#)
- [Starting and stopping VCS](#)
- [Rebooting nodes in a cluster](#)

If you encounter issues while administering VCS, refer to the troubleshooting section for assistance.

See “[Troubleshooting VCSIPC](#)” on page 273.

Viewing available Veritas devices and drivers

To view the available Veritas devices in the Customized Devices object class:

```
# lsdev -C -c vxdrv
```

```
vxdump      Available  Veritas VxDMP Device Driver
vxglim0     Available  N/A
vxgms0      Available  N/A
vxio        Available  Veritas VxIO Device Driver
vxportal0   Available  VERITAS VxPORTAL Device Driver
vxqio0      Defined    VERITAS VxQIO Device Driver
vxspec      Available  Veritas VxSPEC Device Driver
```

To view the devices that are loaded in memory, run the `genkex` command as shown in the following examples.

If you want to view whether or not the driver 'gab' is loaded in memory:

```
# genkex |grep gab
457f000      4ed20 /usr/lib/drivers/gab
```

If you want to view whether or not the 'vx' drivers are loaded in memory:

```
# genkex |grep vx
55de000      2000 /etc/vx/kernel/dmpjbod
55dc000      2000 /etc/vx/kernel/dmpap
55da000      2000 /etc/vx/kernel/dmpaa
55b5000     21000 /usr/lib/drivers/vxodm.ext_61
55b2000      3000 /usr/lib/drivers/vxspec
4de0000     7ce000 /usr/lib/drivers/vxio
4d7a000     62000 /usr/lib/drivers/vxdmp
4cee000      3d000 /usr/lib/drivers/vxgms.ext_61
4c7e000      4d000 /usr/lib/drivers/vxfen
4bf3000       e000 /usr/lib/drivers/vxqio.ext_61
4bef000      3000 /usr/lib/drivers/vxportal.ext_61
4969000    232000 /usr/lib/drivers/vxfs.ext_61
48fa000      3e000 /usr/lib/drivers/vxglm.ext
```

Loading Veritas drivers into memory

Under normal operational conditions, you do not need to load Veritas drivers into memory. You might need to load a Veritas driver only if there is a malfunction.

To load the VxFS driver into memory:

```
# /etc/methods/vxkextadm vxfs load
```

Configuring VCS to start Oracle with a specified Pfile

If you want to configure VCS such that Oracle starts with a specified Pfile, modify the main.cf file for the Oracle group as follows:

```
Oracle Ora_1 (
    Sid @galaxy = PROD11
    Sid @nebula = PROD12
    Owner = oracle
    Home = "/oracle/orahome"
    Pfile @galaxy = "/oracle/VRT/dbs/initPROD11.ora"
    Pfile @nebula = "/oracle/VRT/dbs/initPROD12.ora"
    StartUpOpt = SRVCTLSTART
```

```
ShutDownOpt = SRVCTLSTOP  
OnlineTimeout = 900  
)
```

Verifying VCS configuration

To verify the VCS configuration:

```
# cd /etc/VRTSvcs/conf/config  
# hacf -verify .
```

Starting and stopping VCS

To start VCS on each node:

```
# hstart
```

To stop VCS on each node:

```
# hstop -local
```

You can also use the command "hstop -all"; however, make sure that you wait for port 'h' to close before restarting VCS.

Rebooting nodes in a cluster

If you need to reboot nodes in your cluster, use the `shutdown` command with the `-r` option. The `shutdown -r` command cleanly shuts down VCS and takes the service groups offline.

To reboot nodes:

```
# shutdown -r
```

Administering I/O fencing

This section describes I/O fencing and provides instructions for common I/O fencing administration tasks.

- [About I/O fencing](#)
- [How I/O fencing works in different event scenarios](#)
- [About I/O fencing utilities](#)
- [About vxfcntl utility](#)

- [About vxfenadm utility](#)
- [About vxfenclearpre utility](#)
- [About vxfenswap utility](#)

If you encounter issues while administering I/O fencing, refer to the troubleshooting section for assistance.

See [“Troubleshooting I/O fencing”](#) on page 261.

About I/O fencing

I/O fencing protects the data on shared disks when nodes in a cluster detect a change in the cluster membership that indicates a split brain condition.

The fencing operation determines the following:

- The nodes that must retain access to the shared storage
- The nodes that must be ejected from the cluster

This decision prevents possible data corruption. The `installsfrac` program installs the SF Oracle RAC I/O fencing driver, `VRTSvxfen`. If you want to protect data on shared disks, you must configure I/O fencing after you install and configure SF Oracle RAC.

See *Veritas Storage Foundation for Oracle RAC Installation and Configuration Guide*.

I/O fencing technology uses coordination points for arbitration in the event of a network partition.

See [“About preventing data corruption with I/O fencing”](#) on page 40.

About I/O fencing components

The shared storage for SF Oracle RAC must support SCSI-3 persistent reservations to enable I/O fencing. SF Oracle RAC involves two types of shared storage:

Data disks	Store shared data
Coordination points	Act as a global lock during membership changes

I/O fencing in SF Oracle RAC involves coordination points and data disks. Each component has a unique purpose and uses different physical disk devices. The fencing driver, known as `vxfen`, directs CVM as necessary to carry out actual fencing operations at the disk group level.

About data disks

Data disks are standard disk devices for data storage and are either physical disks or RAID Logical Units (LUNs). These disks must support SCSI-3 PR and are part of standard VxVM or CVM disk groups.

CVM is responsible for fencing data disks on a disk group basis. Disks that are added to a disk group and new paths that are discovered for a device are automatically fenced.

About coordination points

Coordination points provide a lock mechanism to determine which nodes get to fence off data drives from other nodes. A node must eject a peer from the coordination points before it can fence the peer from the data drives. Racing for control of the coordination points to fence data disks is the key to understand how fencing prevents split brain.

Disks that act as coordination points are called coordinator disks. Coordinator disks are three standard disks or LUNs set aside for I/O fencing during cluster reconfiguration. Coordinator disks do not serve any other storage purpose in the SF Oracle RAC configuration. Users cannot store data on these disks or include the disks in a disk group for user data. The coordinator disks can be any three disks that support SCSI-3 PR. Coordinator disks cannot be the special devices that array vendors use. For example, you cannot use EMC gatekeeper devices as coordinator disks.

Symantec recommends using the smallest possible LUNs for coordinator disks. Because coordinator disks do not store any data, cluster nodes need to only register with them and do not need to reserve them.

You can configure coordinator disks to use Veritas Volume Manager Dynamic Multipathing (DMP) feature. Dynamic Multipathing (DMP) allows coordinator disks to take advantage of the path failover and the dynamic adding and removal capabilities of DMP. So, you can configure I/O fencing to use either DMP devices or the underlying raw character devices. I/O fencing uses SCSI-3 disk policy that is either raw or dmp based on the disk device that you use. The disk policy is raw by default.

See the *Veritas Volume Manager Administrator's Guide*.

You can use iSCSI devices as coordinator disks for I/O fencing. However, I/O fencing supports iSCSI devices only when you use DMP disk policy. If you use iSCSI devices as coordinator disks, make sure that the `/etc/vxfsenmode` file has the disk policy set to DMP.

For the latest information on supported hardware visit the following URL:

<http://entsupport.symantec.com/docs/283161>

How I/O fencing works in different event scenarios

[Table 3-1](#) describes how I/O fencing works to prevent data corruption in different failure event scenarios. For each event, review the corrective operator actions.

Table 3-1 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 coordinator disks. If Node A wins race for coordinator disks, Node A ejects Node B from the shared disks and continues.	Node B races for majority of coordinator disks. If Node B loses the race for the coordinator disks, 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.

Table 3-1 I/O fencing scenarios (continued)

Event	Node A: What happens?	Node B: What happens?	Operator action
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 coordinator disks. Node B wins race for coordinator disks and ejects Node A from shared data disks.	Verify private networks function and restart Node A.

Table 3-1 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. Coordinator 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 coordinator disks. 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 coordinator disks. 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 “System panics to prevent potential data corruption” on page 262.

Table 3-1 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 coordinator disks. 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 “ System panics to prevent potential data corruption ” on page 262.
The disk array containing two of the three coordinator disks is powered off. Node B leaves the cluster and the disk array is still powered off.	Node A continues to operate as long as no nodes leave the cluster. Node A races for a majority of coordinator disks. Node A fails because only one of three coordinator disks is available. Node A panics and removes itself from the cluster.	Node B continues to operate as long as no nodes leave the cluster. Node B leaves the cluster.	Power on failed disk array and restart I/O fencing driver to enable Node A to register with all coordinator disks.

About I/O fencing utilities

The I/O fencing feature provides the following utilities that are available through the VRTSvxfen package:

vxfcntlsthdw	Tests hardware for I/O fencing Path: /opt/VRTSvcs/vxfen/bin/vxfcntlsthdw See “About vxfcntlsthdw utility” on page 71.
vxfenconfig	Configures and unconfigures I/O fencing Checks the list of coordinator disks used by the vxfen driver. Path: /sbin/vxfenconfig
vxfenadm	Displays information on I/O fencing operations and manages SCSI-3 disk registrations and reservations for I/O fencing Path: /sbin/vxfenadm See “About vxfenadm utility” on page 80.
vxfenclearpre	Removes SCSI-3 registrations and reservations from disks Path: /opt/VRTSvcs/vxfen/bin/vxfenclearpre See “About vxfenclearpre utility” on page 82.
vxfenswap	Replaces coordinator disks without stopping I/O fencing Path: /opt/VRTSvcs/vxfen/bin/vxfenswap See “About vxfenswap utility” on page 84.
vxferdisk	Generates the list of paths of disks in the diskgroup. This utility requires that Veritas Volume Manager is installed and configured. Path: /opt/VRTSvcs/vxfen/bin/vxferdisk

Refer to the corresponding manual page for more information on the commands.

About vxfcntlsthdw utility

You can use the vxfcntlsthdw utility to verify that shared storage arrays to be used for data support SCSI-3 persistent reservations and I/O fencing. During the I/O fencing configuration, the testing utility is used to test a single disk. The utility has other options that may be more suitable for testing storage devices in other configurations. You also need to test coordinator disk groups.

See *Veritas Storage Foundation for Oracle RAC Installation and Configuration Guide* to set up I/O fencing.

The utility, which you can run from one system in the cluster, tests the storage used for data by setting and verifying SCSI-3 registrations on the disk or disks you specify, setting and verifying persistent reservations on the disks, writing data to the disks and reading it, and removing the registrations from the disks. Refer also to the `vxfcntlsthdw(1M)` manual page.

About general guidelines for using `vxfcntlsthdw` utility

Review the following guidelines to use the `vxfcntlsthdw` utility:

- The utility requires two systems connected to the shared storage.

Caution: The tests overwrite and destroy data on the disks, unless you use the `-r` option.

- The two nodes must have `ssh` (default) or `rsh` communication. If you use `rsh`, launch the `vxfcntlsthdw` utility with the `-n` option.
After completing the testing process, you can remove permissions for communication and restore public network connections.
- To ensure both systems are connected to the same disk during the testing, you can use the `vxflenadm -i diskpath` command to verify a disk's serial number. See [“Verifying that the nodes see the same disk”](#) on page 82.
- For disk arrays with many disks, use the `-m` option to sample a few disks before creating a disk group and using the `-g` option to test them all.
- When testing many disks with the `-f` or the `-g` option, you can review results by redirecting the command output to a file.
- The utility indicates a disk can be used for I/O fencing with a message resembling:

```
The disk /dev/rdisk/clt1d0s2 is ready to be configured for
I/O Fencing on node galaxy
```

If the utility does not show a message stating a disk is ready, verification has failed.

- If the disk you intend to test has existing SCSI-3 registration keys, the test issues a warning before proceeding.

About the `vxfcntlsthdw` command options

[Table 3-2](#) describes the methods that the utility provides to test storage devices.

Table 3-2 vxfststhdw options

vxfststhdw option	Description	When to use
-n	Utility uses rsh for communication.	Use when rsh is used for communication.
-r	Non-destructive testing. Testing of the disks for SCSI-3 persistent reservations occurs in a non-destructive way; that is, there is only testing for reads, not writes. May be used with -m, -f, or -g options.	Use during non-destructive testing. See “Performing non-destructive testing on the disks using the -r option” on page 76.
-t	Testing of the return value of SCSI TEST UNIT (TUR) command under SCSI-3 reservations. A warning is printed on failure of TUR testing.	When you want to perform TUR testing.
-d	Use DMP devices. May be used with -c or -g options.	With the -d option, the script picks the DMP paths for disks in the diskgroup. By default, the script uses the -w option to pick up the OS paths for disks in the disk group.
-w	Use raw devices. May be used with -c or -g options.	By default, the script picks up the raw paths for disks in the diskgroup. If you want the script to use the raw paths for disks in the diskgroup, use the -d option.
-c	Utility tests the coordinator disk group prompting for systems and devices, and reporting success or failure.	For testing disks in coordinator disk group. See “Testing the coordinator disk group using vxfststhdw -c option” on page 74.
-m	Utility runs manually, in interactive mode, prompting for systems and devices, and reporting success or failure. May be used with -r and -t options. -m is the default option.	For testing a few disks or for sampling disks in larger arrays. See “Testing the shared disks using the vxfststhdw -m option” on page 76.

Table 3-2 vxfststhdw options (continued)

vxfststhdw option	Description	When to use
-f <i>filename</i>	Utility tests system/device combinations listed in a text file. May be used with -r and -t options.	For testing several disks. See “Testing the shared disks listed in a file using the vxfststhdw -f option” on page 78.
-g <i>disk_group</i>	Utility tests all disk devices in a specified disk group. May be used with -r and -t options.	For testing many disks and arrays of disks. Disk groups may be temporarily created for testing purposes and destroyed (ungrouped) after testing. See “Testing all the disks in a disk group using the vxfststhdw -g option” on page 79.

Testing the coordinator disk group using vxfststhdw -c option

Use the vxfststhdw utility to verify disks are configured to support I/O fencing. In this procedure, the vxfststhdw utility tests the three disks one disk at a time from each node.

The procedure in this section uses the following disks for example:

- From the node galaxy, the disks are /dev/rdisk/c1t1d0s2, /dev/rdisk/c2t1d0s2, and /dev/rdisk/c3t1d0s2.
- From the node nebula, the same disks are seen as /dev/rdisk/c4t1d0s2, /dev/rdisk/c5t1d0s2, and /dev/rdisk/c6t1d0s2.

Note: To test the coordinator disk group using the vxfststhdw utility, the utility requires that the coordinator disk group, vxfencoordg, be accessible from two nodes.

To test the coordinator disk group using vxfentsthdw -c

- 1 Use the vxfentsthdw command with the -c option. For example:

```
# /opt/VRTSvcs/vxfen/bin/vxfentsthdw -c vxfencoorddg
```

- 2 Enter the nodes you are using to test the coordinator disks:

Enter the first node of the cluster: **galaxy**

Enter the second node of the cluster: **nebula**

- 3 Review the output of the testing process for both nodes for all disks in the coordinator disk group. Each disk should display output that resembles:

```
ALL tests on the disk /dev/rdisk/clt1d0s2 have PASSED.
```

```
The disk is now ready to be configured for I/O Fencing on node  
galaxy as a COORDINATOR DISK.
```

```
ALL tests on the disk /dev/rdisk/c4t1d0s2 have PASSED.
```

```
The disk is now ready to be configured for I/O Fencing on node  
nebula as a COORDINATOR DISK.
```

- 4 After you test all disks in the disk group, the vxfencoorddg disk group is ready for use.

Removing and replacing a failed disk

If a disk in the coordinator disk group fails verification, remove the failed disk or LUN from the vxfencoorddg disk group, replace it with another, and retest the disk group.

To remove and replace a failed disk

- 1 Use the `vxdiskadm` utility to remove the failed disk from the disk group.
Refer to the *Veritas Volume Manager Administrator's Guide*.
- 2 Add a new disk to the node, initialize it, and add it to the coordinator disk group.
See the *Veritas Storage Foundation for Oracle RAC Installation and Configuration Guide* for instructions to initialize disks for I/O fencing and to set up coordinator disk groups.
If necessary, start the disk group.
See the *Veritas Volume Manager Administrator's Guide* for instructions to start the disk group.
- 3 Retest the disk group.
See [“Testing the coordinator disk group using `vxfcntlsthdw -c` option”](#) on page 74.

Performing non-destructive testing on the disks using the `-r` option

You can perform non-destructive testing on the disk devices when you want to preserve the data.

To perform non-destructive testing on disks

- ◆ To test disk devices containing data you want to preserve, you can use the `-r` option with the `-m`, `-f`, or `-g` options.

For example, to use the `-m` option and the `-r` option, you can run the utility as follows:

```
# /opt/VRTSvcs/vxfen/bin/vxfentsthdw -rm
```

When invoked with the `-r` option, the utility does not use tests that write to the disks. Therefore, it does not test the disks for all of the usual conditions of use.

Testing the shared disks using the `vxfcntlsthdw -m` option

Review the procedure to test the shared disks. By default, the utility uses the `-m` option.

This procedure uses the `/dev/rdisk/c1t1d0s2` disk in the steps.

If the utility does not show a message stating a disk is ready, verification has failed. Failure of verification can be the result of an improperly configured disk array. It can also be caused by a bad disk.

If the failure is due to a bad disk, remove and replace it. The `vxfcntlsthdw` utility indicates a disk can be used for I/O fencing with a message resembling:

```
The disk /dev/rdisk/clt1d0s2 is ready to be configured for
I/O Fencing on node galaxy
```

Note: For A/P arrays, run the `vxfcntlsthdw` command only on secondary paths.

To test disks using `vxfcntlsthdw` script

- 1 Make sure system-to-system communication is functioning properly.
- 2 From one node, start the utility.

```
# /opt/VRTSvcs/vxfen/bin/vxfcntlsthdw [-n]
```

- 3 After reviewing the overview and warning that the tests overwrite data on the disks, confirm to continue the process and enter the node names.

```
***** WARNING!!!!!!!!!! *****
THIS UTILITY WILL DESTROY THE DATA ON THE DISK!!

Do you still want to continue : [y/n] (default: n) y
Enter the first node of the cluster: galaxy
Enter the second node of the cluster: nebula
```

- 4 Enter the names of the disks you are checking. For each node, the disk may be known by the same name:

```
Enter the disk name to be checked for SCSI-3 PGR on node
galaxy in the format: /dev/rdisk/cxtxdxsx
/dev/rdisk/c2t13d0s2
Enter the disk name to be checked for SCSI-3 PGR on node
nebula in the format: /dev/rdisk/cxtxdxsx
Make sure it's the same disk as seen by nodes galaxy and nebula
/dev/rdisk/c2t13d0s2
```

If the serial numbers of the disks are not identical, then the test terminates.

- 5 Review the output as the utility performs the checks and report its activities.

- 6 If a disk is ready for I/O fencing on each node, the utility reports success:

```
ALL tests on the disk /dev/rdisk/clt1d0s2 have PASSED
The disk is now ready to be configured for I/O Fencing on node
galaxy
```

- 7 Run the `vxfcntlsthdw` utility for each disk you intend to verify.

Testing the shared disks listed in a file using the `vxfcntlsthdw -f` option

Use the `-f` option to test disks that are listed in a text file. Review the following example procedure.

To test the shared disks listed in a file

- 1 Create a text file `disks_test` to test two disks shared by systems `galaxy` and `nebula` that might resemble:

```
galaxy /dev/rdisk/c2t2d1s2 nebula /dev/rdisk/c3t2d1s2
galaxy /dev/rdisk/c2t2d1s2 nebula /dev/rdisk/c3t2d1s2
```

where the first disk is listed in the first line and is seen by `galaxy` as `/dev/rdisk/c2t2d1s2` and by `nebula` as `/dev/rdisk/c3t2d1s2`. The other disk, in the second line, is seen as `/dev/rdisk/c2t2d2s2` from `galaxy` and `/dev/rdisk/c3t2d2s2` from `nebula`. Typically, the list of disks could be extensive.

- 2 To test the disks, enter the following command:

```
# /opt/VRTSvcs/vxfen/bin/vxfentsthdw -f disks_test
```

The utility reports the test results one disk at a time, just as for the `-m` option.

- 3 To redirect the test results to a text file, enter the following command:

```
# /opt/VRTSvcs/vxfen/bin/vxfentsthdw -f\
disks_test > test_disks.txt
```

Caution: Be advised that by redirecting the command's output to a file, a warning that the testing destroys data on the disks cannot be seen until the testing is done.

Precede the command with “yes” to acknowledge that the testing destroys any data on the disks to be tested.

For example:

```
# yes | /opt/VRTSvcs/vxfen/bin/vxfentsthdw -f\
disks_blue > blue_test.txt
```

Testing all the disks in a disk group using the `vxfentsthdw -g` option

Use the `-g` option to test all disks within a disk group. For example, you create a temporary disk group consisting of all disks in a disk array and test the group.

Note: Do not import the test disk group as shared; that is, do not use the `-s` option.

After testing, destroy the disk group and put the disks into disk groups as you need.

To test all the disks in a diskgroup

- 1 Create a diskgroup for the disks that you want to test.
- 2 Enter the following command to test the diskgroup `test_disks_dg`:

```
# /opt/VRTSvcs/vxfen/bin/vxfentsthdw -g test_disks_dg
```

The utility reports the test results one disk at a time.

- 3 To redirect the test results to a text file for review, enter the following command:

```
# /opt/VRTSvcs/vxfen/bin/vxfentsthdw -g \
test_disks_dg > dgtestdisks.txt
```

Testing a disk with existing keys

If the utility detects that a coordinator disk has existing keys, you see a message that resembles:

```
There are Veritas I/O fencing keys on the disk. Please make sure
that I/O fencing is shut down on all nodes of the cluster before
continuing.
```

```
***** WARNING!!!!!!!!!! *****
```

```
THIS SCRIPT CAN ONLY BE USED IF THERE ARE NO OTHER ACTIVE NODES
IN THE CLUSTER! VERIFY ALL OTHER NODES ARE POWERED OFF OR
INCAPABLE OF ACCESSING SHARED STORAGE.
```

If this is not the case, data corruption will result.

Do you still want to continue : [y/n] (default: n) **y**

The utility prompts you with a warning before proceeding. You may continue as long as I/O fencing is not yet configured.

About vxfenadm utility

Administrators can use the `vxfenadm` command to troubleshoot and test fencing configurations.

The command's options for use by administrators are as follows:

-g	read and display keys
-i	read SCSI inquiry information from device
-m	register with disks
-n	make a reservation with disks
-p	remove registrations made by other systems
-r	read reservations
-x	remove registrations

Refer to the `vxfenadm(1m)` manual page for a complete list of the command options.

About the registration key format

The key defined by VxVM associated with a disk group consists of seven bytes maximum. This key becomes unique among the systems when the VxVM prefixes it with the ID of the system. The key used for I/O fencing, therefore, consists of eight bytes.

[Figure 3-1](#) depicts the format of the key.

Figure 3-1 I/O fencing key format

0							7
Node ID	VxVM defined	VxVM defined	VxVM defined	VxVM defined	VxVM defined	VxVM defined	VxVM defined

The keys currently assigned to disks can be displayed by using the `vxfenadm` command.

For example, from the system with node ID 1, display the key for the disk `/dev/rdisk/c2t1d0s2` by entering:

```
# vxfenadm -g /dev/rdisk/c2t1d0s2
Reading SCSI Registration Keys...
Device Name: /dev/rdisk/c2t1d0s2
Total Number of Keys: 1
key[0]:
    Key Value [Numeric Format]: 65,45,45,45,45,45,45,45
    Key Value [Character Format]: A-----
```

The `-g` option of `vxfenadm` displays all eight bytes of a key value in two formats. In the numeric format, the first byte, representing the Node ID, contains the

system ID 65. The remaining bytes contain the ASCII values of the letters of the key, in this case, “-----.” In the next line, the node ID 0 is expressed as “A;” node ID 1 would be “B.”

Verifying that the nodes see the same disk

To confirm whether a disk (or LUN) supports SCSI-3 persistent reservations, two nodes must simultaneously have access to the same disks. Because a shared disk is likely to have a different name on each node, check the serial number to verify the identity of the disk. Use the `vxfenadm` command with the `-i` option to verify that the same serial number for the LUN is returned on all paths to the LUN.

For example, an EMC disk is accessible by the `/dev/rdisk/c2t13d0s2` path on node A and the `/dev/rdisk/c2t11d0s2` path on node B.

To verify that the nodes see the same disks

- 1 Verify the connection of the shared storage for data to two of the nodes on which you installed SF Oracle RAC.
- 2 From node A, enter the following command:

```
# vxfenadm -i /dev/rdisk/c2t13d0s2

Vendor id      : EMC
Product id     : SYMMETRIX
Revision       : 5567
Serial Number  : 42031000a
```

The same serial number information should appear when you enter the equivalent command on node B using the `/dev/rdisk/c2t11d0s2` path.

On a disk from another manufacturer, Hitachi Data Systems, the output is different and may resemble:

```
# vxfenadm -i /dev/rdisk/c2t11d0s2

Vendor id      : HITACHI
Product id     : OPEN-3      -SUN
Revision       : 0117
Serial Number  : 0401EB6F0002
```

Refer to the `vxfenadm(1M)` manual page.

About vxfenclearpre utility

You can use the `vxfenclearpre` utility to remove SCSI-3 registrations and reservations on the disks.

See [“Removing preexisting keys”](#) on page 83.

Removing preexisting keys

If you encountered a split brain condition, use the `vxfenclearpre` utility to remove SCSI-3 registrations and reservations on the coordinator disks as well as on the data disks in all shared disk groups.

You can also use this procedure to remove the registration and reservation keys created by another node from a disk.

To clear keys after split brain

- 1 Stop VCS and I/O fencing on all nodes.

```
# 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:

Enter the following command on each node:

```
# /etc/init.d/vxfen stop
```

If you have any applications that run outside of VCS control that have access to the shared storage, then shut down all other nodes in the cluster that have access to the shared storage. This prevents data corruption.

- 2 Start the script:

```
# cd /opt/VRTSvcs/vxfen/bin  
# ./vxfenclearpre
```

- 3 Read the script's introduction and warning. Then, you can choose to let the script run.

```
Do you still want to continue: [y/n] (default : n) y
```

Informational messages resembling the following may appear on the console of one of the nodes in the cluster when a node is ejected from a disk/LUN:

```
<date> <system name> scsi: WARNING: /sbus@3,0/lpfs@0,0/
sd@0,1(sd91):
<date> <system name> Error for Command: <undecoded
cmd 0x5f> Error Level: Informational
<date> <system name> scsi: Requested Block: 0 Error Block 0
<date> <system name> scsi: Vendor: <vendor> Serial Number:
0400759B006E
<date> <system name> scsi: Sense Key: Unit Attention
<date> <system name> scsi: ASC: 0x2a (<vendor unique code
0x2a>), ASCQ: 0x4, FRU: 0x0
```

These informational messages may be ignored.

```
Cleaning up the coordinator disks...
```

```
Cleaning up the data disks for all shared disk groups...
```

```
Successfully removed SCSI-3 persistent registration and
reservations from the coordinator disks as well as the
shared data disks.
```

```
Reboot the server to proceed with normal cluster startup...
#
```

- 4 Restart all nodes in the cluster.

About vxfsnwap utility

The vxfsnwap utility allows you to replace coordinator disks in a cluster that is online. The utility verifies that the serial number of the new disks are identical on all the nodes and the new disks can support I/O fencing.

Refer to the vxfsnwap(1M) manual page.

See *Veritas Storage Foundation for Oracle RAC Installation and Configuration Guide* for details on the coordinator disk requirements.

You can replace the coordinator disks without stopping I/O fencing in the following cases:

- The disk becomes defective or inoperable and you want to switch to a new diskgroup.
See [“Replacing I/O fencing coordinator disks when the cluster is online”](#) on page 85.
See [“Replacing the coordinator diskgroup in a cluster that is online”](#) on page 88.
If you want to replace the coordinator disks when the cluster is offline, you cannot use the `vxfsnwap` utility. You must manually perform the steps that the utility does to replace the coordinator disks.
See [“Replacing defective disks when the cluster is offline”](#) on page 264.
- You want to switch the disk interface between raw devices and DMP devices.
See [“Changing the disk interaction policy in a cluster that is online”](#) on page 89.
- The keys that are registered on the coordinator disks are lost.
In such a case, the cluster might panic when a split-brain occurs. You can replace the coordinator disks with the same disks using the `vxfsnwap` command. During the disk replacement, the missing keys register again without any risk of data corruption.
See [“Refreshing lost keys on coordinator disks”](#) on page 90.

If the `vxfsnwap` operation is unsuccessful, then you can use the `vxfsnwap -a cancel` command to manually roll back the changes that the `vxfsnwap` utility does. You must run this command if a node fails during the process of disk replacement, or if you aborted the disk replacement.

Replacing I/O fencing coordinator disks when the cluster is online

Review the procedures to add, remove, or replace one or more coordinator disks in a cluster that is operational.

Warning: The cluster might panic if any node leaves the cluster membership before the `vxfsnwap` script replaces the set of coordinator disks.

To replace a disk in a coordinator diskgroup when the cluster is online

- 1 Make sure system-to-system communication is functioning properly.
- 2 Make sure that the cluster is online.

```
# vxfsadm -d
I/O Fencing Cluster Information:
=====
Fencing Protocol Version: 201
Fencing Mode: SCSI3
Fencing SCSI3 Disk Policy: dmp
Cluster Members:
    * 0 (galaxy)
    1 (nebula)
RFSM State Information:
    node 0 in state 8 (running)
    node 1 in state 8 (running)
```

- 3 Import the coordinator disk group.

The file /etc/vxfendg includes the name of the disk group (typically, vxfscoorddg) that contains the coordinator disks, so use the command:

```
# vxfs -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.

- 4 To remove disks from the disk group, use the VxVM disk administrator utility vxfsadm.

You may also destroy the existing coordinator disk group. For example:

- Verify whether the coordinator attribute is set to on.

```
# vxfs list vxfscoorddg | grep flags: | grep coordinator
```

- If the coordinator attribute value is set to on, you must turn off this attribute for the coordinator disk group.

```
# vxfs -g vxfscoorddg set coordinator=off
```

- Destroy the disk group.

```
# vxdg destroy vxfencoorddg
```

- 5 Add the new disk to the node, initialize it as a VxVM disk, and add it to the vxfencoorddg disk group.

See the *Veritas Storage Foundation for Oracle RAC Installation and Configuration Guide* for detailed instructions.

Note that though the diskgroup content changes, the I/O fencing remains in the same state.

- 6 Make sure that the `/etc/vxfenmode` file is updated to specify the correct disk policy.

See the *Veritas Storage Foundation for Oracle RAC Installation and Configuration Guide* for more information.

- 7 From one node, start the vxfenswap utility. You must specify the diskgroup to the utility.

Do one of the following:

- If you use ssh for communication:

```
# /opt/VRTSvcs/vxfen/bin/vxfenswap -g diskgroup
```

- If you use rsh for communication:

```
# /opt/VRTSvcs/vxfen/bin/vxfenswap -g diskgroup -n
```

The utility performs the following tasks:

- Backs up the existing `/etc/vxfentab` file.
 - Creates a test file `/etc/vxfentab.test` for the diskgroup that is modified on each node.
 - Reads the diskgroup you specified in the vxfenswap command and adds the diskgroup to the `/etc/vxfentab.test` file on each node.
 - Verifies that the serial number of the new disks are identical on all the nodes. The script terminates if the check fails.
 - Verifies that the new disks can support I/O fencing on each node.
- 8 If the disk verification passes, the utility reports success and asks if you want to commit the new set of coordinator disks.

```
Do you wish to commit this change? [y/n] (default: n) y
```

- 9 Review the message that the utility displays and confirm that you want to replace the diskgroup. Else skip to step 10.

If the utility successfully commits, the utility moves the `/etc/vxfentab.test` file to the `/etc/vxfentab` file.

If you specified a diskgroup different to that in the `/etc/vxfendg` file, the utility also updates the `/etc/vxfendg` file.

- 10 If you do not want to replace the diskgroup, answer n.

The `vxfsenwap` utility rolls back any changes to the coordinator diskgroup.

Replacing the coordinator diskgroup in a cluster that is online

You can also replace the coordinator diskgroup using the `vxfsenwap` utility.

To replace the coordinator diskgroup

- 1 Make sure system-to-system communication is functioning properly.
- 2 Make sure that the cluster is online.

```
# vxfsenadm -d
I/O Fencing Cluster Information:
=====
Fencing Protocol Version: 201
Fencing Mode: SCSI3
Fencing SCSI3 Disk Policy: dmp
Cluster Members:
  * 0 (galaxy)
  1 (nebula)
RFSM State Information:
  node 0 in state 8 (running)
  node 1 in state 8 (running)
```

- 3 Find the name of the current coordinator diskgroup (typically `vxfsencoorddg`) that is in the `/etc/vxfendg` file.

```
# cat /etc/vxfendg
vxfsencoorddg
```


- 4 Find the alternative disk groups available to replace the current coordinator diskgroup.

```
# vxdisk -o alldgs list
```

DEVICE	TYPE	DISK	GROUP	STATUS
c2t0d0s2	auto:LVM	- -	LVM	
c2t1d0s2	auto:LVM	- -	LVM	
c4t0d0	auto:LVM	- -	LVM	
c4t0d1	auto:cdsdisk	-	(vxfendg)	online
c4t0d2	auto:cdsdisk	-	(vxfendg)	online
c4t0d3	auto:cdsdisk	-	(vxfendg)	online
c4t0d4	auto:cdsdisk	-	(vxfencoordg)	online
c4t0d5	auto:cdsdisk	-	(vxfencoordg)	online
c4t0d6	auto:cdsdisk	-	(vxfencoordg)	online

- 5 From any node, start the vxfenswap utility. For example, if vxfendg is the new diskgroup that you want to use as the coordinator diskgroup:

```
# /opt/VRTSvcs/vxfen/bin/vxfenswap -g vxfendg [-n]
```

- 6 Verify that the coordinator disk group has changed.

```
# cat /etc/vxfendg  
vxfendg
```

Changing the disk interaction policy in a cluster that is online

In a cluster that is online, you can change the disk interaction policy from raw to dmp using the vxfenswap utility.

To change the disk interaction policy

- 1 Make sure system-to-system communication is functioning properly.
- 2 Make sure that the cluster is online.

```
# vxfenadm -d
I/O Fencing Cluster Information:
=====
Fencing Protocol Version: 201
Fencing Mode: SCSI3
Fencing SCSI3 Disk Policy: dmp
Cluster Members:
    * 0 (galaxy)
    1 (nebula)
RFSM State Information:
    node 0 in state 8 (running)
    node 1 in state 8 (running)
```

- 3 On each node in the cluster, edit the /etc/vxfenmode file to change the disk policy.

```
# cat /etc/vxfenmode
vxfen_mode=scsi3
scsi3_disk_policy=raw
```

- 4 From any node, start the vxfenswap utility:

```
# /opt/VRTSvcs/vxfen/bin/vxfenswap -g vxfencoordg [-n]
```

- 5 Verify the change in the disk policy.

```
# vxfenadm -d
I/O Fencing Cluster Information:
=====
Fencing Protocol Version: 201
Fencing Mode: SCSI3
Fencing SCSI3 Disk Policy: raw
```

Refreshing lost keys on coordinator disks

If the coordinator disks lose the keys that are registered, the cluster might panic when a split-brain occurs.

You can use the `vxfsnwap` utility to replace the coordinator disks with the same disks. The `vxfsnwap` utility registers the missing keys during the disk replacement.

To refresh lost keys on coordinator disks

- 1 Make sure system-to-system communication is functioning properly.
- 2 Make sure that the cluster is online.

```
# vxfsnadm -d
I/O Fencing Cluster Information:
=====
Fencing Protocol Version: 201
Fencing Mode: SCSI3
Fencing SCSI3 Disk Policy: dmp
Cluster Members:
  * 0 (galaxy)
  1 (nebula)
RFSM State Information:
  node 0 in state 8 (running)
  node 1 in state 8 (running)
```

- 3 Run the following command to view the coordinator disks that do not have keys:

```
# vxfsnadm -g all -f /etc/vxfentab
Device Name: /dev/vx/rdmp/clt1d0s2
Total Number of Keys: 0
No keys...
...
```

- 4 On any node, run the following command to start the `vxfsnwap` utility:

```
# /opt/VRTSvcs/vxfen/bin/vxfsnwap -g vxfsncoorddg [-n]
```

- 5 Verify that the keys are atomically placed on the coordinator disks.

```
# vxfsnadm -g all -f /etc/vxfentab
Device Name: /dev/vx/rdmp/clt1d0s2
Total Number of Keys: 4
...
```

Administering CFS

This section provides instructions for the following CFS administration tasks:

- [Adding CFS file systems to VCS configuration](#)
- [Using cfsmount to mount CFS file systems](#)
- [Resizing CFS file systems](#)
- [Verifying the status of CFS file systems](#)
- [Verifying CFS port](#)
- [CFS agent log files](#)

If you encounter issues while administering CFS, refer to the troubleshooting section for assistance.

See [“Troubleshooting CFS”](#) on page 272.

Adding CFS file systems to VCS configuration

To add a CFS file system to the VCS main.cf file without using an editor:

```
# cfsmntadm add galaxy_ocr ocrvotevol /ocrvote grp10g \  
all=suid,rw
```

Mount Point is being added...

/ocrvote added to the cluster-configuration

See the *Veritas Storage Foundation Cluster File System Administrator's Guide* for more information on the command.

Using cfsmount to mount CFS file systems

To mount a CFS file system using cfsmount:

```
# cfsmount /ocrvote  
Mounting...  
[/dev/vx/dsk/galaxy_ocr/ocrvotevol]  
mounted successfully at /ocrvote on galaxy  
[/dev/vx/dsk/galaxy_ocr/ocrvotevol]  
mounted successfully at /ocrvote on nebula
```

See the *Veritas Storage Foundation Cluster File System Administrator's Guide* for more information on the command.

Resizing CFS file systems

If you see a message on the console indicating that a CFS file system is full, you may want to resize the file system. The `vxresize` command lets you resize a CFS file system.

The following command resizes an Oracle binary CFS file system (the Oracle binary volume is CFS mounted):

```
# vxresize -g galaxy_ora ora_vol +2G
```

where `galaxy_ora` is the CVM disk group, `ora_vol` is the volume, and `+2G` indicates the increase in volume by 2 Gigabytes.

Verifying the status of CFS file systems

Run the "cfscluster status" command to see the status of the nodes and their mount points:

```
# cfscluster status
```

```
Node           : nebula
Cluster Manager : not-running
CVM state       : not-running

MOUNT POINT    SHARED VOLUME  DISK GROUP      STATUS
/ocrvote       ocrvotevol     galaxy_ocr      NOT MOUNTED
/oracle        ora_vol        galaxy_ora      NOT MOUNTED
/crshome       ora_crs_vol    galaxy_crs      NOT MOUNTED
/oradata1      ora_data1_vol  galaxy_data1    NOT MOUNTED
/arch          archivol       galaxy_data1    NOT MOUNTED
```

```
Node           : galaxy
Cluster Manager : running
CVM state       : running

MOUNT POINT    SHARED VOLUME  DISK GROUP      STATUS
/ocrvote       ocrvotevol     galaxy_ocr      MOUNTED
/oracle        ora_vol        galaxy_ora      MOUNTED
/crshome       ora_crs_vol    galaxy_crs      MOUNTED
/oradata1      ora_data1_vol  galaxy_data1    MOUNTED
/arch          archivol       galaxy_data1    MOUNTED
```

Verifying CFS port

CFS uses port 'f' for communication between nodes. The CFS port state can be verified as follows:

```
# gabconfig -a | grep "Port f"
```

CFS agent log files

You can use the CFS agent log files located in the directory /var/VRTSvcs/log to debug CFS issues.

```
# cd /var/VRTSvcs/log
# ls
CFSMount_A.log
CFSfsckd_A.log
engine_A.log
```

The agent framework information is located in the engine_A.log file while the agent entry point information is located in the CFSMount_A.log and CFSfsckd_A.log files.

Administering CVM

This section provides instructions for the following CVM administration tasks:

- [Listing all the CVM shared disks](#)
- [Establishing CVM cluster membership manually](#)
- [Manually importing a shared disk group](#)
- [Manually deporting a shared disk group](#)
- [Manually starting shared volumes](#)
- [Evaluating the state of CVM ports](#)
- [Verifying if CVM is running in an SF Oracle RAC cluster](#)
- [Verifying CVM membership state](#)
- [Verifying state of CVM shared disk groups](#)
- [Verifying activation mode](#)
- [CVM log files](#)

If you encounter issues while administering CVM, refer to the troubleshooting section for assistance.

See [“Troubleshooting CVM”](#) on page 267.

Listing all the CVM shared disks

You can use the following command to list all the CVM shared disks:

```
# vxdisk -o alldgs list |grep shared
```

Establishing CVM cluster membership manually

Run the following command to start CVM manually:

```
# vxclustadm -m vcs -t gab startnode
```

```
vxclustadm: initialization completed
```

Note that `vxclustadm` reads `main.cf` for cluster configuration information and is therefore not dependent upon VCS to be running. You do not need to run the `vxclustadm startnode` command as normally the `hastart` (VCS start) command starts CVM automatically.

Manually importing a shared disk group

You can use the following command to manually import a shared disk group:

```
# vxdg -s import diskgroupname
```

Manually deporting a shared disk group

You can use the following command to manually deport a shared disk group:

```
# vxdg deport diskgroupname
```

Note that the deport of a shared disk group removes the SCSI-3 PGR keys on the disks. It also removes the ‘shared’ flag on the disks.

Manually starting shared volumes

Following a manual CVM shared disk group import, the volumes in the disk group need to be started manually, as follows:

```
# vxvol -g diskgroupname startall
```

In order to verify that the volumes are started, run the following command:

```
# vxprint -htrg diskgroupname | grep ^v
```

Evaluating the state of CVM ports

CVM kernel (vxio driver) uses port 'v' for kernel messaging and port 'w' for vxconfigd communication between the cluster nodes. The following command displays the state of CVM ports:

```
# gabconfig -a | egrep "Port v|Port w"
```

Verifying if CVM is running in an SF Oracle RAC cluster

You can use the following options to verify whether CVM is up or not in an SF Oracle RAC cluster.

The following output is displayed on a node that is not a member of the cluster:

```
# vxdctl -c mode
mode: enabled: cluster inactive

# vxclustadm -v nodestate
state: out of cluster
```

On the master node, the following output is displayed:

```
# vxdctl -c mode
mode: enabled: cluster active - MASTER
```

On the slave nodes, the following output is displayed:

```
# vxdctl -c mode
mode: enabled: cluster active - SLAVE
```

The following command allows you to view all the CVM nodes at the same time:

```
# vxclustadm nidmap
```

Name	CVM Nid	CM Nid	State
galaxy	0	0	Joined: Master
nebula	1	1	Joined: Slave

Verifying CVM membership state

The state of CVM can be verified as follows:

```
# vxclustadm -v nodestate

state: joining
      nodeId=0
      masterId=0
```



```
neighborId=0
members=0x1
joiners=0x0
leavers=0x0
reconfig_seqnum=0x0
reconfig: vxconfigd in join
```

The state indicates that CVM has completed its kernel level join and is in the middle of vxconfigd level join.

The `vxctl -c mode` command indicates whether a node is a CVM master or CVM slave.

Verifying state of CVM shared disk groups

You can use the following command to list the shared disk groups currently imported in the SF Oracle RAC cluster:

```
# vxdbg list |grep shared
orabinvol_dg enabled,shared 1052685125.1485.csha3
```

Verifying activation mode

In an SF Oracle RAC cluster, the activation of shared disk group should be set to “shared-write” on each of the cluster nodes, as follows:

```
# vxdbg -g diskgroupname set activation=sw
```

The following command displays whether the “shared-write” activation is set:

```
# vxdbg list galaxy_ora |grep activation
local-activation: shared-write
```

CVM log files

The `/var/VRTSvcs/log` directory contains the agent log files.

```
# cd /var/VRTSvcs/log
# ls -l *CVM* engine_A.log vxfen.log
CVMCluster_A.log      # CVM Agent log
CVMVolDg_A.log        # CVM VolDg Agent log
CVMVxconfigd_A.log    # CVM vxconfigd Agent log
engine_A.log          # VCS log
vxfen.log             # Fencing log
```

You can use the `vxconfigd.log` file to troubleshoot CVM configuration issues. The file is located at `/var/adm/ras/vxconfigd.log`

You can use the `cmdlog` file to view the list of CVM commands that have been executed. The file is located at `/var/adm/vx/cmdlog`

See the *Veritas Volume Manager Administrator's Guide* for more information.

Administering Oracle

This section provides instructions for the following Oracle administration tasks:

- [Creating a database](#)
- [Increasing swap space for Oracle](#)
- [Stopping CRS](#)
- [Determining CRS object status](#)
- [Unregistering Oracle database instances](#)
- [Configuring virtual IP addresses for CRS](#)
- [Configuring Oracle group to start and stop CRS objects](#)
- [Configuring listeners](#)
- [Starting or stopping Oracle listener](#)
- [Using `srvctl` to manage node applications](#)
- [Starting Oracle Enterprise Manager agent](#)
- [Clearing Oracle group faults](#)
- [Starting and stopping Oracle service groups](#)

If you encounter issues while administering Oracle, refer to the troubleshooting section for assistance.

See [“Troubleshooting Oracle”](#) on page 274.

Creating a database

To create a database, run the `dbca` command as follows:

```
$ export DISPLAY=display.ip
```

```
$ dbca
```

For more information, consult the Oracle documentation.

Increasing swap space for Oracle

The minimum swap space required is 4 GB. Symantec recommends two times the size of RAM for swap space. For more information on swap space, consult the Oracle documentation.

To increase swap space for Oracle

- 1 Check the amount of free swap space available.

```
# swap -l

device                maj,min              total              free
/dev/hd6              10,  2              512MB             505MB
```

- 2 Increase the swap space as required.

```
# chps -s '40' hd6
```

Stopping CRS

If you need to manually stop CRS outside of VCS control, run the following command:

```
# $CRS_HOME/bin/crsctl stop crs
Stopping resources.
Successfully stopped CRS resources
Stopping CSSD.
Shutting down CSS daemon.
Shutdown request successfully issued.
```

Determining CRS object status

To determine the status of CRS objects:

```
# $CRS_HOME/bin/crs_stat -t
```

Name	Type	Target	State	Host
ora....11.inst	application	ONLINE	OFFLINE	
ora....12.inst	application	ONLINE	OFFLINE	
ora.sample3.db	application	ONLINE	OFFLINE	
ora....EY.lsnr	application	ONLINE	ONLINE	galaxy
ora.galaxy.gsd	application	ONLINE	ONLINE	galaxy
ora.galaxy.ons	application	ONLINE	ONLINE	galaxy

```
ora.galaxy.vip application      ONLINE      ONLINE      galaxy
ora....ER.lsnr application      ONLINE      ONLINE      nebula
ora....ver.gsd application      ONLINE      ONLINE      nebula
ora....ver.ons application      ONLINE      ONLINE      nebula
ora....ver.vip application      ONLINE      ONLINE      nebula
```

The CRS objects 'gsd', 'ons', 'vip', and 'lsnr' are the nodes applications for the nodes galaxy and nebula.

Unregistering Oracle database instances

You can unregister the Oracle database instances using CRS commands or the srvctl utility.

To unregister Oracle database instances

- 1 First, review the current status of the resources registered with Oracle CRS.

```
# $CRS_HOME/bin/crs_stat -t

Name                Type                Target    State    Host
-----
ora....31.inst application      OFFLINE   OFFLINE
ora....32.inst application      OFFLINE   OFFLINE
ora.sample3.db      application      OFFLINE   OFFLINE
ora....EY.lsnr application      ONLINE    ONLINE   galaxy
ora.galaxy.gsd application      ONLINE    ONLINE   galaxy
ora.galaxy.ons application      ONLINE    ONLINE   galaxy
ora.galaxy.vip application      ONLINE    ONLINE   galaxy
ora....ER.lsnr application      ONLINE    ONLINE   nebula
ora....ver.gsd application      ONLINE    ONLINE   nebula
ora....ver.ons application      ONLINE    ONLINE   nebula
ora....ver.vip application      ONLINE    ONLINE   nebula
```

- 2 Unregister the Oracle database and instances from Oracle CRS. You can unregister using the CRS commands or the srvctl utility.

Using the CRS commands:

```
$ $CRS_HOME/bin/crs_unregister ora.sample3.sample31.inst
$ $CRS_HOME/bin/crs_unregister ora.sample3.sample32.inst
$ $CRS_HOME/bin/crs_unregister ora.sample3.db
```

Using the srvctl utility:

```
$ srvctl remove instance -d sample3 -i sample31
Remove instance sample31 from the database sample3? (y/[n]) y
$ srvctl remove instance -d sample3 -i sample32
Remove instance sample32 from the database sample3? (y/[n]) y
$ srvctl remove database -d sample3
Remove the database sample3? (y/[n]) y
```

- 3** After unregistering the instances, you can view the status of the CRS resources as follows:

```
# $CRS_HOME/bin/crs_stat -t
```

Name	Type	Target	State	Host
ora....EY.lsnr	application	ONLINE	ONLINE	galaxy
ora.galaxy.gsd	application	ONLINE	ONLINE	galaxy
ora.galaxy.ons	application	ONLINE	ONLINE	galaxy
ora.galaxy.vip	application	ONLINE	ONLINE	galaxy
ora....ER.lsnr	application	ONLINE	ONLINE	nebula
ora....ver.gsd	application	ONLINE	ONLINE	nebula
ora....ver.ons	application	ONLINE	ONLINE	nebula
ora....ver.vip	application	ONLINE	ONLINE	nebula

- 4** Register the Oracle database and instances again with Oracle CRS.

```
$ srvctl add database -d sample3 -o $ORACLE_HOME
$ srvctl add instance -d sample3 -i sample31 -n galaxy
$ srvctl add instance -d sample3 -i sample32 -n nebula
```

- 5** View the status of the CRS resources. The CRS resource status should now display as follows:

```
# $CRS_HOME/bin/crs_stat -t
```

Name	Type	Target	State	Host
ora....31.inst	application	OFFLINE	OFFLINE	
ora....32.inst	application	OFFLINE	OFFLINE	
ora.sample3.db	application	OFFLINE	OFFLINE	
ora....EY.lsnr	application	ONLINE	ONLINE	galaxy
ora.galaxy.gsd	application	ONLINE	ONLINE	galaxy
ora.galaxy.ons	application	ONLINE	ONLINE	galaxy

```
ora.galaxy.vip application ONLINE ONLINE galaxy
ora....ER.lsnr application ONLINE ONLINE nebula
ora....ver.gsd application ONLINE ONLINE nebula
ora....ver.ons application ONLINE ONLINE nebula
ora....ver.vip application ONLINE ONLINE nebula
```

- 6 Set the database to Manual mode in the Oracle CRS configuration since VCS is configured to start the database (in this example).

```
$ srvctl modify database -d sample3 -y manual
```

- 7 Review the database configuration using the srvctl utility.

```
$ srvctl config database -d sample3

galaxy sample31 /oracle/orahome
nebula sample32 /oracle/orahome
```

Configuring virtual IP addresses for CRS

To configure virtual IP addresses for CRS:

```
# export DISPLAY=display.ip

# $CRS_HOME/bin/vipca
```

where display.ip is the display IP address.

For more information, consult the Oracle documentation.

Configuring Oracle group to start and stop CRS objects

Symantec recommends that your Oracle group within VCS be configured to start and stop CRS objects.

The following sample main.cf extract shows how to configure VCS to start your CRS instance objects:

```
Oracle Ora_1 (
    Critical = 0
    Sid @galaxy = PROD11
    Sid @nebula = PROD12
    Owner = oracle
    Home = "/oracle/orahome"
    StartUpOpt = SRVCTLSTART
    ShutDownOpt = SRVCTLSTOP
```

```
OnlineTimeout = 900  
)
```

Configuring listeners

To configure listeners:

```
$ export DISPLAY=display.ip  
  
$ netca
```

For more information, consult Oracle documentation.

Starting or stopping Oracle listener

If you have issues with Oracle listener, you can stop and restart the listener as oracle user.

To start Oracle listener:

```
$ lsnrctl start
```

To stop Oracle listener:

```
$ lsnrctl stop
```

To check the status of Oracle listener:

```
$ lsnrctl status
```

For more information, consult the Oracle documentation.

Using srvctl to manage node applications

To start node applications on node galaxy:

```
$ srvctl start nodeapps -n galaxy
```

To stop node applications on node nebula:

```
$ srvctl stop nodeapps -n nebula
```

To remove node applications:

```
$ srvctl remove nodeapps -n galaxy
```

To add node applications:

```
$ srvctl add nodeapps -n galaxy \  
-A galaxy_v/255.255.248.0/en0 -o /crshome/crs
```

To modify the database settings to use manual mode:

```
$ srvctl modify database -d dbsample -y manual
```

To display instances with srvctl:

```
$ srvctl config database -d dbsample
```

```
galaxy dbsample1 /oracle/orahome  
nebula dbsample2 /oracle/orahome
```

For more details, consult the Oracle CRS documentation.

Starting Oracle Enterprise Manager agent

To use Oracle Enterprise Manager, start the agent by running the following command:

```
$ emctl start agent
```

For more information, consult Oracle OEM documentation.

Clearing Oracle group faults

If the Oracle group faults, you can clear the faults and bring the group online by running the following commands:

```
# hagr -clear grp10g -sys galaxy  
  
# hagr -clear grp10g -sys nebula  
  
# hagr -online grp10g -sys galaxy  
  
# hagr -online grp10g -sys nebula
```

Starting and stopping Oracle service groups

To start the Oracle service group "grp10g" on nodes "galaxy" and "nebula":

```
# hagr -online grp10g -sys galaxy  
  
# hagr -online grp10g -sys nebula
```

To stop the Oracle service group "grp10g" on nodes "galaxy" and "nebula":


```
# hagrps -offline grp10g -sys galaxy  
# hagrps -offline grp10g -sys nebula
```

Administering ODM

This section provides instructions for the following ODM administration tasks:

- [Verifying the ODM port](#)
- [Starting ODM](#)

If you encounter issues while administering ODM, refer to the troubleshooting section for assistance.

See “[Troubleshooting ODM](#)” on page 279.

Verifying the ODM port

It is recommended to enable ODM in SF Oracle RAC. Run the following command to verify that ODM is running:

```
# gabconfig -a | grep "Port d"
```

Starting ODM

The following procedure provides instructions for starting ODM.

To start ODM

- 1 Execute:

```
# /etc/init.d/vxgms start
```

- 2 Execute:

```
# /etc/init.d/odm start
```


Managing a database using SF Oracle RAC

- [Configuring and managing the repository database for Oracle](#)
- [Using Storage Checkpoints and Storage Rollback](#)
- [Using Database FlashSnap for backup and off-host processing](#)
- [Using Database Dynamic Storage Tiering](#)

Configuring and managing the repository database for Oracle

This chapter includes the following topics:

- [About the repository database](#)
- [Setting up the repository database using the sfua_db_config script](#)
- [Creating and configuring the repository database](#)
- [Runtime management tasks for the repository](#)
- [Adding a new system to a SF Oracle RAC configuration](#)

About the repository database

The Storage Foundation for Databases (SFDB) repository or repository database stores metadata information required by SF Oracle RAC. This information includes data about user databases, snapshot databases, storage configuration, scheduled tasks, and storage statistics.

Note: The repository database requires only occasional interaction outside of the initial installation and configuration of SF Oracle RAC.

In this release of SF Oracle RAC, the repository is stored in a relational database and is managed by a lightweight embedded relational DBMS, called VxDBMS. VxDBMS is a special OEM version of a Sybase ASA (Adaptive Server Anywhere) DBMS, which is delivered and supported by Symantec. The SFDB repository

database consists of a database file, dbed_db.db, and transaction log files, yymmddxx.log.

VxDBMS supports remote client access from any host in the network that has proper authorization and configuration.

Note: The information in this chapter is only applicable for a SF Oracle RAC configuration. For information about single instance configurations, single instance commands, or any other single instance information, please refer to the appropriate Storage Foundation documentation.

Setting up the repository database using the sfua_db_config script

After installing SF Oracle RAC and the configuration of VCS is complete, you can create and configure a repository database using the sfua_db_config script.

The sfua_db_config script detects that the system is running a SF Oracle RAC configuration and automatically configures the repository database.

The repository database configuration enables you to use the following SF Oracle RAC features:

- Storage checkpoint and storage rollback
- Database FlashSnap
- Database Dynamic Storage Tiering

Note: The SF Oracle RAC features listed above are discussed in the following chapters.

Creating and configuring the repository database

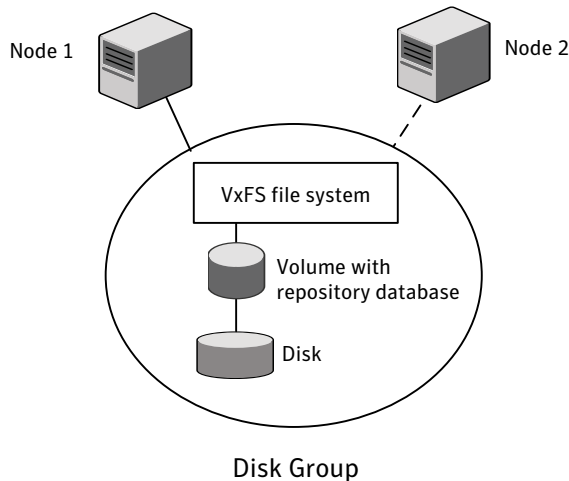
Before running the sfua_db_config script, review the following requirements:

- Make sure a disk group exists with at least one volume. The volume is to be used to store the repository database. A VxFS file system must be created on a VxVM volume in a disk group, on shared storage.
- Storage on the volume must be shared between nodes, although at any given time the disk group is only visible from a single node.

See the [Figure 4-1](#) below for a schematic of a repository database set up with two nodes accessing one disk group.

- The volume must be started and the file system must be mounted.
- Obtain a unique virtual IP address for public NIC interface.
- Obtain a device name for the public NIC interface (for example : hme0).
- Obtain a subnet mask for the public NIC interface.

Figure 4-1 Repository database schematic



To create and configure the repository database

- 1 As a root user, run the `sfua_db_config` script by entering the following command:

```
# /opt/VRTSdbcom/bin/sfua_db_config
```

Note: The above command must be run on one node. Make sure that the disk group is visible to that one node.

- 2 The following is an example of configuring a repository database for SF Oracle RAC:

```
Welcome to the SFORA configuration script.
This script creates repository for standalone and HA
configuration. Please create a Veritas File System on a Veritas
Volume and mount it, before starting configuration using this
script. This mount point will be used to store repository. The
following is required to configure $prod repository for HA
```

solution:

* A mount point of already mounted Veritas Volume on a shared storage, with Veritas File system.
* A public NIC used by each system in the cluster.
* A Virtual IP address and netmask.
Press enter to continue.

Enter Veritas filesystem mount point for SFORA repository:
/sfua_rep

Enter the following information when prompted:

Enter the NIC for system galaxy for HA Repository configuration:**hme0**

Enter the NIC for system nebula for HA Repository configuration:**hme0**

Enter the Virtual IP address for repository failover:**10.182.186.249**

Enter the netmask for public NIC interface:255.255.0.0

The following information will be used for SFORA HA configuration:

Public IP address: 10.182.186.249
Subnet mask: 255.255.0.0

Public interface: galaxy-hme0, nebula-hme0

Mount point: /sfua_rep
Volume Name for mount point: dbed_rep
Diskgroup for mount point: sfua_rep

Is this correct (y/n/q) [y]? **y**
Repository database configured successfully for HA.

3 If you are upgrading, migrate your old repository information into the new repository.

If you are installing or upgrading Veritas Storage Foundation for Oracle RAC, run the `dbed_update` command. The `dbed_update` command either creates or updates the Veritas Storage Foundation for Oracle SFDB repository. If the repository already exists, this command will refresh it.

Note: When upgrading the repository from SFRAC for Oracle version 4.1 to version 5.0, run the `dbed_update` command as described below. When upgrading SFRAC for Oracle version 5.0 to a later version, run `/sfua_db_config` which automatically upgrades the repository.

The `dbed_update` command has the following syntax:

```
dbed_update -S ORACLE_SID -H ORACLE_HOME\  
[-G <SERVICE_GROUP>] [-P <ORACLE_PFILE>]
```

To use the `dbed_update` command, the database must be up and running, and the `ORACLE_SID` and the `ORACLE_HOME` variables must be specified with and `-S` and `-H` options.

Checking the SFDB repository with `sfua_db_config`

If for any reason the VxDBMS process is not running or is having problems, use the `sfua_db_config` command to check the status of the repository database and its server.

To check the repository status, use the command with the `-o dbstatus` option.

For example:

```
# /opt/VRTSdbcom/bin/sfua_db_config -o dbstatus
```

For troubleshooting SFDB repository issues: See [“Troubleshooting the repository database”](#) on page 269.

Setting administrative permissions

To allow database administrators to administer a database using SF Oracle RAC, you must change the permission settings.

During the SF Oracle RAC installation, you are asked if you want to allow database administrator access. If you did not change permissions during the SF Oracle RAC installation, you can do so at a later time.

The default settings at installation time for the `/opt/VRTSdbdirectory` allow only the root login to access the directory.

To enable access for users other than root

- ◆ To enable access for users other than root, use the following `chown` and `chmod` commands.

```
# chmod 750 /opt/VRTSdbed
# chown oracle:dba /opt/VRTSdbed
```

Runtime management tasks for the repository

Most interactions with the SFDB repository are handled automatically by the SFDB code. Administrators only need to handle SFDB backup and restore activities and to monitor the hard disk space usage of the repository. A configuration file `/etc/vx/vxdbed/admin.properties` is created during installation to record the file system and volume used for the SFDB repository. The VxDBMS server starts automatically after a system reboot using `rc` files.

Backing up and restoring the repository with `sfua_rept_adm`

Use the `sfua_rept_adm` command to manage the backing up and restoring of the SFDB repository.

With this command, you can perform the following tasks:

- Create and configure a schedule for automatically backing up the SFDB repository, including specifying the type of backup (full or incremental), its start time and frequency, and the destination directory for the backed up files.
- Restore the repository from backup files.
- Disable or enable an existing backup schedule.
- Create and configure a free space monitoring schedule that emails an alert when the free space on the file system containing the repository falls to a specified threshold.
- Disable or enable an existing free space monitoring schedule.

Note: You must be logged in as root to use the `sfua_rept_adm` command.

Table 4-1 `sfua_rept_adm` command options

Command option	Description
-h	Displays the help for the <code>sfua_rept_adm</code> command.

Table 4-1 sfua_rept_adm command options (*continued*)

Command option	Description
-m backup_dest	This command option specifies a directory where the backup data of the repository is stored.
-o backup_sched	This command option creates a full or incremental backup schedule. The backup type is specified by the accompanying -t option. Only one full and one incremental backup schedule can be created. Creating a second schedule overwrites any existing backup schedule of that same type. After a backup schedule is created it is automatically enabled.
-o backup_enable	This command option re-enables the existing full or incremental backup schedule, as specified by the -t option.
-o backup_disable	This command option disables the existing full or incremental backup schedule, as specified by the -t option.
-o restore	This command option restores the SFDB repository from the backup files in the directory specified by the -m option.
-o space_monitor	This command option creates and configures a free space monitoring schedule that emails a warning when the free space on the file system containing the SFDB repository reaches or goes below a threshold specified by the -w option.
-o space_monitor_enable	This command option enables the free space monitoring schedule, if one exists.
-o space_monitor_disable	This command option disables the free space monitoring schedule, if one exists.
-t full incr	This command option specifies the type of backup schedule being created, enabled , or disabled by the accompanying -o option. Specify full for a full backup or incr for an incremental backup. An incremental backup copies the SFDB repository database transaction logs to the directory specified by the -m option. A full backup copies these transaction log files and the database file. Only two backup schedules can be enabled simultaneously, one incremental backup and one full backup. For example, you cannot have two different incremental backup schedules.

Table 4-1 sfua_rept_adm command options (continued)

Command option	Description
-f backup_freq	This command option specifies the frequency of the scheduled backups as h (hours), d (days), or w (weeks) from the start time specified by the -s option. By default, the backup frequency value is assumed to be hours (h).
-s start_time	This command option specifies the time to start the backup process in hh:mm:ss format.
-w warn_threshold	This command option specifies the warning threshold for the free space monitoring schedule, as a number representing the percentage of free space on the file system containing the backup files. If the percentage of free space falls to this value or lower, then a warning is emailed according to the other settings in this free space monitoring schedule.
-e notify_email	This command option specifies the email address to send the warning message when the repository file system free space falls below the threshold specified by the -w option.
-u smtp_sender	This command option specifies the SMTP sender in whose name the warning email is emailed when the repository file system free space falls below the threshold specified by the -w option.
-s smtp_server	This command option specifies the SMTP email server to use when sending the warning message when the repository file system free space falls below the threshold specified by the -w option.

To create a backup schedule for the SFDB repository

To create a backup schedule for the SFDB repository, use the sfua_rept_adm -o backup_sched command.

For example:

```
# /opt/VRTSdbcom/bin/sfua_rept_adm -o backup_sched -t full|incr -f \
  backup_freq -s start_time -m backup_dest
```

After a backup schedule is created, it is automatically enabled. You can create only one of each type of backup schedule, incremental (-t incr) and full (-t full). If you create a new backup schedule, it automatically replaces any currently-enabled

backup schedule. You can disable the current backup schedule with the `-o backup_disable` command, and re-enable it with `-o backup_enable` command.

In an HA environment, use NFS to mount the backup destination directory on all nodes.

Note: For an SFDB repository, the backup is within the same disk group, but on a separate disk and a separate volume.

To restore the SFDB repository from a backup

To restore the SFDB repository from a backup, use the `sfua_rept_adm -o restore` command.

For example:

```
# /opt/VRTSdbcom/bin/sfua_rept_adm -o restore -m backup_dest
```

This command restores the repository using the full backup and all incremental backup files from the backup directory specified by `-m backup_dest`.

To determine if a repository backup has failed

To determine if a repository backup has failed, use either of the following methods:

- Check the system console for error messages received at the time the backup was scheduled.
- Verify the existence of the proper backup files in the backup directory (specified by `-m backup_dest`). The type of repository backup you schedule determines which files should be found in this directory. If an incremental backup was scheduled and performed, then only repository transaction log files (`yymmddxx.log`) are created there. If a full backup was scheduled and performed, then both transaction log files and a repository database file (`dbed_db.db`) are created.

Monitoring free space for the SFDB repository

To guard against the SFDB repository failing by filling its file system, use the `sfua_rept_adm -o space_monitor` command to create a monitoring schedule.

[Table 4-1](#) shows all the `sfua_rept_adm` command options, including those used for creating, disabling, and re-enabling a free space monitoring schedule.

This schedule monitors the available free space on the repository file system. If the free space falls below a specified threshold (a percentage value), a warning is emailed to a specified user.

Note: You must be logged in as root to use the `sfua_rept_adm` command.

To create a free space monitoring schedule for the repository file system

To create a free space monitoring schedule for the repository file system, use the following `sfua_rept_adm -o space_monitor` command.

For example:

```
# /opt/VRTSdbcom/bin/sfua_rept_adm -o space_monitor -w warn_threshold \
-e notify_email -u smtp_sender -s smtp_server
```

After a free space monitoring schedule is created, it is automatically enabled. When the free space on the file system containing the SFDB repository falls below the threshold specified by `-w warn_threshold`, a warning is sent to the email address specified by `-e notify_email`.

Adding a new system to a SF Oracle RAC configuration

When adding a new system to an existing SF Oracle RAC configuration, you must also add the system to the existing SFDB repository so that it can share the same repository data.

To add a new system to the SFDB repository

- 1 After installing Veritas SF Oracle RAC, add the new system to the cluster.
See the Veritas Cluster Server User's Guide for additional information on this procedure.
- 2 Make sure the system is running using the following `hasys` command:

```
# hasys -state
```

- 3 Add the system to the Sfua_Base group by running the following command sequence:

```
# haconf -makerw

# hagrps -modify Sfua_Base SystemList -add new_sys sys_id

# hares -modify sfua_ip Device new_sys_NIC -sys new_sys

# haconf -dump -makero
```

- 4 Copy the /etc/vx/vxdbc/.odbc.ini file from an existing node to the new system using a remote file copy utility such as rcp, tcp, or scp.

For example, to use rcp:

```
# rcp /etc/vx/vxdbc/.odbc.ini new_sys:/etc/vx/vxdbc
```


Using Storage Checkpoints and Storage Rollback

This chapter includes the following topics:

- [About Storage Checkpoints and Storage Rollback in SF Oracle RAC](#)
- [Using Storage Checkpoints and Storage Rollback for backup and restore](#)
- [Determining space requirements for Storage Checkpoints](#)
- [Storage Checkpoint Performance](#)
- [Backing up and recovering the database using Storage Checkpoints](#)
- [Guidelines for Oracle recovery](#)
- [Using the Storage Checkpoint Command Line Interface \(CLI\)](#)
- [Command Line Interface examples](#)

About Storage Checkpoints and Storage Rollback in SF Oracle RAC

The Veritas Storage Checkpoint feature is available with SF Oracle RAC as part of the Veritas File System package and is used for the efficient backup and recovery of Oracle databases. Storage Checkpoints can also be mounted, allowing regular file system operations to be performed or secondary databases to be started. Review the following information on Storage Checkpoints and Storage Rollback and how to use these technologies through Storage Foundation for Oracle RAC.

Note: Veritas Storage Foundation for Oracle RAC only supports the SFDB features described in this guide. Additionally, the information in this chapter is only applicable for a Veritas Storage Foundation for Oracle RAC configuration. For information about single instance configurations and Storage Checkpoints and Storage Rollback, please refer to the appropriate Storage Foundation documentation.

Using Storage Checkpoints and Storage Rollback for backup and restore

Storage Checkpoints and Storage Rollback enable efficient backup and recovery of Oracle databases.

Storage Checkpoints

A Storage Checkpoint instantly creates an exact image of a database and provides a consistent image of the database from the point in time the Storage Checkpoint was created. The Storage Checkpoint image is managed and available through the Veritas Storage Foundation command line interface (CLI).

Note: A Storage Checkpoint persists after a system reboot.

Storage Rollbacks

A direct application of the Storage Checkpoint facility is Storage Rollback.

Each Storage Checkpoint is a consistent, point-in-time image of a file system, and Storage Rollback is the restore facility for these on-disk backups. Storage Rollback rolls back changed blocks contained in a Storage Checkpoint into the primary file system for faster database restoration.

Storage Checkpoints and Storage Rollback process

A Storage Checkpoint is a disk and I/O efficient snapshot technology for creating a "clone" of a currently mounted file system (the primary file system). Like a snapshot file system, a Storage Checkpoint appears as an exact image of the snapped file system at the time the Storage Checkpoint was made. However, unlike a snapshot file system that uses separate disk space, all Storage Checkpoints share the same free space pool where the primary file system resides unless a Storage Checkpoint allocation policy is assigned.

Note: A Storage Checkpoint can be mounted as read only or read-write, allowing access to the files as if it were a regular file system. A Storage Checkpoint is created using the `dbed_ckptcreate` command.

Initially, a Storage Checkpoint contains no data. The Storage Checkpoint only contains the inode list and the block map of the primary filesset. This block map points to the actual data on the primary file system. Because only the inode list and block map are required and no data is copied, creating a Storage Checkpoint takes only a few seconds and very little space.

A Storage Checkpoint initially satisfies read requests by finding the data on the primary file system, using its block map copy, and returning the data to the requesting process. When a write operation changes a data block in the primary file system, the old data is first copied to the Storage Checkpoint, and then the primary file system is updated with the new data. The Storage Checkpoint maintains the exact view of the primary file system at the time the Storage Checkpoint was taken. Subsequent writes to block *n* on the primary file system do not result in additional copies to the Storage Checkpoint because the old data only needs to be saved once. As data blocks are changed on the primary file system, the Storage Checkpoint gradually fills with the original data copied from the primary file system, and less and less of the block map in the Storage Checkpoint points back to blocks on the primary file system.

Storage Rollback restores a database, a tablespace, or datafiles on the primary file systems to the point-in-time image created during a Storage Checkpoint. Storage Rollback is accomplished by copying the "before" images from the appropriate Storage Checkpoint back to the primary file system. As with Storage Checkpoints, Storage Rollback restores at the block level, rather than at the file level. Storage Rollback is executed using the `dbed_ckptrollback` command.

Whenever you change the structure of the database (for example, by adding or deleting datafiles, converting PFILE to SPFILE, or converting SPFILE to PFILE), you must run the `dbed_update` command.

For example:

```
$ /opt/VRTS/bin/dbed_update -S $ORACLE_SID -H $ORACLE_HOME
```

Mountable Storage Checkpoints can be used for a wide range of application solutions including the following:

- Backups
- Investigations into data integrity
- Staging upgrades

- Database modifications
- Data replication solutions

If you mount a Storage Checkpoint as read-write, the command will not allow you to roll back to this Storage Checkpoint. This ensures that any Storage Checkpoint data that has been modified incorrectly cannot be a source of any database corruption. When a Storage Checkpoint is mounted as read-write, the `dbed_ckptmount` command creates a "shadow" Storage Checkpoint of and mounts this "shadow" Storage Checkpoint as read-write. This allows the database to still be rolled back to the original Storage Checkpoint.

For more information on mountable Storage Checkpoints:

See [“Mounting Storage Checkpoints using `dbed\_ckptmount`”](#) on page 145.

Determining space requirements for Storage Checkpoints

To support Block-level Incremental (BLI) Backup and storage rollback, the file systems need extra disk space to store the Storage Checkpoints. The extra space needed depends on how the Storage Checkpoints are used. Storage Checkpoints that are used to keep track of the block changes contain only file system block maps, and therefore require very little additional space (less than 1 percent of the file system size).

If the database is online while the backup is running, the additional space required by each file system for Storage Checkpoints depends on the duration of the backup and the database workload. If workload is light during the backup or the backup window is relatively short (for example, for incremental backups), for most database configurations, an additional 10 percent of the file system size will be sufficient. If the database has a busy workload while a full backup is running, the file systems may require more space.

To support Storage Checkpoints and storage rollback, VxFS needs to keep track of the original block contents when the Storage Checkpoints were created. The additional space needed is proportional to the number of blocks that have been changed since a Storage Checkpoint was taken. The number of blocks changed may not be identical to the number of changes. For example, if a data block has been changed many times, only the first change requires a new block to be allocated to store the original block content. Subsequent changes to the same block require no overhead or block allocation.

If a file system that has Storage Checkpoints runs out of space, by default VxFS removes the oldest Storage Checkpoint automatically instead of returning an ENOSPC error code (UNIX errno 28- No space left on device), which can cause the

Oracle instance to fail. Removing Storage Checkpoints automatically ensures the expected I/O semantics, but at the same time, eliminates a key recovery mechanism.

When restoring a file system that has data-full Storage Checkpoints from tape or other offline media, you need extra free space on the file system. The extra space is needed to accommodate the copy-on-write algorithm needed for preserving the consistent image of the Storage Checkpoints. The amount of free space required depends on the size of the restore and the number of Storage Checkpoints on the file system.

If you are restoring the entire file system, in most cases, you no longer need the existing Storage Checkpoint. You can simply re-make the file system using the `mkfs` command, and then restore the file system from tape or other offline media.

If you are restoring some of the files in the file system, you should first remove the data-full Storage Checkpoints that are no longer needed. If you have very limited free space on the file system, you may have to remove all data-full Storage Checkpoints in order for the restore to succeed.

To avoid unnecessary Storage Checkpoint removal, instead of using a low quota limit use the SFDB utility to set up a Monitoring Agent to monitor file system space usage. When file system space usage exceeds a preset threshold value (for example, 95 percent full), the Monitoring Agent alerts the system administrator and optionally grows the volume and the file system. Automatic notifications to the system administrator on the status of space usage and file system resizing are available through electronic mail, the `syslogd(1M)` program, or by logging messages to a simple log file.

Always reserve free disk space for growing volumes and file systems. You can also preallocate sufficient space for each file system when the file system is first created or manually grow the file system and logical volume where the file system resides.

For more information, refer to the `vxassist(1)` and `fsadm_vxfs(1)` manual pages.

Storage Checkpoint Performance

Veritas File System attempts to optimize the read and write access performance on both the Storage Checkpoint and the primary file system. Reads from a Storage Checkpoint typically perform at nearly the throughput of reads from a normal VxFS file system, allowing backups to proceed at the full speed of the VxFS file system.

Writes to the primary file system are typically affected by the Storage Checkpoints because the initial write to a data block requires a read of the old data, a write of the data to the Storage Checkpoint, and finally, the write of the new data to the

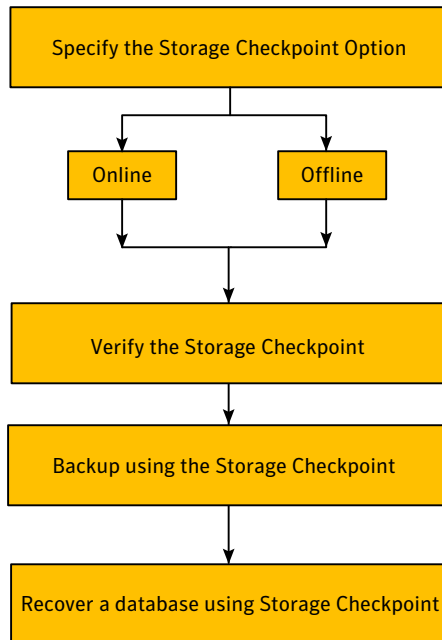
primary file system. Having multiple Storage Checkpoints on the same file system, however, will not make writes slower. Only the initial write to a block suffers this penalty, allowing operations such as writes to the intent log or inode updates to proceed at normal speed after the initial write.

The performance impact of Storage Checkpoints on a database is less when the database files are Direct I/O files. A performance degradation of less than 5 percent in throughput has been observed in a typical OLTP workload when the Storage Checkpoints only keep track of changed information. For Storage Checkpoints that are used for storage rollback, higher performance degradation (approximately 10 to 20 percent) has been observed in an OLTP workload. The degradation should be lower in most decision-support or data-warehousing environments.

Reads from the Storage Checkpoint are impacted if the primary file system is busy, because the reads on the Storage Checkpoint are slowed by all of the disk I/O associated with the primary file system. Therefore, performing database backup when the database is less active is recommended.

Backing up and recovering the database using Storage Checkpoints

[Figure 5-1](#) below describes the general process for backing up and recovering the database using Storage Checkpoints.

Figure 5-1 Backing up and recovering database using Storage Checkpoints

The following sections discuss this process:

- [Specify the Storage Checkpoint option](#)
- [Verifying a Storage Checkpoint](#)
- [Backing up using a Storage Checkpoint](#)
- [Recovering a database using a Storage Checkpoint](#)

Specify the Storage Checkpoint option

Storage Checkpoints can be created by specifying one of the following options:

- Online
- Offline

To create a Storage Checkpoint with the online option, the database should be online and you must enable ARCHIVELOG mode for the database.

Note: Refer to your Oracle documentation for information about enabling the archive log.

For the offline option, the database should be offline.

During the creation of the Storage Checkpoint, the tablespaces are placed in backup mode. Because it only takes a few seconds to take a Storage Checkpoint, the extra redo logs generated while the tablespaces are in online-backup mode are very small. You can roll back the entire database or individual tablespaces or datafiles to an online or offline Storage Checkpoint. After the rollback is complete, you may roll the database forward to restore the database if you have used an online Storage Checkpoint.

Note: To allow the easiest recovery, always keep ARCHIVELOG mode enabled, regardless of whether the database is online or offline when you create Storage Checkpoints.

Verifying a Storage Checkpoint

After creating a Storage Checkpoint and before using it to back up or restore a database, you can verify that the Storage Checkpoint is free of errors.

Usage notes See the `dbed_ckptcreate(1M)` and `dbed_ckptmount(1M)` manual pages for more information.

See [“Creating Storage Checkpoints using dbed_ckptcreate”](#) on page 140.

See [“Mounting Storage Checkpoints using dbed_ckptmount”](#) on page 145.

Storage Checkpoints can only be used to restore from logical errors (for example, a human error). Storage Checkpoints cannot be used to restore files due to a media failure, because all the data blocks are on the same physical device.

A media failure requires a database restore from a tape backup or a copy of the database files kept on a separate medium. The combination of data redundancy (disk mirroring) and Storage Checkpoints is recommended for protecting highly critical data from both physical media failure and logical errors.

To verify that a Storage Checkpoint is error-free

- 1 As oracle user, create and mount a Storage Checkpoint by issuing the following commands:

```
$ /opt/VRTS/bin/dbed_ckptcreate -S PROD1 -H $ORACLE_HOME\  
-o online
```

```
Storage Checkpoint Checkpoint_1210149848 created.
```

```
$ mkdir /tmp/ckpt_ro
```

```
$ /opt/VRTS/bin/dbed_ckptmount -S PROD1\  
-c Checkpoint_1210149848 -m /tmp/ckpt_ro -o ro
```

If the specified mount point directory does not exist, then the `dbed_ckptmount` command creates it before mounting the Storage Checkpoint, as long as the Oracle DBA user has permission to create it.

- 2 Examine the contents of the Storage Checkpoint:

```
$ ls -l /tmp/ckpt_ro/ora_data/PROD/  
total 14023952  
-rw-r----- 1 oracle oinstall 15581184 May 7 16:44 control01.ctl  
-rw-r----- 1 oracle oinstall 15581184 May 7 16:44 control02.ctl  
-rw-r----- 1 oracle oinstall 15581184 May 7 16:44 control03.ctl  
-rw-r----- 1 oracle oinstall 1536 May 4 18:48 orapwPROD  
-rw-r----- 1 oracle oinstall 52429312 May 7 16:36 redo01.log  
-rw-r----- 1 oracle oinstall 52429312 May 7 16:44 redo02.log  
-rw-r----- 1 oracle oinstall 52429312 May 7 15:43 redo03.log  
-rw-r----- 1 oracle oinstall 52429312 May 7 15:42 redo04.log  
-rw-r----- 1 oracle oinstall 4096 May 7 15:42 spfilePROD.ora  
-rw-r----- 1 oracle oinstall 293609472 May 7 16:44 sysaux01.dbf  
-rw-r----- 1 oracle oinstall 5693775872 May 7 16:44 system01.dbf  
-rw-r----- 1 oracle oinstall 791683072 May 6 22:06 temp01.dbf  
-rw-r----- 1 oracle oinstall 41951232 May 7 16:44 undotbs01.dbf  
-rw-r----- 1 oracle oinstall 97525760 May 7 16:44 undotbs02.dbf  
-rw-r----- 1 oracle oinstall 5251072 May 7 16:44 users01.dbf
```

3 Run the dbv tool against the Quick I/O files. For example:

```
$ $ORACLE_HOME/bin/dbv file=/tmp/ckpt_ro/ora_data/PROD/undotbs01.dbf
```

```
DBVERIFY: Release 10.2.0.3.0 - Production on Wed May 7 16:52:15 2008
```

```
Copyright (c) 1982, 2005, Oracle. All rights reserved.
```

```
DBVERIFY - Verification starting : FILE  
= /tmp/ckpt_ro/ora_data/PROD/undotbs01.dbf
```

```
DBVERIFY - Verification complete
```

```
Total Pages Examined          : 5120  
Total Pages Processed (Data)   : 0  
Total Pages Failing (Data)    : 0  
Total Pages Processed (Index) : 0  
Total Pages Failing (Index)   : 0  
Total Pages Processed (Other) : 4841  
Total Pages Processed (Seg)    : 0  
Total Pages Failing (Seg)     : 0  
Total Pages Empty             : 279  
Total Pages Marked Corrupt     : 0  
Total Pages Influx            : 0  
Highest block SCN             : 2516312 (0.2516312)
```

For example:

```
$ $ORACLE_HOME/bin/dbv file=/tmp/ckpt_ro/ora_data/PROD/undotbs02.dbf
```

```
DBVERIFY: Release 10.2.0.3.0 - Production on Wed May 7 16:52:20 2008
```

```
Copyright (c) 1982, 2005, Oracle. All rights reserved.
```

```
DBVERIFY - Verification starting : FILE  
= /tmp/ckpt_ro/ora_data/PROD/undotbs02.dbf
```

```
DBVERIFY - Verification complete
```

```
Total Pages Examined          : 11904  
Total Pages Processed (Data)   : 0
```

```
Total Pages Failing    (Data) : 0
Total Pages Processed  (Index): 0
Total Pages Failing    (Index): 0
Total Pages Processed  (Other): 11662
Total Pages Processed  (Seg) : 0
Total Pages Failing    (Seg) : 0
Total Pages Empty      : 242
Total Pages Marked Corrupt : 0
Total Pages Influx     : 0
Highest block SCN      : 2516289 (0.2516289)
```

Backing up using a Storage Checkpoint

You can back up a database by creating a Storage Checkpoint using the `dbed_ckptcreate` command, mount the Storage Checkpoint as read only using the `dbed_ckptmount` command, and then back it up using tools such as `tar` or `cpio`.

Usage notes See the `dbed_ckptcreate(1M)`, `dbed_ckptmount(1M)`, `tar(1)`, and `cpio(1)` manual pages for more information.

See [“Creating Storage Checkpoints using `dbed\_ckptcreate`”](#) on page 140.

See [“Mounting Storage Checkpoints using `dbed\_ckptmount`”](#) on page 145.

In the example procedure, all the database datafiles reside on one VxFS file system named `/db01`.

To back up a frozen database image using the command line

- 1 As an Oracle user, create a Storage Checkpoint using the `dbed_ckptcreate` command:

```
$ /opt/VRTS/bin/dbed_ckptcreate -S PROD -H /oracle/product \  
-o online
```

```
Creating online Storage Checkpoint of database PROD.
```

```
Storage Checkpoint Checkpoint_903937870 created.
```

- 2 Mount the Storage Checkpoint using the `dbed_ckptmount` command:

```
$ /opt/VRTS/bin/dbed_ckptmount -S PROD -c Checkpoint_903937870 \  
-m /tmp/ckpt_ro
```

If the specified mount point directory does not exist, then the `dbed_ckptmount` command creates it before mounting the Storage Checkpoint, as long as the Oracle DBA user has permission to create it.

- 3 Use tar to back up the Storage Checkpoint:

```
$ cd /tmp/ckpt_ro  
$ ls  
db01  
$ tar cvf /tmp/PROD_db01_903937870.tar ./db01
```

Recovering a database using a Storage Checkpoint

Since Storage Checkpoints record the "before" images of blocks that have changed, you can use them to do a file-system-based storage rollback to the exact time when the Storage Checkpoint was taken. You can consider Storage Checkpoints as backups that are online, and you can use them to roll back an entire database, a tablespace, or a single database file. Rolling back to or restoring from any Storage Checkpoint is generally very fast because only the changed data blocks need to be restored.

Some database changes made after a Storage Checkpoint was taken may make it impossible to perform an incomplete recovery of the databases after Storage Rollback of an online or offline Storage Checkpoint using the current control files. For example, you cannot perform an incomplete recovery of the database to the point right before the control files have recorded the addition or removal of datafiles.

To provide recovery options, a backup copy of the control file for the database is saved under the `/etc/vx/SFDB/$ORACLE_SID/checkpoint_dir/CKPT_NAME` directory immediately after a Storage Checkpoint is created. You can use this file to assist with database recovery, if necessary. If possible, both ASCII and binary versions of the control file will be left under the `/etc/vx/SFDB/$ORACLE_SID/checkpoint_dir/CKPT_NAME` directory. The binary version will be compressed to conserve space.

Warning: Use extreme caution when recovering your database using alternate control files.

Suppose a user deletes a table by mistake right after 4:00 p.m., and you want to recover the database to a state just before the mistake. You created a Storage Checkpoint (Checkpoint_903937870) while the database was running at 11:00 a.m., and you have ARCHIVELOG mode enabled.

To recover the database using a Storage Checkpoint

- 1 As root, freeze the VCS service group for the database.

```
# hagrps -freeze Service_Group
```

- 2 Ensure that the affected datafiles, tablespaces, or database are offline.
- 3 Use storage rollback to roll back any datafiles in the database that contained the table data from the Storage Checkpoint you created at 11:00 a.m.

For example:

```
$ /opt/VRTS/bin/dbed_ckptrollback -S $ORACLE_SID -H\
$ORACLE_HOME -c Checkpoint_1206725007
```

For other examples of this command (for a database, tablespace, or datafile):

See [“Performing Storage Rollback using dbed_ckptrollback”](#) on page 150.

- 4 Start up the database instance if it is down.
- 5 Unfreeze the service group.

```
# hagrps -unfreeze Service_Group
```

- 6 Re-apply archive logs to the point before the table was deleted to recover the database to 4:00 p.m. Use one of the following commands:

```
SQL> recover database until cancel
```

```
SQL> recover database until change
```

```
SQL> recover database until time
```

- 7 Open the database with the following command:

```
SQL> alter database open resetlogs
```

- 8 Delete the Storage Checkpoint you created at 11:00 a.m. and any other Storage Checkpoints created before that time.
- 9 Create a new Storage Checkpoint.

Guidelines for Oracle recovery

For an optimal Oracle recovery, the following steps should be taken:

- [Back up all control files before Storage Rollback](#)
- [Ensure that the control files are not rolled back](#)
- [Ensure that all archived redo logs are available](#)
- [Media recovery procedures](#)

Back up all control files before Storage Rollback

This guideline is recommended in case the subsequent Oracle recovery is not successful.

Oracle recommends that you keep at least two copies of the control files for each Oracle database and that you store the copies on different disks. Control files should also be backed up before and after making structural changes to databases.

Note: The `dbed_ckptcreate` command automatically saves control file and log information when you create a Storage Checkpoint.

See [“Creating Storage Checkpoints using dbed_ckptcreate”](#) on page 140.

Ensure that the control files are not rolled back

A control file is a small binary file that describes the structure of the database and must be available to mount, open, and maintain the database. The control file stores all necessary database file information, log file information, the name of the database, the timestamp of database creation, and synchronization information, such as the Storage Checkpoint and log-sequence information needed for recovery.

Rolling back the control file will result in an inconsistency between the physical database structure and the control file.

If your intention is to roll back the database to recover from structural changes that you do not want to maintain, you may want to use the backup control file that was created by the `dbed_ckptcreate` command. The backup control file is located in the directory

`$VXDBA_DBPATH/$ORACLE_SID/checkpoint_dir/CKPT_NAME`.

Ensure that all archived redo logs are available

A database backup with online and archived logs is required for a complete database recovery.

Query `V$ARCHIVED_LOG` to list all the archived log information and `V$ARCHIVE_DEST` to list the location of archive destinations.

Note: Refer to your Oracle documentation for information about querying archived information.

For SF Oracle RAC, the archive log destination must be on a Veritas cluster file system.

To restore the necessary archived redo log files, you can query `V$LOG_HISTORY` to list all the archived redo log history or query `V$RECOVERY_LOG` to list only the archived redo logs needed for recovery. The required archived redo log files can be restored to the destination specified in the `LOG_ARCHIVE_DEST` parameter or to an alternate location. If the archived redo logs were restored to an alternate location, use the `ALTER DATABASE RECOVER ... FROM` statement during media recovery.

After Storage Rollback, perform Oracle recovery, applying some or all of the archived redo logs.

Note: After rolling back the database (including control files and redo logs) to a Storage Checkpoint, you need to recover the Oracle database instance. Rolling the database forward is not supported; that is, you cannot apply archived redo logs.

Media recovery procedures

The following are the procedures for performing either a complete or incomplete media recovery.

Media recovery procedures

- To perform a complete media recovery:

```
SQL> SET AUTORECOVERY ON;
```

```
SQL> RECOVER DATABASE;
```

- To perform an incomplete media recovery, use one of the following:

```
SQL> RECOVER DATABASE UNTIL CANCEL;
```

or

```
SQL> RECOVER DATABASE UNTIL TIME 'yyyy-mm-dd:hh:mm:ss' ;
```

(You can confirm the time of error by checking the ../bdump/alert*.log file.)

or

```
SQL> RECOVER DATABASE UNTIL TIME 'yyyy-mm-dd:hh:mm:ss' \
using backup controlfile;
```

or

```
SQL> RECOVER DATABASE UNTIL CHANGE scn;
```

- To open the database after an incomplete media recovery, use the following:

```
SQL> ALTER DATABASE OPEN RESETLOGS;
```

RESETLOGS resets the log sequence. The RESETLOGS option is required after an incomplete media recovery. After opening the database with the RESETLOGS

option, remove the Storage Checkpoint you just rolled back to as well as any Storage Checkpoints that were taken before that one. These earlier Storage Checkpoints can no longer be used for storage rollback. After removing these Storage Checkpoints, be sure to create a new Storage Checkpoint.

Warning: Attempting to roll back to the same Storage Checkpoint more than once can result in data corruption. After rolling back, be sure to delete the Storage Checkpoint that you rolled back to and then create a new one.

See your Oracle documentation for complete information on recovery.

Using the Storage Checkpoint Command Line Interface (CLI)

Veritas Storage Foundation for Oracle RAC provides a command line interface to many key operations. The command line interface lets you incorporate command operations into scripts and other administrative processes.

Note: The SF Oracle RAC command line interface depends on certain tablespace and container information that is collected and stored in a repository. Some CLI commands update the repository by default. It is also important to regularly ensure that the repository is up-to-date by using the `dbed_update` command.

Note: For SF Oracle RAC database, when you issue the commands, replace `$ORACLE_SID` with `$ORACLE_SID=instance_name` and provide the instance name on which the instance is running.

Commands Overview

SF Oracle RAC commands supported in the command line interface are located in the `/opt/VRTS/bin` directory.

The online manual pages for these commands are located in the `/opt/VRTS/man` directory.

[Table 5-1](#) summarizes the commands available to you from the command line.

Table 5-1 Veritas Storage Foundation for Oracle RAC Checkpoint Commands

Command	Description
<code>dbed_update</code>	Command that creates or updates the Veritas Storage Foundation for Oracle SFDB repository. See “Creating or updating the repository using <code>dbed_update</code>” on page 140.
<code>dbed_checkconfig</code>	Command that checks the configuration of an Oracle database in a Veritas Storage Foundation for Oracle RAC environment. Note: This command only checks the local instance for SF Oracle RAC.
<code>dbed_saveconfig</code>	Command that saves the configuration of an Oracle database in a Veritas Storage Foundation for Oracle RAC environment. Note: This command only saves the configuration for the local instance for SF Oracle RAC.
<code>dbed_ckptcreate</code>	Command that creates a Storage Checkpoint for an Oracle database. See “Creating Storage Checkpoints using <code>dbed_ckptcreate</code>” on page 140.
<code>dbed_ckptdisplay</code>	Command that displays the Storage Checkpoints associated with an Oracle instance. See “Displaying Storage Checkpoints using <code>dbed_ckptdisplay</code>” on page 142.
<code>dbed_ckptmount</code>	Command that mounts a Storage Checkpoint for an Oracle instance. See “Mounting Storage Checkpoints using <code>dbed_ckptmount</code>” on page 145.
<code>dbed_ckptpolicy</code>	Command that creates and administers Storage Checkpoint allocation policies for a Multi-Volume File System (MVS). You can display, create, update, assign, and remove Storage Checkpoint allocation policies using this command. See “Creating and working with Storage Checkpoint allocation policies using <code>dbed_ckptpolicy</code>” on page 147.

Table 5-1 Veritas Storage Foundation for Oracle RAC Checkpoint Commands
(continued)

Command	Description
<code>dbed_ckptquota</code>	Command that administers quotas for Storage Checkpoints. Note: This command only administers quotas for Storage Checkpoints for the local instance for SF Oracle RAC.
<code>dbed_ckptumount</code>	Command that unmounts a Storage Checkpoint for an Oracle instance. See “Unmounting Storage Checkpoints using <code>dbed_ckptumount</code>” on page 146.
<code>dbed_ckptrollback</code>	Command that rolls back an Oracle instance to a Storage Checkpoint point-in-time image. See “Performing Storage Rollback using <code>dbed_ckptrollback</code>” on page 150.
<code>dbed_ckptremove</code>	Command that removes a Storage Checkpoint for an Oracle instance. See “Removing Storage Checkpoints using <code>dbed_ckptremove</code>” on page 152.
<code>dbed_clonedb</code>	Command that creates a copy of an Oracle database by cloning all existing database files and recreating the control file accordingly. This cloned database can only be started on the same host as the existing database as long as it uses a different SID. See “Cloning the Oracle instance using <code>dbed_clonedb</code>” on page 152.

Command Line Interface examples

This section displays examples of SF Oracle RAC commands that are used to perform administrative operations for Storage Checkpoints and Storage Rollbacks.

Note: For detailed information about these commands, their command syntax, and available options, see the individual manual pages.

Prerequisites

Review the prerequisites and usage notes listed below for each command before using that command.

Creating or updating the repository using `dbed_update`

You can use the `dbed_update` command to create or update the repository.

Note: Any time you change the structure of the database (for example, by adding or deleting datafiles, converting PFILE to SPFILE, or converting SPFILE to PFILE), you must run the `dbed_update` command.

Before creating or updating the repository, the following conditions must be met:

Prerequisites	■ You must be logged on as the database administrator (typically, the user ID <code>oracle</code>).
Usage notes	■ The <code>dbed_update</code> command creates a repository in the <code>/etc/vx/vxdbed/\$ORACLE_SID</code> directory where information used by SF Oracle RAC is kept. If the repository already exists, the command will refresh the information. ■ The database must be up and running, and the <code>ORACLE_SID</code> and the <code>ORACLE_HOME</code> variable arguments must be specified with the <code>-S</code> and <code>-H</code> options, respectively. ■ See the <code>dbed_update(1M)</code> manual page for more information.

To update the repository

- ◆ Use the `dbed_update` command as follows:

```
$ /opt/VRTS/bin/dbed_update -S PROD -H /oracle/product/ORA_HOME
```

To view the status of the repository

- ◆ Use the `dbed_update` command as follows:

```
$ /opt/VRTS/bin/dbed_update -S PROD -H /oracle/product/ORA_HOME
Repository is up to date.
```

Creating Storage Checkpoints using `dbed_ckptcreate`

You can use the `dbed_ckptcreate` command to create a Storage Checkpoint for an Oracle database from the command line.

Storage Checkpoints can be either online or offline. By default, Storage Checkpoints are offline. If online is specified, the database is put into hot-backup mode when the Storage Checkpoint is created. If offline is specified, the database is expected to be down.

Before creating a Storage Checkpoint, the following conditions must be met:

Prerequisites	■ You must be logged on as the database administrator (typically, the user ID oracle). ■ For best recoverability, always keep ARCHIVELOG mode enabled when you create Storage Checkpoints.
Usage notes	■ <code>dbed_ckptcreate</code> stores Storage Checkpoint information under the following directory: <code>/etc/vx/vxdbed/\$ORACLE_SID/checkpoint_dir</code> ■ See the <code>dbed_ckptcreate(1M)</code> manual page for more information.

To create Storage Checkpoints while the database is online

- ◆ Use the `dbed_ckptcreate` command as follows:

```
$/opt/VRTS/bin/dbed_ckptcreate -S PROD \  
-H /oracle/product/ORA_HOME -o online  
  
Storage Checkpoint Checkpoint_971672043 created.
```

To create Storage Checkpoints without updating the repository while the database is online

- ◆ Use the `dbed_ckptcreate` command as follows:

```
$/opt/VRTS/bin/dbed_ckptcreate -S PROD \  
-H /oracle/product/ORA_HOME -o online -n  
  
Storage Checkpoint Checkpoint_971672046 created.
```

To create Storage Checkpoints while the database is offline

- ◆ Use the `dbed_ckptcreate` command as follows:

```
$/opt/VRTS/bin/dbed_ckptcreate -S PROD \  
-H /oracle/product/ORA_HOME -o offline  
  
Storage Checkpoint Checkpoint_971672049 created.
```

Note: The default option is offline.

To assign a Storage Checkpoint allocation policy to a Storage Checkpoint

- ◆ Use the `dbed_ckptcreate` command as follows:

```
$/opt/VRTS/bin/dbed_ckptcreate -S PROD \  
-H /oracle/product/ORA_HOME -o online -p ckpt_data,ckpt_metadata  
  
Storage Checkpoint Checkpoint_971672055 created.
```

Displaying Storage Checkpoints using `dbed_ckptdisplay`

You can use the `dbed_ckptdisplay` command to display the Storage Checkpoints associated with an Oracle database from the command line.

You can also use it to display fileset quota values.

Before displaying Storage Checkpoints, the following conditions must be met:

Prerequisites	■ You must be logged on as the database administrator.
Usage Notes	■ In addition to displaying the Storage Checkpoints created by SF Oracle RAC, <code>dbed_ckptdisplay</code> also displays other Storage Checkpoints (for example, Storage Checkpoints created by the Capacity Planning Utility and NetBackup). ■ The Status field identifies if the Storage Checkpoint is partial (P), complete (C), invalid (I), mounted (M), read only (R), writable (W), or of type online (ON), offline (OF), instant (IN), or unknown (UN). Note that instant (IN) Storage Checkpoints are not supported in an SF Oracle RAC environment. ■ Database FlashSnap commands are integrated with Storage Checkpoint functionality. It is possible to display and mount Storage Checkpoints carried over with snapshot volumes to a secondary host. However limitations apply. See “Mounting the snapshot volumes and backing up” on page 199. ■ See the <code>dbed_ckptdisplay(1M)</code> manual page for more information.

To display Storage Checkpoints created by Veritas Storage Foundation for Oracle RAC

- ◆ Use the `dbed_ckptdisplay` command as follows to display information for Storage Checkpoints created by SF Oracle RAC:

```
$ /opt/VRTS/bin/dbed_ckptdisplay -S PROD \
-H /oracle/product/ORA_HOME

Checkpoint_975877759      Sun Apr 3 12:50:59 2005      P+R+IN
Checkpoint_974428422_wr001Thu May 16 17:28:42 2005      C+R+ON
Checkpoint_974428423      Thu May 16 17:28:42 2004      P+R+ON
```

To display other Storage Checkpoints

- ◆ Use the `dbed_ckptdisplay` command as follows: :

```
$ /opt/VRTS/bin/dbed_ckptdisplay -S PROD \
-H /oracle/product/ORA_HOME -o other

NetBackup_incr_PROD_955187480      NBU      /db01
NetBackup_full_PROD_95518725      54      NBU      /db01
```

To display other Storage Checkpoints without updating the repository

- ◆ Use the `dbed_ckptdisplay` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptdisplay -S PROD \
-H /oracle/product/ORA_HOME -o other -n

NetBackup_incr_PROD_955187480      NBU      /db01
NetBackup_full_PROD_95518725      54      NBU      /db01
```

To display all Storage Checkpoints

- ◆ Use the `dbed_ckptdisplay` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptdisplay -S PROD \
-H /oracle/product/ORA_HOME -o all

Checkpoint_971672042      Sun May 15 13:55:53 2005      C+R+IN
Checkpoint_903937870      Fri May 13 22:51:10 2005      C+R+ON
Checkpoint_901426272      Wed May 11 16:17:52 2005      P+R+ON
NetBackup_incr_PROD_955133480      NBU      /db01
NetBackup_full_PROD_9551329      52      NBU      /db01
```

To display all Storage Checkpoints without updating the repository

- ◆ Use the `dbed_ckptdisplay` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptdisplay -S PROD \  
-H /oracle/product/ORA_HOME -o all -n  
Checkpoint_971672042    Sun May 15 13:55:53 2005      C+R+IN  
Checkpoint_903937870    Fri May 13 22:51:10 2005      C+R+ON  
Checkpoint_901426272    Wed May 11 16:17:52 2005      P+R+ON  
NetBackup_incr_PROD_955133480      NBU      /db01  
NetBackup_full_PROD_9551329    52      NBU      /db01
```

To display fileset quota values

- ◆ Use the `dbed_ckptdisplay` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptdisplay -S PROD -c \  
Checkpoint_903937870 -Q  
Checkpoint_903937870    Wed Mar 19 9:12:20 2005      C+R+ON  
Filesystem              HardLim    SoftLim    CurrentUse  
/oradata1/indx1_1      100000     50000      2028  
/oradata1/user1_1      100000     50000      2028  
/oradata1/temp          150000     80000      2142  
/oradata1/system1      150000     70000      3092
```

Scheduling Storage Checkpoints using `dbed_ckptcreate` and `cron`

You can use the `dbed_ckptcreate` command to schedule Storage Checkpoint creation in a `cron` job or other administrative script.

Before scheduling Storage Checkpoints, the following conditions must be met:

- Prerequisites
- You must be logged on as the database administrator (typically, the user ID `oracle`).

Usage notes

- Create a new crontab file or edit an existing crontab file to include a Storage Checkpoint creation entry with the following space-delimited fields:
minute hour day_of_month month_of_year day_of_week
`/opt/VRTS/bin/dbed_ckptcreate`
 where:
minute - numeric values from 0-59 or *
hour - numeric values from 0-23 or *
day_of_month - numeric values from 1-31 or *
month_of_year - numeric values from 1-12 or *
day_of_week - numeric values from 0-6, with 0=Sunday or *
 Each of these variables can either be an asterisk (meaning all legal values) or a list of elements separated by commas. An element is either a number or two numbers separated by a hyphen (meaning an inclusive range).
- See the `dbed_ckptcreate(1M)`, `cron(1M)`, and `crontab(1)` manual pages for more information.

Scheduling Storage Checkpoint creation in a cron job

Depending on when you want to schedule Storage Checkpoint creation, make entries to the crontab file.

- To create a Storage Checkpoint at 1:00 a.m. every Sunday while the database is offline, include the following entry in your crontab file:

```
0 1 * * 0 /opt/VRTS/bin/dbed_ckptcreate -S PROD \  
-H /oracle/product/ORA_HOME -o offline
```

Note: This is a crontab example for user oracle.

Mounting Storage Checkpoints using dbed_ckptmount

You can use the `dbed_ckptmount` command to mount a Storage Checkpoint for the database from the command line.

Before mounting Storage Checkpoints, the following conditions must be met:

Prerequisites

- You must be logged on as the database administrator.

Usage notes

- The `dbed_ckptmount` command is used to mount a Storage Checkpoint into the file system namespace. Mounted Storage Checkpoints appear as any other file system on the machine and can be accessed using all normal file system based commands.
- Storage Checkpoints can be mounted as read only or read-write. By default, Storage Checkpoints are mounted as read only.
- If the `rw` (read-write) option is used, `_wrxxx`, where `xxx` is an integer, will be appended to the Storage Checkpoint name.
- If the specified mount point directory does not exist, then `dbed_ckptmount` creates it before mounting the Storage Checkpoint, as long as the Oracle database owner has permission to create it.
- Database FlashSnap commands are integrated with Storage Checkpoint functionality. It is possible to display and mount Storage Checkpoints carried over with snapshot volumes to a secondary host. However limitations apply.
See [“Mounting the snapshot volumes and backing up”](#) on page 199.
- See the `dbed_ckptmount(1M)` manual page for more information.

To mount Storage Checkpoints with the read/write option

- ◆ Use the `dbed_ckptmount` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptmount -S PROD -c Checkpoint_971672042 \
-m /tmp/ckpt_rw -o rw
Creating Storage Checkpoint on /tmp/ckpt_rw/share/oradata with
name Checkpoint_971672042_wr001
```

To mount Storage Checkpoints with the read only option

- ◆ Use the `dbed_ckptmount` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptmount -S PROD -c Checkpoint_971672042 \
-m /tmp/ckpt_ro -o ro
```

Unmounting Storage Checkpoints using `dbed_ckptumount`

You can use the `dbed_ckptumount` command to unmount a Storage Checkpoint for an Oracle database from the command line.

Before unmounting Storage Checkpoints, the following conditions must be met:

Prerequisites

- You must be logged on as the database administrator.

- Usage notes
- The `dbed_ckptumount` command is used to unmount a mounted Storage Checkpoint from the file system namespace. Mounted Storage Checkpoints appear as any other file system on the machine and can be accessed using all normal file system based commands. When mounted Storage Checkpoints are not required, they can be unmounted.
 - See the `dbed_ckptumount(1M)` manual page for more information.

To unmount Storage Checkpoints

- ◆ Use the `dbed_ckptumount` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptumount -S PROD \  
-c Checkpoint_971672042_wr001
```

Creating and working with Storage Checkpoint allocation policies using `dbed_ckptpolicy`

You can use the `dbed_ckptpolicy` command to create and administer Storage Checkpoint allocation policies for Multi-Volume File Systems (MVS). Storage Checkpoint allocation policies specify a list of volumes and the order in which to allocate data to them.

Before creating or working with Storage Checkpoint allocation policies, the following conditions must be met:

- Prerequisites
- You must be logged on as the database administrator (typically, the user ID `oracle`).

- Usage notes
- The `dbed_ckptpolicy` command can be used only on file systems using disk layout Version 6.
 - The VxVM volume set and VxFS Multi-Volume File System features must be enabled to use Storage Checkpoint allocation policies.
 - If you want to set a Storage Checkpoint allocation policy for a particular file system in the database, the VxFS Multi-Volume File System feature must be enabled on that file system.
 - The status of a Storage Checkpoint allocation policy is either partial or complete. A partial policy is one that does not exist on all file systems used by the database. A complete policy is one that exists on all file systems.
 - After an allocation policy is assigned to a Storage Checkpoint, the allocation mechanism attempts to satisfy requests from each device in the order specified in the allocation policy. If the request cannot be satisfied from any of the devices in the allocation policy, the request will fail, even if other devices that have space exist in the file system. Only devices listed in the policy can be allocated.
 - See the `dbed_ckptpolicy(1M)` manual page for more information.

To create a Storage Checkpoint allocation policy

- ◆ Use the `dbed_ckptpolicy` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptpolicy -S ORACLE_SID \  
-o create -p ckpt_policy
```

Output similar to the following is displayed:

```
File System: /mvsfs/v2 (MVS volumes: mvsv4,mvsv5)  
Assigned Data Policy: NONE (MVS Volumes: N/A)  
Assigned Meta Data Policy: NONE (MVS Volumes: N/A)  
Please enter the volume name(s), sperated by space, for the  
policy ckpt_policy [skip,quit]: mvsv4
```

```
File System: /mvsfs/v1 (MVS volumes: mvsv1,mvsv2,mvsv3)  
Assigned Data Policy: NONE (MVS Volumes: N/A)  
Assigned Meta Data Policy: NONE (MVS Volumes: N/A)  
Please enter the volume name(s), separated by space, for the  
policy ckpt_policy [skip,quit]: mvsv2
```

```
The following information will be used to create policy  
ckpt_sample  
ckpt_sample          /mvsfs/v2          mvsv4  
ckpt_sample          /mvsfs/v1          mvsv2
```

This example assumes the following:

- Two MVS file systems /mvsfs/v1 and /mvsfs/v2 are used for datafiles.
- File system /mvsfs/v1 is created on volume set mvsvset1.
- File system /mvsfs/v2 is created on volume set mvsvset2.
- Volume set mvsvset1 contains volumes mvsv1, mvsv2, and mvsv3.
- Volume set mvsvset2 contains volumes mvsv4 and mvsv5.

To display Storage Checkpoint allocation policy within the database

- ◆ Use the `dbed_ckptpolicy` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptpolicy -S ORACLE_SID \
-n -o display [-c storage_ckpt | -p ckpt_policy]
```

If `-p ckpt_policy` and `-c storage_ckpt` options are not specified, output similar to the following is displayed:

Policy Name	File System Coverage
ckpt	Complete
ckpt_data	Complete
ckpt_metadata	Complete
new_ckpt	Partial
ckpt_sample	Complete

If `-p ckpt_policy` option is specified, output similar to the following is displayed:

Policy Name	File System	MVS volumes
ckpt_sample	/mvsfs/v2	mvsv4
ckpt_sample	/mvsfs/v1	mvsv2

If the `-c storage_ckpt` option is specified, output similar to the following is displayed:

Storage Checkpoint File System	Data Policy	Meta Data Policy
Checkpoint_1095125037/mvsfs/v2	ckpt_data	ckpt_metadata
Checkpoint_1095125037/mvsfs/v1	ckpt_data	ckpt_metadata

To update a Storage Checkpoint allocation policy

- ◆ Use the `dbed_ckptpolicy` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptpolicy -S ORACLE_SID \  
-n -o update -p ckpt_policy
```

Output similar to the following is displayed:

```
File System: /mvsfs/v2 (MVS volumes: mvsv4,mvsv5)  
Policy: ckpt_sample (MVS volumes: mvsv4)  
Please enter the volume name(s), separated by space, for the  
policy ckpt_sample [skip,quit]: mvsv5
```

```
File System: /mvsfs/v1 (MVS volumes: mvsv1,mvsv2,mvsv3)  
Policy: ckpt_sample (MVS volumes: mvsv2)  
Please enter the volume name(s), separated by space, for the  
policy ckpt_sample [skip,quit]: mvsv2,mvsv3
```

```
The following information will be used to create policy  
ckpt_sample  
ckpt_sample           /mvsfs/v2             mvsv5  
ckpt_sample           /mvsfs/v1  
mvsv2,mvsv3
```

To assign a Storage Checkpoint allocation policy

- ◆ Use the `dbed_ckptpolicy` command as follows to assign an allocation policy to a specified Storage Checkpoint:

```
$ /opt/VRTS/bin/dbed_ckptpolicy -S ORACLE_SID \  
-n -o assign -c ckpt_name -p ckpt_policy[,ckpt_metadata_policy]
```

To remove a Storage Checkpoint allocation policy

- ◆ Use the `dbed_ckptpolicy` command as follows to remove an allocation policy from a specified Storage Checkpoint:

```
$ /opt/VRTS/bin/dbed_ckptpolicy -S ORACLE_SID \  
-n -o remove -p ckpt_policy
```

Performing Storage Rollback using `dbed_ckptrollback`

You can use the `dbed_ckptrollback` command to rollback an Oracle database to a Storage Checkpoint.

Before performing a Storage Rollback, the following conditions must be met:

Prerequisites	■ You must be logged on as the database administrator.
Usage notes	■ The <code>dbed_ckptrollback</code> command rolls an Oracle database back to a specified Storage Checkpoint. You can perform a Storage Rollback for the entire database, a specific tablespace, or list of datafiles. Database rollback for the entire database requires that the database be inactive before Storage Rollback commences. The <code>dbed_ckptrollback</code> command will not commence if the Oracle database is active. However, to perform a Storage Rollback of a tablespace or datafile, only the tablespace or datafile to be rolled back must be offline (not the entire database). ■ You must run the <code>dbed_update</code> command after upgrading to Storage Foundation 5.0 for Oracle RAC from a previous release. This will allow you to roll back to a Storage Checkpoint that was created with an earlier version of this product. ■ See the <code>dbed_ckptrollback(1M)</code> manual page for more information.

To roll back an Oracle database to a Storage Checkpoint

- ◆ Use the `dbed_ckptrollback` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptrollback -S PROD \  
-H /oracle/product/ORA_HOME -c Checkpoint_903937870
```

To rollback a tablespace to a Storage Checkpoint

- ◆ Use the `dbed_ckptrollback` command with the `-T` option as follows:

```
$ /opt/VRTS/bin/dbed_ckptrollback -S PROD \  
-H /oracle/product/ORA_HOME -T DATA01 -c Checkpoint_903937870
```

If the Oracle database is running, you must take the tablespace offline before running this command. If the tablespace is online, the command will fail.

To rollback datafiles to a Storage Checkpoint

- ◆ Use the `dbed_ckptrollback` command with the `-F` option as follows:

```
$ /opt/VRTS/bin/dbed_ckptrollback -S PROD \  
-H /oracle/product/ORA_HOME \  
-F /share/oradata1/data01.dbf,/share/oradata2/index01.dbf \  
-c Checkpoint_903937870
```

If the Oracle database is running, you must take the datafile offline before running this command. If the datafile is online, the command will fail.

Removing Storage Checkpoints using `dbed_ckptremove`

You can use the `dbed_ckptremove` command to remove a Storage Checkpoint for an Oracle database at the command line.

Before removing Storage Checkpoints, the following conditions must be met:

Prerequisites	■ You must be logged on as the database administrator.
Usage notes	■ The <code>dbed_ckptremove</code> command is used to remove a Storage Checkpoint from the file system, or file systems, it is associated with. The Storage Checkpoint must have been created using the <code>dbed_ckptcreate(1M)</code> command. ■ You must unmount the Storage Checkpoint before you can remove it. ■ See the <code>dbed_ckptremove(1M)</code> manual page for more information.

To remove Storage Checkpoints

- ◆ Use the `dbed_ckptremove` command as follows:

```
$ /opt/VRTS/bin/dbed_ckptremove -S PROD \  
-c Checkpoint_971672042_wr001
```

Cloning the Oracle instance using `dbed_clonedb`

You can use the `dbed_clonedb` command to clone an Oracle instance using a Storage Checkpoint.

Cloning an existing database using a Storage Checkpoint must be done on the same host.

You have the option to manually or automatically recover the database when using the `dbed_clonedb` command:

- Manual (interactive) recovery, which requires using the `-i` option, of the clone database allows the user to control the degree of recovery by specifying which archive log files are to be replayed.
- Automatic (non-interactive) recovery, which is the default usage of the command, recovers the entire database and replays all of the archive logs. You will not be prompted for any archive log names.

Before cloning the Oracle instance, the following conditions must be met:

Prerequisites	■ You must first create a Storage Checkpoint. See “Creating Storage Checkpoints using dbed_ckptcreate” on page 140. ■ You must be logged in as the database administrator. ■ Make sure you have enough space and system resources to create a clone database on your system. ■ A clone database takes up as much memory and machine resources as the primary database.
Usage notes	■ The <code>dbed_clonedb</code> command is used to create a copy of a database, cloning all existing database files to new locations. ■ The <code>ORACLE_SID</code> and <code>ORACLE_HOME</code> environment variables must be set to the primary database. ■ It is assumed that the user has a basic understanding of the database recovery process. ■ See the <code>dbed_clonedb(1M)</code> manual page for more information.
Limitations for SF Oracle RAC	■ Note that the database cloning using Instant Checkpoint is not supported for SF Oracle RAC. ■ When you clone the database by using Checkpoint, the node can be any node in the same SF Oracle RAC cluster but the archive log destination is required to be on CFS file system. Otherwise, you must manually copy the archive log files.

[Table 5-2](#) lists the options for the `dbed_clonedb` command.

Table 5-2 `dbed_clonedb` command options

Option	Description
<code>-S CLONE_SID</code>	Specifies the name of the new Oracle SID, which will be the name of the new database instance.
<code>-m MOUNT_POINT</code>	Indicates the new mount point of the Storage Checkpoint.
<code>-c CKPT_NAME</code>	Indicates the name of the Storage Checkpoint.

Table 5-2 dbed_clonedb command options (continued)

Option	Description
-i	Runs the command in interactive mode where you must respond to prompts by the system. The default mode is non-interactive. (Optional)
-o umount	Shuts down the clone database and unmounts the Storage Checkpoint file system.
-o restartdb	Mounts the Storage Checkpoint file system and starts the clone database. The -o restartdb option will not attempt to recover the clone database.
-d	Used with the -o umount option. If the -d option is specified, the Storage Checkpoint used to create the clone database will be removed along with the clone database.
-p	Specifies a file containing initialization parameters to be modified or added to the clone database's initialization parameter file prior to startup. The format of the pfile_modification_file is the same as that of the Oracle initialization parameter file.

To clone an Oracle instance with manual Oracle recovery

- ◆ Use the dbed_clonedb command as follows:

```
$ /opt/VRTS/bin/dbed_clonedb -S NEW10 -m /local/ORA_HOME/1 \
-c Checkpoint_988813047 -i
Primary Oracle SID is TEST10i
New Oracle SID is NEW10
Checkpoint_988813047 not mounted at /local/oracle10g/1
Mounting Checkpoint_988813047 at /local/oracle10g/1
Using environment-specified parameter file
  /local/oracle10g/links/dbs/initTEST10i.ora
Default Oracle parameter file found:
  /local/oracle10g/links/dbs/initTEST10i.ora
Copying /local/oracle10g/links/dbs/initTEST10i.ora to
  /local/oracle10g/1/testvol
Control file 'ora_control2' path not explicitly specified in
init file; assuming ORACLE_HOME/dbs

All redo-log files found
Copying initTEST10i.ora to initNEW10.ora
  in /local/oracle10g/1/testvol
```

```
Altering db_name in initNEW10.ora
Altering control file locations in initNEW10.ora
Creating new link for clone database init file
Creating archive log directory

About to start up new database and begin reconfiguration
```

```
Database NEW10 is being reconfigured
Altering clone database archive log directory
Updating log_archive_dest in clone database init file
Found archive log destination at /testvol
```

```
The latest archive log(s) must now be applied. To apply the
logs, open a new window and perform the following steps:
1. copy required archive log(s) from primary to clone:
    primary archive logs in /testvol
    clone archive logs expected in /local/oracle10g/1/testvol
2. ORACLE_SID=NEW10; export ORACLE_SID # sh and ksh, OR
    setenv ORACLE_SID NEW10 #csh
3. /local/oracle10g/links/bin/sqlplus /nolog
4. CONNECT / AS SYSDBA
5. RECOVER DATABASE UNTIL CANCEL USING BACKUP CONTROLFILE
6. enter the archive log(s) you wish to apply
7. EXIT
```

```
Press <Return> after you have completed the above steps.
<Return>
```

```
Resetting logs on new database NEW10
Database instance NEW10 is up and running
```

To clone an Oracle instance with automatic Oracle recovery

- ◆ Use the `dbed_clonedb` command as follows:

```
$/opt/VRTS/bin/dbed_clonedb -S NEW10 -m /local/ORA_HOME/1 \
-c Checkpoint_988813047
Primary Oracle SID is TEST10i
New Oracle SID is NEW10
Checkpoint_988813047 not mounted at /local/oracle10g/1
Mounting Checkpoint_988813047 at /local/oracle10g/1
Using environment-specified parameter file
    /local/oracle10g/links/dbs/initTEST10i.ora
Default Oracle parameter file found:
```

```

/local/oracle10g/links/dbs/initTEST10i.ora
Copying /local/oracle10g/links/dbs/initTEST10i.ora
to /local/oracle10g/1/testvol
Control file 'ora_control2' path not explicitly specified in
init file; assuming ORACLE_HOME/dbs

All redo-log files found
Copying initTEST10i.ora to initNEW10.ora
in /local/oracle10g/1/testvol
Altering db_name in initNEW10.ora
Altering control file locations in initNEW10.ora
Creating new link for clone database init file
Creating archive log directory

About to start up new database and begin reconfiguration
Database NEW10 is being reconfigured
Starting automatic (full) database recovery
Shutting down clone database
Altering clone database archive log directory
Updating log_archive_dest in clone database init file
Found archive log destination at /testvol
Mounting clone database
Resetting logs on new database NEW10
Database instance NEW10 is up and running

```

To shut down the clone database and unmount the Storage Checkpoint

- ◆ Use the `dbed_clonedb` command as follows:

```
$ opt/VRTS/bin/dbed_clonedb -S NEW -o umount
```

To mount a Storage Checkpoint file system and start the clone database

- ◆ Use the `dbed_clonedb` command as follows:

```

$/opt/VRTS/bin/dbed_clonedb -S NEW -o restartdb
Database instance NEW is up and running.

```

To delete a clone database and the Storage Checkpoint used to create it

- ◆ Use the `dbed_clonedb` command as follows:

```
$ /opt/VRTS/bin/dbed_clonedb -S NEW -o umount -d
```

Using Database FlashSnap for backup and off-host processing

This chapter includes the following topics:

- [About Veritas Database FlashSnap](#)
- [Planning to use Database FlashSnap](#)
- [Preparing hosts and storage for Database FlashSnap](#)
- [About creating database snapshots](#)
- [FlashSnap commands](#)

About Veritas Database FlashSnap

Database FlashSnap lets you capture an online image of an actively changing database at a given instant, known as a snapshot. You can then perform backups and off-host processing tasks on these snapshots while still maintaining continuous availability of your critical data. Database FlashSnap offers you a flexible way to efficiently manage multiple point-in-time copies of your data, and reduce resource contention on your business-critical servers.

A database snapshot can be used on the same host as the production database or on a secondary host sharing the same storage.

A database snapshot can be used for the following off-host processing applications:

- Data backup
- Data warehousing

■ Decision-support queries

When the snapshot is no longer needed, the database administrator can import the original snapshot back to the primary host and resynchronize the snapshot to the original database volumes.

Database FlashSnap commands are executed from the command-line interface.

Database FlashSnap significantly reduces the time it takes to backup your database, increase the availability of your production database, and still maintain your production database's performance.

Note: The information in this chapter is only applicable for a SF Oracle RAC configuration. For information about any other configuration types and Database FlashSnap, please refer to the appropriate Storage Foundation documentation.

Solving typical database problems with Database FlashSnap

Database FlashSnap allows database administrators to create a snapshot without root privileges.

Database FlashSnap is designed to enable you to use database snapshots to overcome the following types of problems encountered in enterprise database environments:

- In many companies, there is a clear separation between the roles of system administrators and database administrators. Creating database snapshots typically requires superuser (root) privileges, privileges that database administrators do not usually have.
- In some companies, database administrators are granted root privileges, but managing storage is typically neither central to their job function nor their core competency.
- Creating database snapshots is a complex process, especially in large configurations where thousands of volumes are used for the database. One mistake can render the snapshots useless.

Because root privileges are not required, Database FlashSnap overcomes these obstacles by enabling database administrators to easily create consistent snapshots of the database. The snapshots can be utilized for repetitive use.

Database FlashSnap applications

[Table 6-1](#) displays typical applications of Database FlashSnap.

Table 6-1 Typical Database FlashSnap applications

Application	Description
Database Backup and Restore	Enterprises require 24/7 online data availability. Enterprises cannot afford the downtime involved in backing up critical data offline. By creating a clone database or a duplicate volume snapshot of data, and then using it to back up your data, your business-critical applications can continue to run without extended down time or impacted performance. After a clone database or snapshot volume is created, it can be used as a source to back up the original database.
Decision-Support Analysis and Reporting	Operations such as decision-support analysis and business reporting may not require access to real-time information. You can direct such operations to use a clone database that you have created from snapshots using Database FlashSnap, rather than allowing them to compete for access to the primary volume or database. When required, you can quickly resynchronize the clone database with the primary database to get up-to-date information.
Application Development and Testing	Development or service groups can use a clone database created with Database FlashSnap as a test database for new applications. A clone database provides developers, system testers, and quality assurance groups with a realistic basis for testing the robustness, integrity, and performance of new applications.
Logical Error Recovery	Logical errors caused by an administrator or an application program can compromise the integrity of a database. You can recover a database by restoring the database files from a volume snapshot or by recovering logical objects (such as tables, for example) from a clone database created from volume snapshots using Database FlashSnap. These solutions are faster than fully restoring database files from tape or other backup media.

Using Database FlashSnap

Database snapshot requirements are defined in a file called a snapplan.. The system administrator needs to configure storage according to the requirements specified in the snapplan.

See [“Preparing hosts and storage for Database FlashSnap”](#) on page 162.

Database FlashSnap allows you to check the storage setup against requirements set forth in the snapplan. Depending on the results, the database administrator may need to modify the snapplan or the system administrator may need to adjust the storage configuration. Properly configuring storage is the only aspect of using Database FlashSnap that requires the system administrator's participation.

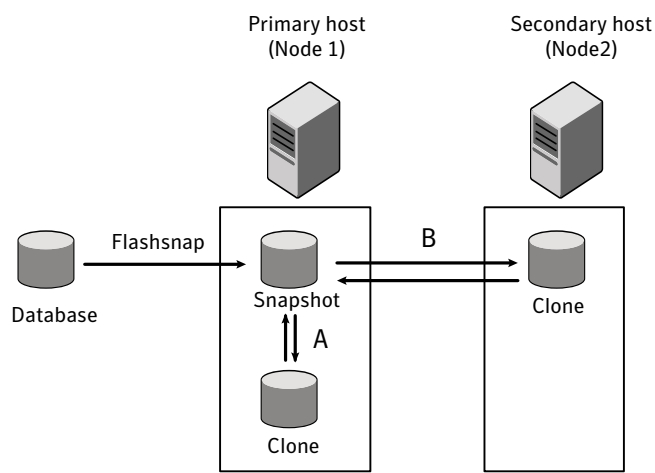
To use Database FlashSnap, a database administrator must first define the snapshot requirements. For example, the database administrator needs to determine whether off-host processing is required and, if it is, which host should be used for it. In addition, it is also important to consider how much database downtime can be tolerated. Database snapshot requirements are defined in the snapplan file.

See [“Creating a snapplan \(dbed_vmchecksnap\)”](#) on page 182.

After creating the snapplan, the database administrator must validate it to ensure that it is correct. During validation, the snapplan is copied to the repository before using it to create a snapshot. Depending on the validation results, the database administrator may need to modify the snapplan or the system administrator may need to adjust the storage configuration.

After storage is configured as specified in the snapplan and the snapplan has been validated, the database administrator can create snapshots of the database and create database clones based on the snapshots on either the same host or a secondary one (see [Figure 6-1](#) below).

Figure 6-1 Database FlashSnaps on either the primary or secondary host



A database clone can be used on a secondary host for off-host processing, including decision-support analysis and reporting, application development and testing, database backup, and logical error recovery. After a user has finished using the

clone on a secondary host, the database administrator can shut down the clone and move the snapshot database back to the primary host. Regardless of whether a snapshot is used on the primary or secondary host, it can be resynchronized with the primary database using Database FlashSnap. Database FlashSnap uses Veritas Volume Manager FastResync to quickly resynchronize the changed section between the primary and snapshot.

See the *Veritas Volume Manager User's Guide* for details about the Volume Manager FastResync.

Using Database FlashSnap commands

Table 6-2 describes the three Database FlashSnap commands. All of these commands can be executed by the Oracle database administrator and do not require superuser (root) privileges.

Table 6-2 Database FlashSnap commands

Command	Description
<code>dbed_vmchecksnap</code>	This command creates and validates the snapshot plan that is used to create a snapshot image of an Oracle database. You can also use this command to copy, list, or remove a snapplan or make sure the storage is properly configured for the task.
<code>dbed_vmsnap</code>	This command is used to create a snapshot image of an Oracle database by splitting the mirror volumes used by the database. You can also use the <code>dbed_vmsnap</code> command to resynchronize snapshot volumes with their original volumes. Note: Snapplan creation and validation is only supported on CVM master host.
<code>dbed_vmclonedb</code>	This command mounts and starts a clone database using snapshot volumes. It can also shut down a clone database and deport its volumes, as well as restart a clone database that has been shut down. Note: The <code>dbed_vmclonedb</code> command does not support instant snapshot for database cloning.

Using the Database FlashSnap online option

The Database FlashSnap supports an online option for creating database snapshots. With the online option, the tablespaces are put into online backup mode before the snapshot is created. This type of snapshot is a valid database backup.

Note: This option can be used if you are performing a point-in-time recovery from logical errors.

In this release of SF Oracle RAC, Database FlashSnap supports third mirror break-off snapshots only. Third mirror break-off snapshots are fully synchronized, full-sized snapshots.

See the *Veritas Volume Manager Administrator's Guide* for more information.

Planning to use Database FlashSnap

Selecting the snapshot mode

Select the online option for the snapshot mode:

See [“Using the Database FlashSnap online option”](#) on page 161.

Preparing hosts and storage for Database FlashSnap

Review the following details to prepare the hosts and storage for Database FlashSnap.

Setting up hosts

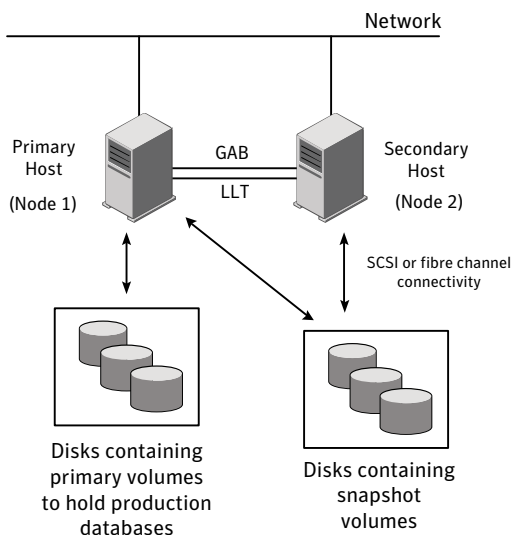
Database FlashSnap requires sufficient disk space in the disk group to add a mirror of equal size of the existing database.

Setting up a storage configuration for Database FlashSnap operations is a system administrator's responsibility and requires superuser (root) privileges. Database FlashSnap utilities do not address setting up an appropriate storage configuration.

Database FlashSnap in a cluster configuration

A Database FlashSnap configuration with two hosts (nodes) in a cluster allows CPU- and I/O-intensive operations to be performed for online backup and decision support without degrading the performance of the primary host running the production database. Both the primary and secondary hosts share the storage in which the snapshot database is created. Both the primary and secondary hosts have access to the disks containing the snapshot volumes.

[Figure 6-2](#) shows a Database FlashSnap configuration with two hosts (nodes) in a cluster.

Figure 6-2 Example of off-host Database FlashSnap solution

Host and storage requirements

Before using Database FlashSnap, ensure that the following requirements are met:

- All files are on VxFS file systems over VxVM volumes. Raw devices are not supported.
- Symbolic links to datafiles are not supported.
- ORACLE_HOME is on a separate file system.
- Archive logs are on a separate VxFS file system and are separate from the VxFS file system containing Oracle data files or ORACLE_HOME.
- The database does not contain BFILES and external tables.
- Oracle datafiles, archive logs, redo logs, and control files are in a single disk group.

Creating a snapshot mirror of a volume or volume set used by the database

With Database FlashSnap, you can mirror the volumes used by the database to a separate set of disks, and those mirrors can be used to create a snapshot of the database. These snapshot volumes can be split and placed in a separate disk group. This snapshot disk group can be imported on a separate host, which shares the

same storage with the primary host. The snapshot volumes can be resynchronized periodically with the primary volumes to get recent changes of the datafiles. If the primary datafiles become corrupted, you can quickly restore them from the snapshot volumes. Snapshot volumes can be used for a variety of purposes, including backup and recovery, and creating a clone database.

You must create snapshot mirrors for all of the volumes used by the database datafiles before you can create a snapshot of the database. This section describes the procedure used to create snapshot mirrors of volumes.

Use the `vxsnap` command to create a snapshot mirror or synchronize a snapshot mirror.

Prerequisites

- You must be logged in as superuser (root).
- The disk group must be version 110 or later.
For more information on disk group versions, see the `vxchg(1M)` online manual page.
- Be sure that a data change object (DCO) and a DCO log volume are associated with the volume for which you are creating the snapshot.
- Persistent FastResync must be enabled on the existing database volumes and disks must be assigned for the snapshot volumes.
FastResync optimizes mirror resynchronization by tracking updates to stored data that have been missed by a mirror. When a snapshot mirror is reattached to its primary volumes, only the updates that were missed need to be re-applied to resynchronize it. FastResync increases the efficiency of the volume snapshot mechanism to better support operations such as backup and decision support.
For detailed information about FastResync, see the *Veritas Volume Manager Administrator's Guide*.
- Snapshot mirrors and their associated DCO logs should be on different disks than the original mirror plexes, and should be configured correctly for creating snapshots by the system administrator.
- When creating a snapshot mirror, create the snapshot on a separate controller and separate disks from the primary volume.
- Allocate separate volumes for archive logs.
- Do not place any datafiles, including control files, in the `$ORACLE_HOME/dbs` directory.

Usage Notes

- Create a separate disk group for Oracle database-related files.
- Do not share volumes between Oracle database files and other software.
- ORACLE_HOME cannot be included in the snapshot mirror.
- Resynchronization speed varies based on the amount of data changed in both the primary and snapshot volumes during the break-off time.
- Do not share any disks between the original mirror and the snapshot mirror.
- Snapshot mirrors for datafiles and archive logs should be created so that they do not share any disks with the data of the original volumes. If they are not created in this way, the VxVM disk group cannot be split and, as a result, Database FlashSnap will not work.

Note: Database FlashSnap commands support third-mirror break-off snapshots only. The snapshot mirror must be in the SNAPDONE state.

The following sample procedure is for existing volumes without existing snapshot plexes or associated snapshot volumes. In this procedure, `volume_name` is the name of either a volume or a volume set.

Note: You must be logged in as superuser (root) to issue the commands in the following procedure.

To create a snapshot mirror of a volume or volume set

- 1 Enter the following command on the hosts (nodes) in your cluster to locate the CVM master node:

```
# vxdctl -c mode
```

In an example with an application cluster consisting of two nodes (nebula and galaxy), on the node which is the master the following output appears:

```
mode: enabled: cluster active - MASTER
master: galaxy
```

On the node which is the slave, the following output appears:

```
mode: enabled: cluster active - SLAVE
master: galaxy
```

After determining the CVM master node in your application cluster, be sure to issue the following commands in this procedure on that CVM master node.

- 2 To prepare the volume for being snapshot, use the `vxsnap prepare` command:

```
# vxsnap -g diskgroup prepare volume \
alloc="storage_attribute ..."
```

The `vxsnap prepare` command automatically creates a DCO and DCO volumes and associates them with the volume, and enables Persistent FastResync on the volume. Persistent FastResync is also set automatically on any snapshots that are generated from a volume on which this feature is enabled.

For enabling persistent FastResync on a volume in VxVM 4.1 or 5.0, either from the command line or from within a script, use the `vxsnap prepare` command as described above.

- 3 To verify that FastResync is enabled on the volume, use the `vxprint` command:

```
# vxprint -g diskgroup -F%fastresync volume_name
```

This returns on if FastResync is on. Otherwise, it returns off.

- 4 To verify that a DCO and DCO log volume are attached to the volume, use the `vxprint` command:

```
# vxprint -g diskgroup -F%hasdcolog volume_name
```

This returns on if a DCO and DCO log volume are attached to the volume. Otherwise, it returns off.

- 5 Create a mirror of a volume:

```
# vxsnap -g diskgroup addmir volume_name alloc=diskname
```

There is no option for creating multiple mirrors at the same time. Only one mirror can be created at a time.

- 6 List the available mirrors:

```
# vxprint -g diskgroup -F%name -e"pl_v_name in \"volume_name\""
```

- 7 Enable database FlashSnap to locate the correct mirror plexes when creating snapshots:

- Set the `dbed_flashsnap` tag for the data plex you want to use for breaking off the mirror. You can choose any tag name you like, but it needs to match the tag name specified in the snapplan.

```
# vxedit -g diskgroup set putil2=dbed_flashsnap plex_name
```

- Verify that the `dbed_flashsnap` tag has been set to the desired data plex:

```
# vxprint -g diskgroup -F%name -e"pl_v_name in \
\"volume_name\" && p2 in \"dbed_flashsnap\""
```

If you require a backup of the data in the snapshot, use an appropriate utility or operating system command to copy the contents of the snapshot to tape or to some other backup medium.

Example procedure to create a snapshot mirror of a volume

This example shows the steps involved in creating a snapshot mirror for the volume `data_vol` belonging to the disk group `PRODDg`.

Note: You must be logged in as superuser (root) to issue the commands in the following procedure.

To create a snapshot mirror of the volume data_vol

- 1 Prepare the volume data_vol for mirroring:

```
# vxsnap -g PRODDg prepare data_vol alloc=PRODDg01,PRODDg02
```

- 2 Verify that FastResync is enabled:

```
# vxprint -g PRODDg -F%fastresync data_vol  
  
on
```

- 3 Verify that a DCO and a DCO log are attached to the volume:

```
# vxprint -g PRODDg -F%hasdcolog data_vol  
  
on
```

- 4 Create a snapshot mirror of data_vol:

```
# vxsnap -g PRODDg addmir data_vol alloc=PRODDg02
```

- 5 List the data plexes:

```
# vxprint -g PRODDg -F%name -e"pl_v_name in \"data_vol\""  
  
data_vol-01  
  
data_vol-02
```

- 6 Choose the plex that is in the SNAPDONE state. Use the vxprint -g diskgroup command to identify the plex that is in the SNAPDONE state.

- 7 Identify the plex name in the above step and set the dbed_flashsnap tag for it:

```
# vxedit -g PRODDg set putil2=dbed_flashsnap data_vol-02
```


- 8 Verify that the `dbed_flashsnap` tag has been set to the desired data plex, `data_vol-02`:

```
# vxprint -g PRODDg -F%name -e"pl_v_name in \"data_vol\" \"  
&& p2 in \"dbed_flashsnap\""  
data_vol-02
```

9 To verify that the snapshot volume was created successfully, use the `vxprint -g dg` command as follows:

```
# vxprint -g PRODdg
```

The following output appears on a system running the Solaris OS.

TY	NAME	ASSOC	KSTATE	LENGTH	PLOFFS	STATE	TUTIL0	PUTIL0
dg	PRODdg	PRODdg	-	-	-	-	-	-
dm	PRODdg01	clt2d0s2	-	35358848	-	-	-	-
dm	PRODdg02	clt3d0s2	-	17674896	-	-	-	-
dm	PRODdg03	clt1d0s2	-	17674896	-	-	-	-
		v data_vol	fsgen					
		ENABLED	4194304			-	ACTIVE	-
		pl data_vol-01	data_vol					
		ENABLED	4194304			-	ACTIVE	-
		sd PRODdg03-01	data_vol-01					
		ENABLED	4194304			0	-	-
		pl data_vol-02	data_vol					
		ENABLED	4194304			-	SNAPDONE	-
		sd PRODdg02-01	data_vol-02					
		ENABLED	4194304			0	-	-
		dc data_vol_dco	data_vol					
		-	-			-	-	-
		v data_vol_dcl	gen					
		ENABLED	560			-	ACTIVE	-
		pl data_vol_dcl-01	data_vol_dcl	ENABLED				
		560	-	ACTIVE	-	-		
		sd PRODdg01-01	data_vol_dcl-01	ENABLED				
		560	0	-	-	-		
		pl data_vol_dcl-02	data_vol_dcl	DISABLED				
		560	-	DCOSNP	-	-		
		sd PRODdg02-02	data_vol_dcl-02	ENABLED				
		560	0	-	-	-		

Identify that the specified plex is in the SNAPDONE state. In this example, it is data_vol-02.

The snapshot mirror is now ready to be used.

Upgrading existing volumes to use Veritas Volume Manager 5.0

The procedure in this section describes how to upgrade a volume created using a version older than VxVM 5.0, so that it can take advantage of Database FlashSnap.

Note the following requirements and caveats for this procedure:

- The plexes of the DCO volume require persistent storage space on disk to be available. To make room for the DCO plexes, you may need to add extra disks to the disk group, or reconfigure existing volumes to free up space in the disk group. Another way to add disk space is to use the disk group move feature to bring in spare disks from a different disk group.
- Existing snapshot volumes created by the `vxassist` command are not supported. A combination of snapshot volumes created by `vxassist` and `vxsnap` are also not supported.
- You must be logged in as superuser (root) to issue the commands in the following procedure. Additionally, all operations involving the creation or modification using the commands `vxassist` or `vxdg` require that the user perform the task on the master CVM node.

To upgrade an existing volume created with an earlier version of VxVM

- 1 Upgrade the disk group that contains the volume, to a version 120 or higher, before performing the remainder of the procedure described in this section. Use the following command to check the version of a disk group:

```
# vxdg list diskgroup
```

To upgrade a disk group to the latest version, use the following command:

```
# vxdg upgrade diskgroup
```

- 2 If the volume to be upgraded has a DRL plex or subdisk from an earlier version of VxVM, use the following command to remove this:

```
# vxassist [-g diskgroup] remove log volume [nlog=n]
```

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

- 3 For a volume that has one or more associated snapshot volumes, use the following command to reattach and resynchronize each snapshot:

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

If persistent FastResync was enabled on the volume before the snapshot was taken, the data in the snapshot plexes is quickly resynchronized from the original volume. If persistent FastResync was not enabled, a full resynchronization is performed.

- 4 Use the following command to turn off persistent FastResync for the volume:

```
# vxvol [-g diskgroup] set fastresync=off volume
```

- 5 Use the following command to dissociate a DCO object from an earlier version of VxVM, DCO volume and snap objects from the volume:

```
# vxassist [-g diskgroup] remove log volume logtype=dcv
```

- 6 Use the following command on the volume to upgrade it:

```
# vxsnap [-g diskgroup] prepare volume \  
alloc="disk_name1,disk_name2"
```

Provide two disk names to avoid overlapping the storage of the snapshot DCO plex with any other non-moving data or DCO plexes.

The `vxsnap prepare` command automatically enables persistent FastResync on the volume and on any snapshots that are generated from it. It also associates a DCO and DCO log volume with the volume to be snapshot.

- 7 To view the existing DCO plexes and see whether there are enough for the existing data plexes, enter:

```
# vxprint -g diskgroup
```

There needs to be one DCO plex for each existing data plex.

- 8 If there are not enough DCO plexes for the existing data plexes, create more DCO plexes:

```
# vxsnap [-g diskgroup] addmir dcv_volume_name \  
[alloc=disk_name]
```

where *dcv_volume_name* is the name of the DCO volume you are creating.

- 9 If the plex is in a SNAPDONE state, convert it to an ACTIVE state:

```
# vxplex [-g diskgroup] convert state=ACTIVE data_plex
```

- 10 Convert the data plexes to a SNAPDONE state and associate a DCO plex with the data plex that will be used for snapshot operations:

```
# vxplex [-g diskgroup] -o dcoplex=dcv_plex_name convert \state=SNAPDONE data_plex
```

where *dcv_plex_name* is the name of the DCO plex you are creating.

Example procedure to upgrade existing volumes to use Veritas Volume Manager 5.0

Note: You must be logged in as superuser (root) to issue the commands in the following procedure. Additionally, all operations involving the creation or modification using the commands `vxassist` or `vxchg` require that the user perform the task on the master CVM node.

In this example, the volume, `data_vol`, is upgraded to make use of VxVM 5.0 features.

To upgrade an existing volume created with an earlier version of VxVM

- 1 Upgrade the disk group, `PRODDg`.

`vxchg upgrade PRODDg`
- 2 Remove the DRL plexes or subdisks, belonging to an earlier version of VxVM, from the volume to be upgraded.

`vxassist -g PRODDg remove log data_vol logtype=drl`
- 3 Reattach any snapshot volume back to the primary volume to be upgraded.

`vxsnap -g PRODDg snapback SNAP-data_vol`
- 4 Turn off FastResync on the volume to be upgraded.

`vxvol -g PRODDg set fastresync=off data_vol`
- 5 Disassociate and remove any older DCO object and DCO volumes.

`vxassist -g PRODDg remove log data_vol logtype=dco`
- 6 Upgrade the volume by associating a new DCO object and DCO volume.

`vxsnap -g PRODDg prepare data_vol alloc="PRODDg01,PRODDg02"`
- 7 View the existing DCO plexes and plex state.

Scenario 1

In this scenario, there are enough DCO plexes for the data plexes. Also, no data plex is associated with a DCO plex.

```
# vxprint -g PRODDg
```

The following output appears on a system running the Solaris OS.

TY	NAME	ASSOC	KSTATE	LENGTH	PLOFFS	STATE	TUTILO	PUTILO
dg	PRODdg	PRODdg	-	-	-	-	-	-
dm	PRODdg01	clt2d0s2	-	35358848	-	-	-	-
dm	PRODdg02	clt3d0s2	-	17674896	-	-	-	-
dm	PRODdg03	clt1d0s2	-	17674896	-	-	-	-
		v data_vol	fsgen					
		ENABLED	4194304		-	ACTIVE	-	-
		pl data_vol-01	data_vol					
		ENABLED	4194304		-	ACTIVE	-	-
		sd PRODdg01-01	data_vol-01					
		ENABLED	4194304		0	-	-	-
		pl data_vol-04	data_vol					
		ENABLED	4194304		-	SNAPDONE	-	-
		sd PRODdg02-03	data_vol-04					
		ENABLED	4194304		0	-	-	-
		dc data_vol_dco	data_vol					
		-	-		-	-	-	-
		v data_vol_dcl	gen					
		ENABLED	560		-	ACTIVE	-	-
		pl data_vol_dcl-01	data_vol_dcl					
		ENABLED	560		-	ACTIVE	-	-
		sd PRODdg01-02	data_vol_dcl-01					
		ENABLED	560		0	-	-	-
		pl data_vol_dcl-02	data_vol_dcl					
		ENABLED	560		-	ACTIVE	-	-
		sd PRODdg02-02	data_vol_dcl-02					
		ENABLED	560		0	-	-	-

■ Convert the data plex state from SNAPDONE to ACTIVE.

```
# vxplex -g PRODdg convert state=ACTIVE data_vol-04
```

■ Associate the data plex with a new DCO plex and convert it back to a SNAPDONE state.

```
# vxplex -g PRODdg -o dcoplex=data_vol_dcl-02 \  
convert state=SNAPDONE data_vol-04  
  
# vxprint -g PRODdg
```

The following output appears on a system running the Solaris OS.

TY NAME	ASSOC	KSTATE	LENGTH	PLOFFS	STATE	TUTIL0	PUTIL0
dg PRODDg	PRODDg	-	-	-	-	-	-
dm PRODDg01	c1t2d0s2	-	35358848	-	-	-	-
dm PRODDg02	c1t3d0s2	-	17674896	-	-	-	-
dm PRODDg03	c1t1d0s2	-	17674896	-	-	-	-
	pl data_vol-03	-					
	DISABLED		4194304		-	-	-
	sd PRODDg02-01 data_vol-03						
	ENABLED		4194304		0	-	-
	v data_vol	fsgen					
	ENABLED		4194304		-	ACTIVE	-
	pl data_vol-01 data_vol						
	ENABLED		4194304		-	ACTIVE	-
	sd PRODDg01-01 data_vol-01						
	ENABLED		4194304		0	-	-
	pl data_vol-04 data_vol						
	ENABLED		4194304		-	SNAPDONE	-
	sd PRODDg02-03 data_vol-04						
	ENABLED		4194304		0	-	-
	dc data_vol_dco data_vol						
	-	-			-	-	-
	v data_vol_dcl gen						
	ENABLED		560		-	ACTIVE	-
	pl data_vol_dcl-01 data_vol_dcl						
	ENABLED		560		-	ACTIVE	-
	sd PRODDg01-02 data_vol_dcl-01						
	ENABLED		560		0	-	-
	pl data_vol_dcl-02 data_vol_dcl						
	DISABLED		560		-	DCOSNP	-
	sd PRODDg02-02 data_vol_dcl-02						
	ENABLED		560		0	-	-

Scenario 2

In this scenario, there are fewer DCO plexes than data plexes.

```
# vxprint -g PRODDg
```

The following output appears on a system running the Solaris OS.

TY NAME	ASSOC	KSTATE	LENGTH	PLOFFS	STATE	TUTIL0	PUTIL0
dg PRODDg	PRODDg	-	-	-	-	-	-
dm PRODDg01	c1t2d0s2	-	35358848	-	-	-	-

dm	PRODdg02	c1t3d0s2	-	17674896	-	-	-	-
dm	PRODdg03	c1t1d0s2	-	17674896	-	-	-	-
	pl	data_vol-03	-					
		DISABLED		4194304	-	-	-	-
	sd	PRODdg02-01	data_vol-03					
		ENABLED		4194304	0	-	-	-
	v	data_vol	fsgen					
		ENABLED		4194304	-	ACTIVE	-	-
	pl	data_vol-01	data_vol					
		ENABLED		4194304	-	ACTIVE	-	-
	sd	PRODdg01-01	data_vol-01					
		ENABLED		4194304	0	-	-	-
	pl	data_vol-04	data_vol					
		ENABLED		4194304	-	ACTIVE	-	-
	sd	PRODdg02-03	data_vol-04					
		ENABLED		4194304	0	-	-	-
	dc	data_vol_dco	data_vol					
	-	-	-	-	-	-	-	-
	v	data_vol_dcl	gen					
		ENABLED		560	-	ACTIVE	-	-
	pl	data_vol_dcl-01	data_vol_dcl					
		ENABLED		560	-	ACTIVE	-	-
	sd	PRODdg01-02	data_vol_dcl-01					
		ENABLED		560	0	-	-	-

- Add a DCO plex to the DCO volume using the vxassist mirror command.

```
# vxsnap -g PRODdg addmir data_vol_dcl alloc=PRODdg02
```

- Associate the data plex with the new DCO plex and convert it to a SNAPDONE state.

The following command is used for a system running the Solaris OS.

```
# vxplex -g PRODdg -o dcoplex=data_vol_dcl-02 \  
convert state=SNAPDONE data_vol-04
```

The following output appears on a system running the Solaris OS.

TY	NAME	ASSOC	KSTATE	LENGTH	PLOFFS	STATE	TUTIL0	PUTIL0
dg	PRODdg	PRODdg	-	-	-	-	-	-
dm	PRODdg01	c1t2d0s2	-	35358848	-	-	-	-
dm	PRODdg02	c1t3d0s2	-	17674896	-	-	-	-
dm	PRODdg03	c1t1d0s2	-	17674896	-	-	-	-

pl data_vol-03	-				
DISABLED	4194304	-	-	-	-
v data_vol	fsgen				
ENABLED	4194304	-	ACTIVE	-	-
pl data_vol-01	data_vol				
ENABLED	4194304	-	ACTIVE	-	-
sd PRODDg01-01	data_vol-01				
ENABLED	4194304	0	-	-	-
pl data_vol-04	data_vol				
ENABLED	4194304	-	SNAPDONE	-	-
sd PRODDg02-03	data_vol-04				
ENABLED	4194304	0	-	-	-
dc data_vol_dco	data_vol				
-	-	-	-	-	-
v data_vol_dcl	gen				
ENABLED	560	-	ACTIVE	-	-
pl data_vol_dcl-01	data_vol_dcl				
ENABLED	560	-	ACTIVE	-	-
sd PRODDg01-02	data_vol_dcl-01				
ENABLED	560	0	-	-	-
pl data_vol_dcl-02	data_vol_dcl				
DISABLED	560	-	DCOSNP	-	-
sd PRODDg02-02	data_vol_dcl-02				
ENABLED	560	0	-	-	-

About creating database snapshots

A snapshot can be a source for backing up the database or creating a clone database for decision-support purposes. You can use Database FlashSnap commands to create a snapshot of your entire database on the same host (node) or on a different one.

Online database snapshots

Database FlashSnap supports online database snapshot types.

When the SNAPSHOT_MODE specified in the snapplan is set to online, the dbed_vmsnap command first puts the tablespaces to be snapshot into backup mode. After the snapshot is created, the tablespaces are taken out of backup mode, the log files are switched to ensure that the extra redo logs are archived, and a snapshot of the archive logs is created.

Online snapshots provide a valid backup copy of the database

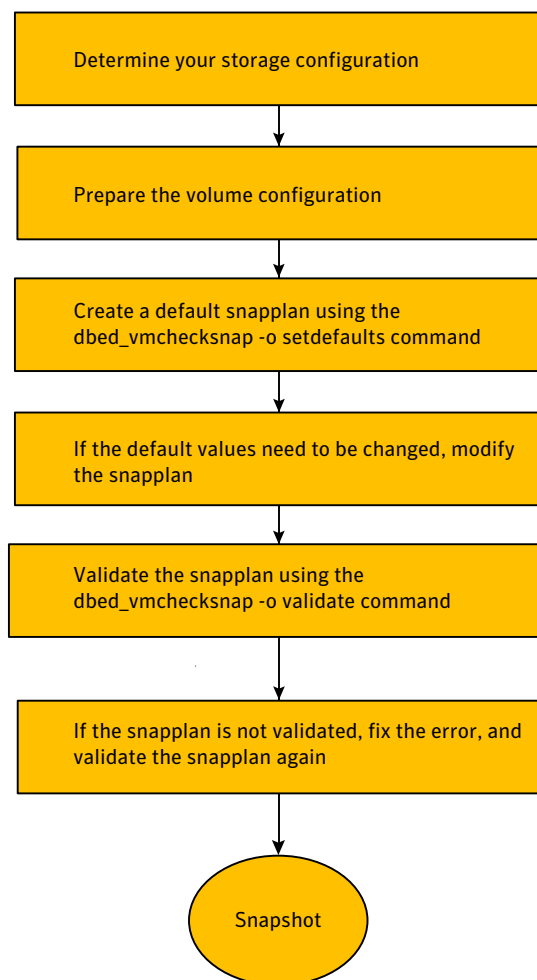
Tasks before creating a snapshot

Review the details on how to create snapshots of all volumes on a database using the snapplan.

Optionally, you can use the VxVM command (`vxsnap`) to create volume snapshots. However, unlike the Database FlashSnap commands, the `vxsnap` command does not automate disk group content reorganization functions.

[Figure 6-3](#) depicts the sequence of steps leading up to taking a snapshot using Database FlashSnap.

Figure 6-3 Prerequisites for creating a snapshot of your database



Creating a snapshot

Make sure the volumes used by the database are configured properly before attempting to take a snapshot. This database configuration requires superuser (root) privileges.

Note: Database FlashSnap commands must be run by the Oracle database administrator.

Whenever you change the structure of the database (for example, by adding or deleting datafiles, converting PFILE to SPFILE, or converting SPFILE to PFILE), you must run `dbed_update`. For example:

```
$ /opt/VRTS/bin/dbed_update -s $ORACLE_SID -H $ORACLE_HOME
```

To create a snapshot image of a database

- 1 Create a snapshot mirror of a volume or volume set.

See [“To create a snapshot mirror of a volume or volume set”](#) on page 166.

- 2 Use the `dbed_vmchecksnap` command to create a snapplan template and check the volume configuration to ensure that it is valid for creating volume snapshots of the database.

The snapplan contains detailed database and volume configuration information that is needed for snapshot creation and resynchronization. You can modify the snapplan template with a text editor.

The `dbed_vmchecksnap` command can also be used to:

List all snapplans associated with a specific ORACLE_SID	<code>dbed_vmchecksnap -o list</code>

Remove the snapplan from the SFDB repository	<code>dbed_vmchecksnap -o remove -f SNAPPLAN</code>

Copy a snapplan from the SFDB repository to your local directory	<code>dbed_vmchecksnap -o copy -f SNAPPLAN</code>

See [“Creating a snapplan \(dbed_vmchecksnap\)”](#) on page 182.

- 3 Use the `dbed_vmsnap` command to create snapshot volumes for the database.

See [“Creating a snapshot \(dbed_vmsnap\)”](#) on page 194.

- 4 On the secondary host, use the `dbed_vmclonedb` command to create a clone database using the disk group deported from the primary host. For more information:

See “Cloning a database (`dbed_vmclonedb`)” on page 201.

If the primary and secondary hosts specified in the snapplan are different, the `dbed_vmclonedb` command takes the following actions:

- Imports the disk group that was deported from the primary host
- Recovers the snapshot volumes
- Mounts the file systems
- Recovers the database
- Brings the database online with a different Oracle SID name than the primary host.

If the secondary host is different, the database name can be same. You can use the `-o recoverdb` option to let `dbed_vmclonedb` perform an automatic database recovery, or you can use the `-o mountdb` option to perform your own point-in-time recovery and bring up the database manually. For a point-in-time recovery, the snapshot mode must be online.

You can also create a clone on the primary host. Your snapplan settings specify whether a clone should be created on the primary or secondary host.

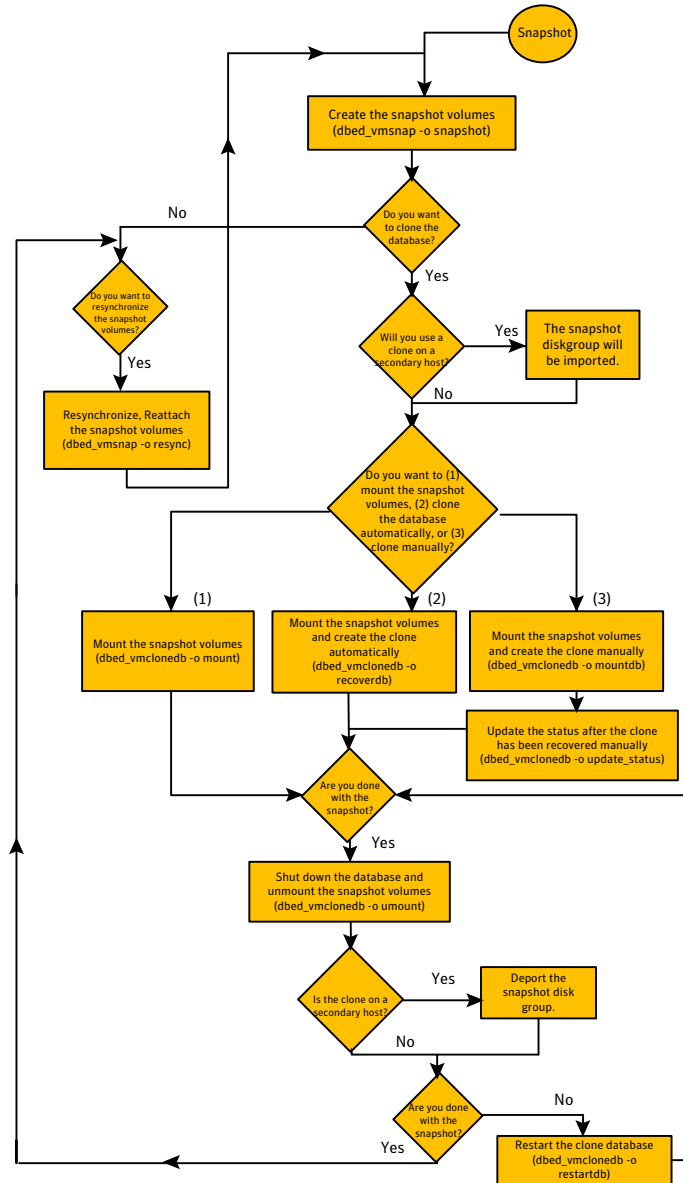
- 5 You can now use the clone database to perform database backup and other off-host processing work.
- 6 The clone database can be used to reverse resynchronize the original volume from the data in the snapshot, or can be discarded by rejoining the snapshot volumes with the original volumes (that is, by resynchronizing the snapshot volumes) for future use.

Tasks after creating a snapshot

There are many actions you can take after creating a snapshot of your database using Database FlashSnap. You can create a clone of the database for backup and off-host processing purposes. You can resynchronize the snapshot volumes with the primary database. In the event of primary database failure, you can recover it by reverse resynchronizing the snapshot volumes.

[Figure 6-4](#) is a flow chart that depicts the actions you can perform after creating a snapshot of your database using Database FlashSnap.

Figure 6-4 Actions you can perform after creating a snapshot of your database



FlashSnap commands

Database FlashSnap actions can be performed by using the following FlashSnap commands described in the following sections:

- [Creating a snapplan \(dbed_vmchecksnap\)](#)
- [Validating a snapplan \(dbed_vmchecksnap\)](#)
- [Displaying, copying, and removing a snapplan \(dbed_vmchecksnap\)](#)
- [Creating a snapshot \(dbed_vmsnap\)](#)
- [Backing up the database from snapshot volumes \(dbed_vmclonedb\)](#)
- [Cloning a database \(dbed_vmclonedb\)](#)
- [Resynchronizing the snapshot to your database](#)
- [Removing a snapshot volume](#)

Creating a snapplan (dbed_vmchecksnap)

The `dbed_vmchecksnap` command creates a snapplan that `dbed_vmsnap` uses to create a snapshot of an Oracle database.

You can name a snapplan file whatever you choose. Each entry in the snapplan file is a line in `parameter=argument` format.

[Table 6-3](#) describes the parameters that can be set when using the `dbed_vmchecksnap` command to create or validate a snapplan.

Table 6-3 Parameter values for `dbed_vmchecksnap`

Parameter	Value
SNAPSHOT_VERSION	Specifies the snapshot version for this major release of SF Oracle RAC.
PRIMARY_HOST	The name of the host where the primary database resides.
SECONDARY_HOST	The name of the host where the database will be imported.
PRIMARY_DG	The name of the VxVM disk group used by the primary database.

Table 6-3 Parameter values for `dbed_vmchecksnap` (*continued*)

Parameter	Value
SNAPSHOT_DG	The name of the disk group containing the snapshot volumes. The snapshot volumes will be put into this disk group on the primary host and deported. The secondary host will import this disk group to start a clone database.
ORACLE_SID	The name of the Oracle database. By default, the name of the Oracle database is included in the snapplan.
ARCHIVELOG_DEST	The full path of the archive logs. There are several archive log destinations that can be used for database recovery if you are multiplexing the archive logs. You must specify which archive log destination to use. It is recommended that you have the archive log destination on a separate volume if <code>SNAPSHOT_ARCHIVE_LOG</code> is yes.
SNAPSHOT_ARCHIVE_LOG	yes or no Specifies whether to create a snapshot of the archive log volumes. Specify yes to split the archive log volume mirrors and deport them to the secondary host. When using the Oracle remote archive log destination feature to send the archive logs to the secondary host, you can specify no to save some space. Because the archive logs may not always be delivered to the secondary host reliably, it is recommended that you specify yes.
SNAPSHOT_MODE	online Specifies what the database snapshot should be (online). ■ If the snapshot is created while the database is online, the <code>dbed_vmsnap</code> command will put the tablespaces into backup mode. After <code>dbed_vmsnap</code> finishes creating the snapshot, it will take the tablespaces out of backup mode, switch the log files to ensure that the extra redo logs are archived, and create a snapshot of the archived logs.

Table 6-3 Parameter values for `dbed_vmchecksnap` (*continued*)

Parameter	Value
SNAPSHOT_PLAN_FOR	The default value is database and cannot be changed. Specifies the database object for which you want to create a snapshot.
SNAPSHOT_PLEX_TAG	Specifies the snapshot plex tag. Use this variable to specify a tag for the plexes to be snapshot. The maximum length of the <code>plex_tag</code> is 15 characters. The default plex tag is <code>dbed_flashsnap</code> .
SNAPSHOT_VOL_PREFIX	Specifies the snapshot volume prefix. Use this variable to specify a prefix for the snapshot volumes split from the primary disk group. A volume name cannot be more than 32 characters. You should consider the length of the volume name when assigning the prefix.
SNAPSHOT_MIRROR	Specifies the number of plexes to be snapshot. The default value is 1.

When you first run `dbed_vmchecksnap`, use the `-o setdefaults` option to create a `snapplan` using default values for variables. You may then edit the file manually to set the variables for different snapshot scenarios.

Note: The `dbed_vmchecksnap -o validate` command must be run on the CVM master.

Note: You cannot access Database FlashSnap commands (`dbed_vmchecksnap`, `dbed_vmsnap`, and `dbed_vmclonedb`) with the SFDB menu utility.

Before creating a `snapplan`, make sure the following conditions have been met:

Prerequisites	■ Storage must be configured as specified: See “Preparing hosts and storage for Database FlashSnap” on page 162. ■ You must be the Oracle database administrator. ■ The disk group must be version 110 or later. For more information on disk group versions, see the <code>vxvg(1M)</code> manual page. ■ Be sure that a DCO and DCO volume are associated with the volume for which you are creating the snapshot. ■ Snapshot plexes and their associated DCO logs should be on different disks than the original plexes, and should be configured correctly for creating snapshots by the system administrator. ■ Persistent FastResync must be enabled on the existing database volumes and disks must be assigned for the snapshot volumes. ■ The database must be running in archive log mode. Archive log mode is set in the Oracle initialization parameter file. ■ The Oracle database must have at least one mandatory archive destination. ■ ORACLE_HOME cannot reside on disk which will be used for snapshot. ■ The Oracle database files and archive log files should use different volumes with unique disks in same disk group.
Usage Notes	■ The snapplan must be created on the primary host. ■ After creating the snapplan using the <code>dbed_vmchecksnap</code> command, you can use a text editor to review and update the file, if necessary. ■ It is recommended that you create a local working directory to store your snapplans in. ■ See the <code>dbed_vmchecksnap(1M)</code> online manual page for more information. ■ If the <code>SNAPSHOT_MODE</code> for the database is set to online, the primary and secondary hosts can be the same.

Note: You must issue commands as an Oracle database administrator in the following procedure.

To create a snapplan

- 1 Change directories to the working directory you want to store your snapplan in.

```
$ cd /working_directory
```

- 2 Create a snapplan with default values using the `dbed_vmchecksnap` command:

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S ORACLE_SID \  
-H ORACLE_HOME -f SNAPPLAN -o setdefaults -t host_name \  
[-p PLEX_TAG]
```

- 3 Open the snapplan file in a text editor and modify it as needed.

Example snapplans created for a snapshot image

In this example, a snapplan, snap1, is created for a snapshot image in a same-node configuration and default values are set. The host is named host1 and the working directory is `/export/snap_dir`.

The following is an example of the `dbed_vmchecksnap` command and sample output:

```
$ cd /export/snap_dir

$ /opt/VRTS/bin/dbed_vmchecksnap -S PROD \  
-H /oracle/product/orahome -f snap1 -o setdefaults -t host1
Snapplan snap1 for PROD.
=====
SNAPSHOT_VERSION=5.0
PRIMARY_HOST=host1
SECONDARY_HOST=host1
PRIMARY_DG=PRODDg
SNAPSHOT_DG=SNAP_PRODDg
ORACLE_SID=PROD
ARCHIVELOG_DEST=/prod_ar
SNAPSHOT_ARCHIVE_LOG=yes
SNAPSHOT_MODE=online
SNAPSHOT_PLAN_FOR=database
SNAPSHOT_PLEX_TAG=dbed_flashsnap
SNAPSHOT_VOL_PREFIX=SNAP_
ALLOW_REVERSE_RESYNC=no
SNAPSHOT_MIRROR=1
```

In this second example, a snapplan, snap2, is created for a snapshot image in a two-node in the cluster configuration, and default values are set. The primary host is host1, the secondary host is host2, and the working directory is /export/snap_dir.

The following is an example of the `dbed_vmchecksnap` command and sample output:

```
$cd /export/snap_dir

$/opt/VRTS/bin/dbed_vmchecksnap -S PROD \
-H /oracle/product/orahome -f snap2 -o setdefaults -t host2
Snapplan snap2 for PROD.
=====
SNAPSHOT_VERSION=5.0
PRIMARY_HOST=host1
SECONDARY_HOST=host2
PRIMARY_DG=PRODDg
SNAPSHOT_DG=SNAP_PRODDg
ORACLE_SID=PROD
ARCHIVELOG_DEST=/mytest/arch
SNAPSHOT_ARCHIVE_LOG=yes
SNAPSHOT_MODE=online
SNAPSHOT_PLAN_FOR=database
SNAPSHOT_PLEX_TAG=dbed_flashsnap
SNAPSHOT_VOL_PREFIX=SNAP_
ALLOW_REVERSE_RESYNC=no
SNAPSHOT_MIRROR=1
```

By default, a snapplan's `SNAPSHOT_PLEX_TAG` value is set as `dbed_flashsnap`. You can use the `-p` option to assign a different tag name. Make use of the `-p` option when creating the snapplan with the `setdefaults` option.

In the following example, the `-p` option is used with `setdefaults` to assign `my_tag` as the `SNAPSHOT_PLEX_TAG` value.

```
$ dbed_vmchecksnap -S PROD -H $ORACLE_HOME -O setdefaults \
-p my_tag -f snap1 -t host2
Snapplan snap1 for PROD
=====
SNAPSHOT_VERSION=5.0
PRIMARY_HOST=host1
SECONDARY_HOST=host2
PRIMARY_DG=PRODDg
SNAPSHOT_DG=SNAP_PRODDg
```

```
ORACLE_SID=PROD
ARCHIVELOG_DEST=/arch_data
SNAPSHOT_ARCHIVE_LOG=yes
SNAPSHOT_MODE=online
SNAPSHOT_PLAN_FOR=database
SNAPSHOT_PLEX_TAG=my_tag
SNAPSHOT_VOL_PREFIX=SNAP_
ALLOW_REVERSE_RESYNC=no
SNAPSHOT_MIRROR=1
```

Creating multi-mirror snapshots

To make the Database Snapshots highly available, the snapped snapshot volume should contain more than one mirror. This makes the snapshot volumes available even if one of the mirrors becomes disabled. Snapshot volumes can be mounted and the entire database snapshot is usable even if one of the mirrors becomes disabled. The multi-mirror snapshots are enabled by `SNAPSHOT_MIRROR=<n>` in the snapplan.

Note: There are no changes to the Command Line usage or arguments for the Flashsnap tools.

Before taking the snapshot, make sure all tagged snapshot mirrors are in SNAPDONE state.

For information about snapshot mirrors, refer to the *Veritas Volume Manager Administrator's Guide*.

Validating a snapplan (dbed_vmchecksnap)

After creating a snapplan, the next steps are to validate the snapplan parameters and check whether the snapshot volumes have been configured correctly for creating snapshots. If validation is successful, the snapplan is copied to the repository. The snapplan is validated using the `dbed_vmchecksnap` command with the `-o validate` option.

Consider the following prerequisites and notes before validating a snapplan:

Prerequisites	■ The database must be up and running while executing the <code>dbed_vmchecksnap</code> command.

Usage Notes

- The `dbed_vmchecksnap` command must be run as the Oracle database administrator.
- After validating the snapplan, you have the option of modifying the snapplan file to meet your storage configuration requirements.
- When using `dbed_vmchecksnap -o validate` to validate the snapplan and storage, you can save the validation output. The system administrator can use this information to adjust the storage setup if the validation fails.
- If a snapplan is updated or modified, you must re-validate it. It is recommended that snapplans are revalidated when changes are made in the database disk group.
- The `dbed_vmchecksnap` command can be used on the primary or secondary host.
- See the `dbed_vmchecksnap(1M)` manual page for more information.

If you modify the default snapplan, use the virtual host (node) name defined for the resource group for the PRIMARY_HOST and/or SECONDARY_HOST, and run validation.

Note: You must issue commands as an Oracle database administrator in the following procedure.

To validate a snapplan

- 1 Change directories to the working directory your snapplan is stored in:

```
$ cd /working_directory
```

- 2 Validate the snapplan using the `dbed_vmchecksnap` command:

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S ORACLE_SID \  
-H ORACLE_HOME -f SNAPPLAN -o validate
```

Example to validate a snapplan snap1 for a snapshot image

In the following example, a snapplan, snap1, is validated for a snapshot image in a same-node configuration. The primary host is host1 and the working directory is `/export/snap_dir`.

Note: You must issue commands as an Oracle database administrator in the following procedure.

The following is an example of the `dbed_vmchecksnap` command and sample output:

```
$ cd /export/snap_dir
$ /opt/VRTS/bin/dbed_vmchecksnap -S PROD -H /oracle/product/orahome \
-f snap1 -o validate
```

```
PRIMARY_HOST is host1
SECONDARY_HOST is host1
The version of PRIMARY_DG-PRODDg is 110.
The primary diskgroup PRODDg is a shared disk group
SNAPSHOT_DG is SNAP_PRODDg
```

```
SNAPSHOT_MODE is online
```

```
The database is running in archivelog mode.
```

```
ARCHIVELOG_DEST is /prod_ar
SNAPSHOT_PLAN_FOR is database
SNAPSHOT_ARCHIVE_LOG is yes
ARCHIVELOG_DEST=/prod_ar is mount on /dev/vx/dsk/PRODDg/prod_ar.
```

```
Examining Oracle volume and disk layout for snapshot
```

```
Volume prod_db on PRODDg is ready for snapshot.
Original plex and DCO log for prod_db is on PRODDg01.
Snapshot plex and DCO log for prod_db is on PRODDg02.
SNAP_PRODDg for snapshot will include: PRODDg02
ALLOW_REVERSE_RESYNC is no
```

```
The snapplan snap1 has been created.
```

In the following example, a snapplan, `snap2`, is validated for a snapshot image in a two node in a cluster configuration. The primary host is `host1`, the secondary host is `host2`, and the working directory is `/export/snap_dir`.

The following is an example of the `dbed_vmchecksnap` command and sample output:

```
$ cd /export/snap_dir
$ /opt/VRTS/bin/dbed_vmchecksnap -S PROD -H \
/oracle/product/orahome -f snap2 -o validate
```

```
PRIMARY_HOST is host1
```

```
SECONDARY_HOST is host2
The version of PRIMARY_DG-PRODDg is 110.
The primary diskgroup PRODDg is a shared disk group
SNAPSHOT_DG is SNAP_PRODDg
SNAPSHOT_MODE is online
```

The database is running in archivelog mode.

```
ARCHIVELOG_DEST is /mytest/arch
SNAPSHOT_PLAN_FOR is database
SNAPSHOT_ARCHIVE_LOG is yes
ARCHIVELOG_DEST=/mytest/arch is mount on
/dev/vx/dsk/PRODDg/arch.
```

```
Examining Oracle volume and disk layout for snapshot.
Volume arch on PRODDg is ready for snapshot.
Original plex and DCO log for arch is on PRODDg01.
Snapshot plex and DCO log for arch is on PRODDg02.
```

```
Volume prod_db on PRODDg is ready for snapshot.
Original plex and DCO log for prod_db is on PRODDg01.
Snapshot plex and DCO log for prod_db is on PRODDg04.
```

SNAP_PRODDg for snapshot will include: PRODDg02

ALLOW_REVERSE_RESYNC is no

The snapplan snap2 has been created.

Displaying, copying, and removing a snapplan (dbed_vmchecksnap)

Consider the following usage notes before listing all snapplans for a specific Oracle database, displaying a snapplan file, or copying and removing snapplans.

Usage Notes	■ If the local snapplan is updated or modified, you must revalidate it. ■ If the database schema or disk group is modified, you must revalidate it after running <code>dbed_update</code>.

Displaying a snapplan

You can use the `dbed_vmchecksnap` command to list all available snapplans, and then use the `dbed_vmchecksnap` command to display detailed information for a particular snapplan.

To list all available snapplans for a specific Oracle database

- ◆ Use the `dbed_vmchecksnap` command as follows:

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S ORACLE_SID -o list
```

In the following example, all available snapplans are listed for the database PROD.

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S PROD -o list
```

The following snapplan(s) are available for PROD:

SNAP_PLAN	SNAP_STATUS
DB_STATUS	SNAP_READY
snap1	init_full
init	yes
snap2	init_full
init	yes

The command output displays all available snapplans, their snapshot status (SNAP_STATUS), database status (DB_STATUS), and whether a snapshot may be taken (SNAP_READY).

For Database FlashSnap status information, see the *Veritas Storage Foundation for Oracle Administrator's Guide*.

To display detailed information for a snapplan

- ◆ Use the `dbed_vmchecksnap` command as follows:

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S \  
ORACLE_SID -f SNAPPLAN -o list
```

In the following example, the snapplan `snap1` is displayed.

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S PROD -f snap1 -o list  
SNAPSHOT_VERSION=5.0  
PRIMARY_HOST=host1  
SECONDARY_HOST=host1  
PRIMARY_DG=PRODdg  
SNAPSHOT_DG=SNAP_PRODdg  
ORACLE_SID=PROD  
ARCHIVELOG_DEST=/prod_ar  
SNAPSHOT_ARCHIVE_LOG=yes  
SNAPSHOT_MODE=online  
SNAPSHOT_PLAN_FOR=database  
SNAPSHOT_PLEX_TAG=dbed_flashsnap  
SNAPSHOT_VOL_PREFIX=SNAP_  
ALLOW_REVERSE_RESYNC=no  
SNAPSHOT_MIRROR=1  
STORAGE_INFOPRODdg02  
SNAP_PLEX=prod_ar-02  
  
STATUS_INFO  
SNAP_STATUS=init_full  
DB_STATUS=init
```

Copying a snapplan

If you want to create a snapplan similar to an existing snapplan, you can simply create a copy of the existing snapplan and modify it. To copy a snapplan from the SFDB repository to your current directory, the snapplan must not already be present in the current directory.

To copy a snapplan from the SFDB repository to your current directory

- ◆ Use the `dbed_vmchecksnap` command as follows:

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S ORACLE_SID \  
-f SNAPPLAN -o copy
```

In the following example, the snapplan, snap1, is copied from the VxDBA repository to the current directory.

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S PROD -f snap1 -o copy  
Copying 'snap1' to '/export/snap_dir'
```

Removing a snapplan

A snapplan can be removed from a local directory or repository if the snapplan is no longer needed.

To remove a snapplan from the SFDB repository

- ◆ Use the `dbed_vmchecksnap` command as follows:

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S ORACLE_SID -f\  
SNAPPLAN -o remove
```

In the following example, the snapplan, snap1, is removed from the SFDB repository.

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S PROD -f snap1 -o remove  
The snapplan snap1 has been removed.
```

Creating a snapshot (dbed_vmsnap)

The `dbed_vmsnap` command creates a snapshot of an Oracle database by splitting the mirror volumes used by the database into a snapshot database. You can use the snapshot image on either the same host as the database or on a secondary host provided storage is shared by the two hosts.

The snapshot image created by `dbed_vmsnap` is a frozen image of an Oracle database's datafiles. The `dbed_vmsnap` command ensures that a backup control file is created when the snapshot database is created, which allows for complete data recovery, if needed.

For Database FlashSnap status information, see the *Veritas Storage Foundation for Oracle Administrator's Guide*.

Note: You cannot access Database FlashSnap commands (`dbed_vmchecksnap`, `dbed_vmsnap`, and `dbed_vmclonedb`) with the SFDB menu utility.

Prerequisites	■ You must be logged in as the Oracle database administrator. ■ You must create and validate a snapplan using <code>dbed_vmchecksnap</code> before you can create a snapshot image with <code>dbed_vmsnap</code>.
Usage Notes	■ The <code>dbed_vmsnap</code> command can only be used on the primary host. ■ Do not share volumes between Oracle database files and other software. ■ When creating a snapshot volume, create the snapshot on a separate controller and on separate disks from the primary volume. ■ Make sure your archive log destination is separate from your Oracle database volumes. ■ Do not place any datafiles, including control files, in the <code>\$ORACLE_HOME/dbs</code> directory. ■ Resynchronization speed varies based on the amount of data changed in both the primary and secondary volumes when the mirror is broken off. ■ See the <code>dbed_vmsnap (1M)</code> manual page for more information.

Note the following points:

- To force snapshot creation, use the `-F` option. The `-F` option can be used after a snapshot operation has failed and the problem was fixed without using Veritas Storage Foundation commands. (That is, the volumes were synchronized without using Veritas Storage Foundation commands.) In this situation, the status of the snapplan will appear as unavailable for creating a snapshot. The `-F` option ignores the unavailable status, checks for the availability of volumes, and creates the snapshot after the volumes pass the availability check.
- After the snapshot is created, `dbed_vmsnap` returns values you will need to run `dbed_vmclonedb`. These values include the snapshot disk group, the snapplan name, and the SFDB repository volume for a two node in a cluster configuration. Make a note of these values so you have them when running `dbed_vmclonedb`.
- You can also use the command `dbed_vmchecksnap -f snapplan -o list` to access the information regarding the snapshot disk group, the snapplan name, and the SFDB repository.

Note: You must issue commands as an Oracle database administrator in the following procedure.

To create a snapshot

- 1 Change directories to the working directory in which your snapplan is stored:

```
$ cd /working_directory
```

- 2 If SNAPSHOT_MODE is set to offline in the snapplan, shut down the database.
- 3 Create the snapshot image using the `dbed_vmsnap` command.

```
$ /opt/VRTS/bin/dbed_vmsnap -S ORACLE_SID -f SNAPPLAN \  
-o snapshot [-F]
```

The snapshot volumes now represent a consistent backup copy of the database. You can backup the database by copying the snapshot volumes to tape or other backup media.

See [“Backing up the database from snapshot volumes \(dbed_vmclonedb\)”](#) on page 197.

- 4 You can also create another Oracle database for decision-support purposes. See [“Cloning a database \(dbed_vmclonedb\)”](#) on page 201.

Example to create a snapshot image of the database PROD

In this example, a snapshot image of the database, PROD, is created for a same-node configuration. In this case, the `SECONDARY_HOST` parameter is set the same as the `PRIMARY_HOST` parameter in the snapplan.

Note: You must issue commands as an Oracle database administrator in the following procedure.

```
$ /opt/VRTS/bin/dbed_vmsnap -S PROD -f snap1 -o snapshot
```

```
dbed_vmsnap started at 2006-03-02 14:15:27  
VxDBA repository is up to date.  
The database is running in archivelog mode.  
A snapshot of ORACLE_SID PROD is in DG SNAP_PRODdg.  
Snapplan snap1 is used for the snapshot.
```

If `-r <relocate_path>` is used in `dbed_vmclonedb`, make sure

<relocate_path> is created and owned by Oracle DBA. Otherwise, the following mount points need to be created and owned by Oracle DBA:

```
/prod_db.  
/prod_ar.
```

dbed_vmsnap ended at 2006-03-02 14:16:11

In this example, a snapshot image of the primary database, PROD, is created for a two node in a cluster configuration. In this case, the SECONDARY_HOST parameter specifies a different host name than the PRIMARY_HOST parameter in the snapplan.

```
$ /opt/VRTS/bin/dbed_vmsnap -S PROD -f snap2 -o snapshot
```

```
dbed_vmsnap started at 2005-03-02 23:01:10  
VxDBA repository is up to date.  
The database is running in archivelog mode.  
A snapshot of ORACLE_SID PROD is in DG SNAP_PRODdg.  
Snapplan snap2 is used for the snapshot.  
VxDBA repository volume is SNAP_arch.
```

If -r <relocate_path> is used in dbed_vmclonedb, make sure <relocate_path> is created and owned by Oracle DBA. Otherwise, the following mount points need to be created and owned by

Oracle DBA:

```
/prod_db.  
/prod_ar.
```

dbed_vmsnap ended at 2005-03-02 23:02:58

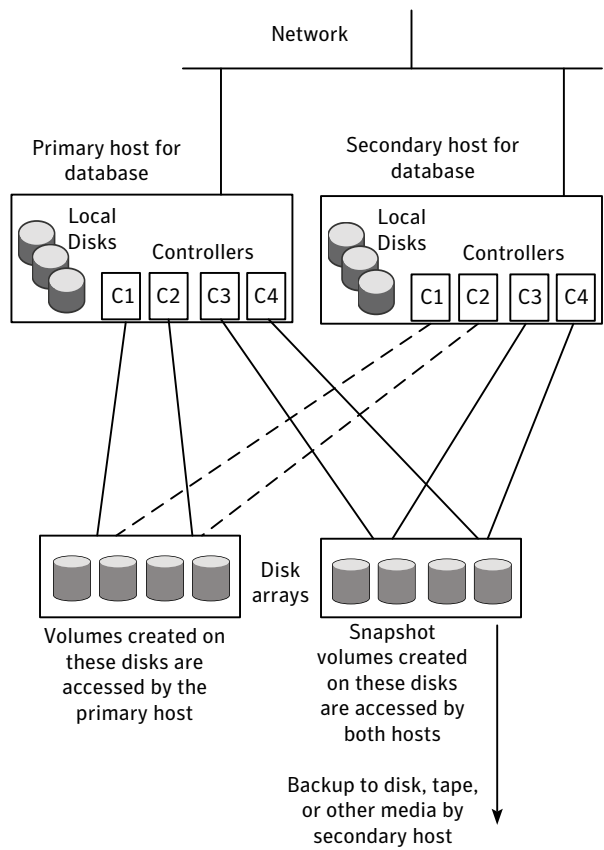
Backing up the database from snapshot volumes (dbed_vmclonedb)

Snapshots are most commonly used as a source for backing up a database. The advantage of using snapshot volumes is that the backup will not contest the I/O bandwidth of the physical devices. Making the snapshot volumes available on a secondary host will eliminate the extra loads put on processors and I/O adapters by the backup process on the primary host.

A clone database can also serve as a valid backup of the primary database. You can back up the primary database to tape using snapshot volumes.

Figure 6-5 shows a typical configuration when snapshot volumes are used on a secondary host.

Figure 6-5 Example system configuration for database backup on a secondary host



Prerequisites	■ You must be logged in as the Oracle database administrator to use <code>dbed_vmclonedb</code> command. ■ Before you can use the <code>dbed_vmclonedb</code> command, you must validate a snapplan and create a snapshot. See “About creating database snapshots” on page 177. See “Validating a snapplan (<code>dbed_vmchecksnap</code>)” on page 188. See “Creating a snapshot (<code>dbed_vmsnap</code>)” on page 194. ■ The volume snapshot must contain the entire database. ■ Before you can use the <code>dbed_vmclonedb</code> command with the <code>-r relocate_path</code> option (which specifies the initial mount point for the snapshot image), the system administrator must create the mount point and then change the owner to the Oracle database administrator.
Usage Notes	■ The <code>dbed_vmclonedb</code> command can be used on the secondary host. ■ In a same-node configuration, the primary and secondary hosts are the same. ■ In a same-node configuration, <code>-r relocate_path</code> is required. ■ In a two node in a cluster configuration, the <code>SFDBvol=vol_name</code> option is required. ■ See the <code>dbed_vmclonedb(1M)</code> manual page for more information.

Note: You cannot access Database FlashSnap commands (`dbed_vmchecksnap`, `dbed_vmsnap`, and `dbed_vmclonedb`) with the SFDB menu utility.

Mounting the snapshot volumes and backing up

Before using the snapshot volumes to do a backup, you must first mount them.

Note: You must issue commands as an Oracle database administrator in the following procedure.

Note: If you use the Oracle online backup method, you must also back up all the archived log files in order to do a complete restore and recovery of the database.

To mount the snapshot volumes

- ◆ Use the `dbed_vmclonedb` command as follows:

```
$ /opt/VRTS/bin/dbed_vmclonedb -S ORACLE_SID -g snap_dg \  
-o mount,new_sid=new_sid,server_name=svr_name -f SNAPPLAN [-H ORACLE_HOME] \  
[-r relocate_path]
```

You can now back up an individual file or a group of files under a directory onto the backup media.

In this example, snapshot volumes are mounted.

```
$ /opt/VRTS/bin/dbed_vmclonedb -S PROD -g SNAP_PRODDg \  
-o mount,new_sid=NEWPROD,server_name=svr_name -f snap1 -r /clone/single
```

```
dbed_vmclonedb started at 2004-04-02 15:35:41  
Mounting /clone/single/prod_db on  
/dev/vx/dsk/SNAP_PRODDg/SNAP_prod_db.  
Mounting /clone/single/prod_ar on  
/dev/vx/dsk/SNAP_PRODDg/SNAP_prod_ar.  
dbed_vmclonedb ended at 2004-04-02 15:35:50
```

The following is an example of creating a snapshot on the same host (host nobody):

```
$ /opt/VRTS/bin/dbed_vmclonedb -S PROD -g SNAP_PRODDg \  
-o mount,new_sid=NEWPROD,server_name=nobody -f snap1 -r /clone/single
```

```
dbed_vmclonedb started at 2006-10-24 10:44:54  
Mounting /clone/single/archivelogs on /dev/vx/dsk/SNAP_PRODDg/SNAP_archvol.  
Mounting /clone/single/oradata on /dev/vx/dsk/SNAP_PRODDg/  
SNAP_ora_data_vol.  
dbed_vmclonedb ended at 2006-10-24 10:45:49
```

Note: A usage error is displayed if the `server_name` is not given in the above command.

To mount a Storage Checkpoint carried over from the snapshot volumes to a secondary host

- 1 On the secondary host, list the Storage Checkpoints carried over from the primary database using the `dbed_ckptdisplay` command.

For example:

```
$ /opt/VRTS/bin/dbed_ckptdisplay -S ORACLE_SID -n
```

- 2 You can mount one of the listed Storage Checkpoints using the `dbed_ckptmount` command.

For example:

```
$ /opt/VRTS/bin/dbed_ckptmount -S ORACLE_SID -c CKPT_NAME \  
-m MOUNT_POINT
```

Note the following limitations:

- Any mounted Storage Checkpoints must be unmounted before running the following commands:

```
$ /opt/VRTS/bin/dbed_vmclonedb -o umount,new_sid=new_sid,server_name=svr_name \  
-f SNAPPLAN
```

- It is only possible to mount a Storage Checkpoint carried over with the snapshot volumes in a two node in a cluster configuration if the snapshot volumes were mounted with the `dbed_vmclonedb` command with the `-o mount` option without the use of `-r relocate_path`.
- Storage Checkpoints carried over with the snapshot volumes can be mounted before a clone database is created using `dbed_vmclonedb` with the `-o mount` option. After a clone database is created using `dbed_vmclonedb` with the `-o recoverdb` option, however, Storage Checkpoints are no longer present.

To back up the database using the snapshot

- ◆ Copy the snapshot volumes to tape or other appropriate backup media.

Cloning a database (dbed_vmclonedb)

Veritas Storage Foundation lets you create a clone database using snapshot volumes. You can use snapshots of a primary database to create a clone of the

database at a given point in time. You can then implement decision-support analysis and report generation operations that take their data from the database clone rather than from the primary database to avoid introducing additional burdens on the production database.

A clone database can also serve as a valid backup of the primary database.

See [“Backing up the database from snapshot volumes \(dbed_vmclonedb\)”](#) on page 197.

You can also back up the primary database to tape using snapshot volumes.

The resynchronization functionality of Database FlashSnap allows you to quickly refresh the clone database with up-to-date information from the primary database. Reducing the time taken to update decision-support data also lets you generate analysis reports more frequently.

Using Database FlashSnap to clone a database

In a same-node configuration, the `dbed_vmclonedb` command creates a clone database on the same host. The command can also be used to shut down the clone database and unmount its file systems. When creating or unmounting the clone database in a same-node configuration, `-r relocate_path` is required so that the clone database’s file systems use different mount points than those used by the primary database.

When used in a two node in a cluster configuration, the `dbed_vmclonedb` command imports the snapshot disk group `SNAP_dg`, mounts the file systems on the snapshot volumes, and starts a clone database. It can also reverse the process by shutting down the clone database, unmounting the file systems, and deporting the snapshot disk group. When creating the clone off host, `-o SFDBvol=vol_name` is required.

Warning: When creating a clone database, all Storage Checkpoints in the original database are discarded.

Prerequisites	■ You must be logged in as the Oracle database administrator. ■ Before you can use the <code>dbed_vmclonedb</code> command, you must validate a snapplan and create a snapshot. See “About creating database snapshots” on page 177. See “Validating a snapplan (dbed_vmchecksnap)” on page 188. See “Creating a snapshot (dbed_vmsnap)” on page 194. ■ The volume snapshot must contain the entire database. ■ The system administrator must provide the database administrator with access to the necessary volumes and mount points. ■ Before you can use the <code>dbed_vmclonedb</code> command with the <code>-r relocate_path</code> option (which specifies the initial mount point for the snapshot image), the system administrator must create the mount point and then change the owner to the Oracle database administrator. ■ The Oracle database must have at least one mandatory archive destination.
Usage Notes	■ The <code>dbed_vmclonedb</code> command can be used on the secondary host. ■ In a same-node configuration, <code>-r relocate_path</code> is required. This command is also needed if the name of the clone database is different than the primary database. ■ In a two node in a cluster configuration, the <code>SFDBvol=vol_name</code> option is required. ■ The initialization parameters for the clone database are copied from the primary database. This means that the clone database takes up the same memory and machine resources as the primary database. If you want to reduce the memory requirements for the clone database, shut down the clone database and then start it up again using a different <code>init.ora</code> file that has reduced memory requirements. If the host where <code>dbed_vmclonedb</code> is run has little available memory, you may not be able to start up the clone database and the cloning operation may fail. ■ See the <code>dbed_vmclonedb(1M)</code> manual page for more information.

Note: You must issue commands as an Oracle database administrator in the following procedure.

To mount a database and recover it manually

- 1 Start and mount the clone database to allow manual database recovery:

```
$ /opt/VRTS/bin/dbed_vmclonedb -S ORACLE_SID -g snap_dg \
  -o mountdb,new_sid=new_sid,server_name=svr_name -f SNAPPLAN \
  [-H ORACLE_HOME] [-r relocate_path]
```

- 2 Recover the database manually.
- 3 Update the snapshot status information for the clone database in the SFDB repository:

```
$ /opt/VRTS/bin/dbed_vmclonedb -o update_status,\
  new_sid=new_sid,server_name=svr_name -f SNAPPLAN [-r relocate_path]
```

Example: Mounting the file systems without bringing up the clone database

In this example, file systems are mounted without bringing up the clone database. The clone database must be manually created and recovered before it can be used. This example is for a clone created on the same host as the primary database.

Note: You must issue commands as an Oracle database administrator in the following procedure.

```
$ /opt/VRTS/bin/dbed_vmclonedb -S PROD -g SNAP_PRODDg \
  -o mountdb,new_sid=NEWPROD,server_name=svr_name -f snap1 -r /clone
```

```
dbed_vmclonedb started at 2006-03-02 15:34:41
Mounting /clone/prod_db on /dev/vx/dsk/SNAP_PRODDg/SNAP_prod_db.
Mounting /clone/prod_ar on /dev/vx/dsk/SNAP_PRODDg/SNAP_prod_ar.
All redo-log files found.
Altering instance_name paramter in initabc.ora.
Altering instance_number paramter in initabc.ora.
Altering thread paramter in initabc.ora.Starting automatic database recovery.
Database NEWPROD (SID=NEWPROD) is in recovery mode.
If the database NEWPROD is recovered manually, you must run
dbed_vmclonedb -o update_status to change the snapshot status.
dbed_vmclonedb ended at 2006-03-02 15:34:59
```

The database status (database_recovered) needs to be updated for a clone database on the primary host after manual recovery has been completed.

```
$ /opt/VRTS/bin/dbed_vmclonedb -o update_status,\
  new_sid=NEWPROD,server_name=svr_name -f snap1 -r /clone
```

```
dbed_vmclonedb started at 2006-03-02 15:35:16
The snapshot status has been updated.
dbed_vmclonedb ended at 2006-03-02 15:35:42
```

Example: Mounting the file systems without recovering the clone database

Note: You must issue commands as an Oracle database administrator in the following procedure.

In this example, file systems are mounted without recovering the clone database. The clone database must be manually recovered before it can be used. This example is for a clone created on a secondary host.

```
$ /opt/VRTS/bin/dbed_vmclonedb -S HOST2_SID -g SNAP_PRODDg \
  -o mountdb,new_sid=NEWPROD,server_name=host2 -f snap2
```

```
dbed_vmclonedb started at 2006-03-09 23:26:50
Mounting /clone/arch on /dev/vx/dsk/SNAP_PRODDg/SNAP_arch.
Mounting /clone/prod_db on /dev/vx/dsk/SNAP_PRODDg/SNAP_prod_db.
All redo-log files found.
Altering instance_name paramter in initabc.ora.
Altering instance_number paramter in initabc.ora.
Altering thread paramter in initabc.ora.
Starting automatic database recovery.
Database NEWPROD (SID=NEWPROD) is in recovery mode.
```

If the database NEWPROD is recovered manually, you must run `dbed_vmclonedb -o update_status` to change the snapshot status.

```
dbed_vmclonedb ended at 2006-03-09 23:27:17
```

The database is recovered manually.

The snapshot status (`database_recovered`) is updated for a clone database on a secondary host after manual recovery has been completed.

```
$ /opt/VRTS/bin/dbed_vmclonedb -o update_status,\
  new_sid=NEWPROD,server_name=host2 -f snap2
```

```
dbed_vmclonedb started at 2006-03-09 23:34:01
The snapshot status has been updated.
dbed_vmclonedb ended at 2006-03-09 23:34:35
```

To clone the database automatically

- ◆ Use the `dbed_vmclonedb` command as follows:

```
$ /opt/VRTS/bin/dbed_vmclonedb -S ORACLE_SID -g snap_dg \  
-o recoverdb,new_sid=new_sid,server_name=svr_name -f SNAPPLAN \  
[-H ORACLE_HOME] [-r relocate_path]
```

Where:

ORACLE_SID	Represents the name of the Oracle database used to create the snapshot.
snap_dg	Represents the name of the diskgroup that contains all the snapshot volumes.
new_sid	Specifies the ORACLE_SID for the clone database.
server_name	Specifies the server name as svr_name.
SNAPPLAN	Represents the name of the snapplan file.
ORACLE_HOME	Represents the ORACLE_HOME setting for the ORACLE_SID database.
relocate_path	Represents the name of the initial mount point for the snapshot image.

When cloning a database on a secondary host, ensure that PRIMARY_HOST and SECONDARY_HOST parameters in the snapplan file are different.

When the `-o recoverdb` option is used with `dbed_vmclonedb`, the clone database is recovered automatically using all available archive logs. If the `-o recoverdb` option is not used, you can perform point-in-time recovery manually.

In the following example, a clone of the primary database is automatically created on the same host as the primary database.

```
$ /opt/VRTS/bin/dbed_vmclonedb -S PROD -g SNAP_PRODDg \  
-o recoverdb,new_sid=NEWPROD,server_name=svr_name -f snap1 -r /clone
```

```
dbed_vmclonedb started at 2006-03-02 14:42:10  
Mounting /clone/prod_db on /dev/vx/dsk/SNAP_PRODDg/SNAP_prod_db.  
Mounting /clone/prod_ar on /dev/vx/dsk/SNAP_PRODDg/SNAP_prod_ar.  
All redo-log files found.  
Altering instance_name paramter in initabc.ora.  
Altering instance_number paramter in initabc.ora.
```

```
Altering thread paramter in initabc.ora.  
Starting automatic database recovery.  
Database NEWPROD (SID=NEWPROD) is running.  
dbed_vmclonedb ended at 2006-03-02 14:43:05
```

In the following example, a clone of the primary database is automatically created on a secondary host.

```
$ /opt/VRTS/bin/dbed_vmclonedb -S PROD -g SNAP_PRODDg \  
-o recoverdb,new_sid=NEWPROD,server_name=svr_name -f snap2  
  
dbed_vmclonedb started at 2006-03-09 23:03:40  
Mounting /clone/arch on /dev/vx/dsk/SNAP_PRODDg/SNAP_arch.  
Mounting /clone/prod_db on /dev/vx/dsk/SNAP_PRODDg/SNAP_prod_db.  
All redo-log files found.  
Altering instance_name paramter in initabc.ora.  
Altering instance_number paramter in initabc.ora.  
Altering thread paramter in initabc.ora.  
Starting automatic database recovery.  
Database NEWPROD (SID=NEWPROD) is running.  
dbed_vmclonedb ended at 2006-03-09 23:04:50
```

Shutting down the clone database and unmounting file systems

When you are done using the clone database, you can shut it down and unmount all snapshot file systems with the `dbed_vmclonedb -o umount` command. If the clone database is used on a secondary host that has shared disks with the primary host, the `-o umount` option also deports the snapshot disk group.

Note: Any mounted Storage Checkpoints mounted need to be unmounted before running `dbed_vmclonedb -o umount`.

To shut down the clone database and unmount all snapshot file systems

- ◆ Use the `dbed_vmclonedb` command as follows:

```
$ /opt/VRTS/bin/dbed_vmclonedb -o umount,new_sid=new_sid,server_name=svr_name \  
-f SNAPPLAN [-r relocate_path]
```

In this example, the clone database is shut down and file systems are unmounted for a clone on the same host as the primary database (a same-node configuration).

```
$ /opt/VRTS/bin/dbed_vmclonedb -o umount,new_sid=NEWPROD,server_name=svr_name \  
-f snap1 -r /clone
```

```
dbed_vmclonedb started at 2006-03-02 15:11:22
```

```
dbed_vmclonedb ended at 2006-03-02 15:11:47
```

In this example, the clone database is shut down, file systems are unmounted, and the snapshot disk group is deported for a clone on a secondary host (two node in a cluster configuration).

```
$ /opt/VRTS/bin/dbed_vmclonedb -o umount,new_sid=NEWPROD,server_name=svr_name \  
-f snap2
```

```
dbed_vmclonedb started at 2006-03-09 23:09:21
```

```
dbed_vmclonedb ended at 2006-03-09 23:09:50
```

Restarting a Clone Database

If the clone database is down as a result of using `dbed_vmclonedb -o umount` or rebooting the system, you can restart it with the `-o restartdb` option.

Note: This option can only be used when a clone database is created successfully. If the clone database is recovered manually, `-o update_status` must be run to update the status before `-o restartdb` will work.

To start the clone database

◆ Use the `dbed_vmclonedb` command as follows:

```
$ /opt/VRTS/bin/dbed_vmclonedb -S ORACLE_SID -g snap_dg \  
-o restartdb,new_sid=new_sid,server_name=svr_name -f SNAPPLAN [-H ORACLE_HOME] \  
[-r relocate_path]
```

In this example, the clone database is re-started on the same host as the primary database (same-node configuration).

```
$ /opt/VRTS/bin/dbed_vmclonedb -S PROD -g SNAP_PRODDg \  
-o restartdb,new_sid=NEWPROD,server_name=svr_name -f snap1 -r /clone
```

```
dbed_vmclonedb started at 2006-03-02 15:14:49  
Mounting /clone/prod_db on /dev/vx/dsk/SNAP_PRODDg/SNAP_prod_db.  
Mounting /clone/prod_ar on /dev/vx/dsk/SNAP_PRODDg/SNAP_prod_ar.  
Oracle instance NEWPROD successfully started.  
dbed_vmclonedb ended at 2006-03-02 15:15:19
```

In this example, the clone database is re-started on the secondary host (two node in a cluster configuration).

```
$ /opt/VRTS/bin/dbed_vmclonedb -S PROD -g SNAP_PRODDg \  
-o restartdb,new_sid=NEWPROD,server_name=svr_name -f snap2
```

```
dbed_vmclonedb started at 2006-03-09 23:03:40  
Mounting /clone/arch on /dev/vx/dsk/SNAP_PRODDg/SNAP_arch.  
Mounting /clone/prod_db on /dev/vx/dsk/SNAP_PRODDg/SNAP_prod_db.  
Oracle instance NEWPROD successfully started.  
dbed_vmclonedb ended at 2006-03-09 23:04:50
```

Recreating Oracle tempfiles

After a clone database is created and opened, the tempfiles are added if they were residing on the snapshot volumes. If the tempfiles were not residing on the same file systems as the datafiles, `dbed_vmsnap` does not include the underlying volumes in the snapshot. In this situation, `dbed_vmclonedb` issues a warning message and you can then recreate any needed tempfiles on the clone database as described in the following procedure.

To recreate the Oracle tempfiles

- 1 If the tempfiles were not residing on the same file systems as the datafiles, the `dbed_vmclonedb` command displays WARNING and INFO messages similar to the following:

```
WARNING: Not all tempfiles were included in snapshot for
$ORACLE_SID, there is no snapshot volume for
/clone_path/temp02.dbf.
WARNING: Could not recreate tempfiles for $ORACLE_SID due to
lack of free space.INFO: The sql script for adding tempfiles to $ORACLE_SID
```

where \$ORACLE_SID is the name of the clone database.

- 2 A script named `add_tf.$ORACLE_SID.sql` is provided in the `/tmp` directory for the purpose of recreating Oracle tempfiles. This script contains the SQL*Plus commands to recreate the missing tempfiles.
- 3 Make a copy of the `/tmp/add_tf.$ORACLE_SID.sql` script and open it to view the list of missing tempfiles.

An example of the `add_tf.$ORACLE_SID.sql` script is shown below:

```
$ cat /tmp/add_tf.$ORACLE_SID.sql
-- Commands to add tempfiles to temporary tablespaces.
-- Online tempfiles have complete space information.
-- Other tempfiles may require adjustment.
ALTER TABLESPACE TEMP ADD TEMPFILE
'/clone_path/temp01.dbf'
SIZE 4194304 REUSE AUTOEXTEND ON NEXT 1048576 MAXSIZE 33554432 ;
ALTER TABLESPACE TEMP ADD TEMPFILE
'/clone_path/temp02.dbf' REUSE;
ALTER DATABASE TEMPFILE '/clone_path2/temp02.dbf'
OFFLINE;
```

- 4 Evaluate whether you need to recreate any temp files. If you want to recreate tempfiles, proceed to the next step.
- 5 In the `add_tf.$ORACLE_SID.sql` file, edit the sizes and default path names of the tempfiles as needed to reside on cloned volumes configured for database storage.

Warning: Do not run the script without first editing it because path names may not exist and the specified mount points may not contain sufficient space.

- 6 After you have modified the `add_tf.$ORACLE_SID.sql` script, execute it against your clone database.
- 7 After you have successfully run the script, you may delete it.

Resynchronizing the snapshot to your database

When you have finished using a clone database or want to refresh it, you can resynchronize it with the original database. This is also known as refreshing the snapshot volume or merging the split snapshot image back to the current database image. After resynchronizing, the snapshot can be retaken for backup or decision-support purposes.

When resynchronizing the data in a volume:

- Resynchronize the snapshot from the original volume. This procedure is explained in this section.

You can resynchronize the snapshot from the original volume.

Prerequisites

- You must be logged in as the Oracle database administrator.
- Before you can resynchronize the snapshot image, you must validate a snapplan and create a snapshot.
 - See [“About creating database snapshots”](#) on page 177.
 - See [“Validating a snapplan \(dbed_vmchecksnap\)”](#) on page 188.
 - See [“Creating a snapshot \(dbed_vmsnap\)”](#) on page 194.
- If a clone database has been created, shut it down and unmount the file systems using the `dbed_vmclosedb -o umount` command. This command also departs the disk group if the primary and secondary hosts are different.
 - See [“Shutting down the clone database and unmounting file systems”](#) on page 207.
- The Oracle database must have at least one mandatory archive destination.

Usage Notes

- The `dbed_vmsnap` command can only be executed on the primary host.
- In a two node in a cluster configuration, the `dbed_vmsnap` command imports the disk group that was deported from the secondary host and joins the disk group back to the original disk group. The snapshot volumes again become plexes of the original volumes. The snapshot is then resynchronized.
- See the `dbed_vmsnap(1M)` manual page for more information.

Note: You must issue commands as an Oracle database administrator in the following procedure.

To resynchronize the snapshot image

- ◆ Use the `dbed_vmsnap` command as follows:

```
$ /opt/VRTS/bin/dbed_vmsnap -S ORACLE_SID -f SNAPPLAN -o resync
```

In this example, the snapshot image is resynchronized with the primary database.

```
$ /opt/VRTS/bin/dbed_vmsnap -S PROD -f snap1 -o resync
dbed_vmsnap started at 2006-03-02 16:19:05
The option resync has been completed.
dbed_vmsnap ended at 2006-03-02 16:19:26
```

Now, you can again start creating snapshots.

Removing a snapshot volume

If a snapshot volume is no longer needed, you can remove it and free up the disk space for other uses by using the `vxedit rm` command.

Prerequisites

- You must be logged in as superuser.
- If the volume is on a mounted file system, you must unmount it before removing the volume.

To remove a snapplan and snapshot volume

- 1 To remove the snapshot and free up the storage used by it:

- If the snapshot has been taken, remove the snapshot as follows:

```
# vxsnap -g diskgroup dis snapshot_volume

# vxvol -g diskgroup stop snapshot_volume

# vxedit -g diskgroup -rf rm snapshot_volume
```

- If the snapshot has not been taken and the snapshot plex (mirror) exists, remove the snapshot as follows:

```
# vxsnap -g diskgroup rmmir volume
```

2 Remove the DCO and DCO volume:

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

3 Remove the snapplan.

```
# /opt/VRTS/bin/dbed_vmchecksnap -S PROD -f snapplan -o remove
```

For example, the following commands will remove a snapshot volume from disk group PRODDg:

```
# vxsnap -g PRODDg dis snap_v1  
# vxvol -g PRODDg stop snap_v1  
# vxedit -g PRODDg -rf rm snap_v1
```


Using Database Dynamic Storage Tiering

This chapter includes the following topics:

- [About Database Dynamic Storage Tiering](#)
- [Dynamic Storage Tiering policy management](#)
- [Configuring Database Dynamic Storage Tiering](#)
- [Extent balancing in a database environment](#)
- [Running Database Dynamic Storage Tiering reports](#)
- [Oracle Database Dynamic Storage Tiering use cases](#)

About Database Dynamic Storage Tiering

Database Dynamic Storage Tiering (DST) matches data storage with data usage requirements. After data matching, the data can then be relocated based upon data usage and other requirements determined by the database administrator (DBA).

As more and more data is retained over a period of time, eventually, some of that data is needed less frequently. The data that is needed less frequently still requires a large amount of disk space. DST enables the database administrator to manage data so that less frequently used data can be moved to slower, less expensive disks. This also permits the frequently accessed data to be stored on faster disks for quicker retrieval.

Tiered storage is the assignment of different types of data to different storage types to improve performance and reduce costs. With DST, storage classes are

used to designate which disks make up a particular tier. There are two common ways of defining storage classes:

- **Performance, or storage, cost class:** The most-used class consists of fast, expensive disks. When data is no longer needed on a regular basis, the data can be moved to a different class that is made up of slower, less expensive disks.
- **Resilience class:** Each class consists of non-mirrored volumes, mirrored volumes, and n-way mirrored volumes.
 For example, a database is usually made up of data, an index, and logs. The data could be set up with a three-way mirror because data is critical. The index could be set up with a two-way mirror because the index is important, but can be recreated. The logs are not required on a daily basis and could be set up without mirroring.

Dynamic Storage Tiering policies control initial file location and the circumstances under which existing files are relocated. These policies cause the files to which they apply to be created and extended on specific subsets of a file systems's volume set, known as placement classes. The files are relocated to volumes in other placement classes when they meet specified naming, timing, access rate, and storage capacity-related conditions.

In addition to preset policies, you can manually move files to faster or slower storage with DST, when necessary. You can also run reports that list active policies, display file activity, display volume usage, or show file statistics.

Database Dynamic Storage Tiering building blocks

To use Database Dynamic Storage Tiering, your storage must be managed using the following features:

- VxFS multi-volume file system
- VxVM volume set
- Volume tags
- Dynamic Storage Tiering policies

About VxFS multi-volume file systems

Multi-volume file systems are file systems that occupy two or more virtual volumes. The collection of volumes is known as a volume set, and is made up of disks or disk array LUNs belonging to a single Veritas Volume Manager (VxVM) disk group. A multi-volume file system presents a single name space, making the existence of multiple volumes transparent to users and applications. Each volume retains a separate identity for administrative purposes, making it possible to control the

locations to which individual files are directed. This feature is available only on file systems meeting the following requirements:

- The minimum Diskgroup version is 140.
- The minimum filesystem version is 7.

To convert your existing VxFS file system to a VxFS multi-volume file system, you must convert a single volume to a volume set. See [“Converting a VxFS file system to a VxFS multi-volume file system”](#) on page 226.

The VxFS volume administration utility (fsvoladm utility) can be used to administer VxFS volumes. The fsvoladm utility performs administrative tasks, such as adding, removing, resizing, encapsulating volumes, and setting, clearing, or querying flags on volumes in a specified Veritas File System.

See the fsvoladm (1M) manual page for additional information about using this utility.

About VxVM volume sets

Volume sets allow several volumes to be represented by a single logical object. Volume sets cannot be empty. All I/O from and to the underlying volumes is directed via the I/O interfaces of the volume set. The volume set feature supports the multi-volume enhancement to Veritas File System (VxFS). This feature allows file systems to make best use of the different performance and availability characteristics of the underlying volumes. For example, file system metadata could be stored on volumes with higher redundancy, and user data on volumes with better performance.

About volume tags

You make a VxVM volume part of a placement class by associating a volume tag with it. For file placement purposes, VxFS treats all of the volumes in a placement class as equivalent, and balances space allocation across them. A volume may have more than one tag associated with it. If a volume has multiple tags, the volume belongs to multiple placement classes and is subject to allocation and relocation policies that relate to any of the placement classes.

Warning: Multiple tagging should be used carefully.

A placement class is a Dynamic Storage Tiering attribute of a given volume in a volume set of a multi-volume file system. This attribute is a character string, and is known as a volume tag.

About Dynamic Storage Tiering policies

Dynamic Storage Tiering allows administrators of multi-volume VxFS file systems to manage the placement of files on individual volumes in a volume set by defining placement policies that control both initial file location and the circumstances under which existing files are relocated. These placement policies cause the files to which they apply to be created and extended on specific subsets of a file system's volume set, known as placement classes. The files are relocated to volumes in other placement classes when they meet the specified naming, timing, access rate, and storage capacity-related conditions.

Database Dynamic Storage Tiering in a High Availability (HA) environment

Veritas Cluster Server does not provide a bundled agent for volume sets. If issues arise with volumes or volume sets, the issues can only be detected at the DiskGroup and Mount resource levels.

The DiskGroup agent brings online, takes offline, and monitors a Veritas Volume Manager (VxVM) disk group. This agent uses VxVM commands. When the value of the StartVolumes and StopVolumes attributes are both 1, the DiskGroup agent onlines and offlines the volumes during the import and deport operations of the disk group. When using volume sets, set StartVolumes and StopVolumes attributes of the DiskGroup resource that contains the volume set to 1. If a file system is created on the volume set, use a Mount resource to mount the volume set.

The Mount agent brings online, takes offline, and monitors a file system or NFS client mount point.

If you are using any of the Database Dynamic Storage Tiering commands in a high availability (HA) environment, the time on each system in the cluster must be synchronized. Otherwise, the scheduled task may not be executed at the expected time after a service group failover.

For additional information, see the *Veritas Cluster Server Bundled Agents Reference Guide*.

Dynamic Storage Tiering policy management

You can choose to manually relocate files or tablespaces, or you can use a preset Dynamic Storage Tiering (DST) policy.

Note: You must issue commands as an Oracle database administrator in the following procedures.

Relocating files

Table 7-1 shows the `dbdst_file_move` command options.

Table 7-1 `dbdst_file_move` command options

Command options	Description
<code>-o archive[n] flashback</code>	Specifies which archive logs or Flashback logs to move. Do not use this option with the <code>-f</code> option.
<code>-o external datafile</code>	Specifies whether to move external files or datafiles. Use this option with the <code>-f</code> option.
<code>-f listfile</code>	Specifies a listfile that contains a list of files or directories to be moved.
<code>-c class [:days]</code>	Specifies the storage class to which the files should be moved. If the days option is used, the files will be moved to the class specified if they have not been accessed in the number of days specified. Do not specify days if you are using the <code>-o datafile</code> option.
<code>-R</code>	Removes the policy for the specified object.

Before relocating a file, review the following information:

Usage notes

- Multiple partitions cannot reside on the same tablespace.

To relocate a file

- Use the `dbdst_file_move` command as follows:

```
$ /opt/VRTS/bin/dbdst_file_move -S $ORACLE_SID -o datafile \  
-f listfile -c storage_class:days [-c storage_class:days]
```

Relocating tablespaces

Use the `dbdst_tbs_move` command to move tablespaces to the desired storage class. The command queries the SFDB repository for the tablespace file names, then performs a one-time move based on your immediate requirements.

To relocate a tablespace

- ◆ Use the `dbdst_tbs_move` command as follows:

```
$ /opt/VRTS/bin/dbdst_tbs_move -S $ORACLE_SID -t tablespace \
-c class
```

where

- *tablespace* indicates which tablespace to move.
- *class* indicates to which class the tablespace should be moved.

Relocating table partitions

Use the `dbdst_partition_move` to move table partitions. The command queries the database to validate the names of the table and partition. From this information, a list of datafiles is derived and a one-time move of the files to the desired class is executed.

Before relocating table partitions, review the following information:

Prerequisites	The database must be up when you run the <code>dbdst_partition_move</code> command.

To relocate a table partition

- ◆ Use the `dbdst_partition_move` command as follows:

```
$ /opt/VRTS/bin/dbdst_partition_move -S $ORACLE_SID -T table_name \
-p partition_name -c class
```

where

- `-T` indicates the table name.
- `-p` indicates the partition name.
- `-c` indicates the class to which the table partition is to be moved.

For example, to move the `SALES_Q1` partition of the `SALES` table to storage class `SLOW`, use the `dbdst_partition_move` as follows:

```
$ /opt/VRTS/bin/dbdst_partition_move -S $ORACLE_SID -T SALES \
-p SALES_Q1 -c SLOW
```

Using preset policies

Use the `dbdst_preset_policy` command to set a policy based on file name patterns before the files are created.

Table 7-2 shows the preset policies command options.

Table 7-2 `dbdst_preset_policy` command options

Command option	Description
<code>-d directory</code>	Indicates the directory on which the placement policy will be applied.
<code>-e</code>	Enforces the file system of the specified directory. Use this option if there was an error in the previous enforcement that has been corrected and needs to be enforced again.
<code>-R</code>	Removes all pattern-based placement policies related to this directory.
<code>-l</code>	Lists the existing file placement that is set to the specified directory.
<code>-P pattern_spec</code>	Specifies file patterns and class assignment. This option will automatically place files in the desired class as soon as they are created. Existing files and newly created files will be moved immediately to the class specified.
<code>-f pattern file</code>	Specifies a file that contains a particular class and pattern. New files with this pattern will be placed in the class immediately. Existing files will be moved as well.
<code>-E</code>	Specifies that existing files should be moved to the designated class in a one-time move to be scheduled at a later time, such as the sweeptime specified in the <code>dbdst_admin</code> command.

To create a preset policy

- ◆ Use the `dbdst_preset_policy` command as follows:

```
$ /opt/VRTS/bin/dbdst_preset_policy -S $ORACLE_SID -d directory \  
-P pattern_spec
```

Configuring Database Dynamic Storage Tiering

To use database Dynamic Storage Tiering, the following requirements must be met:

- An Oracle database must be up and running.
- Only the Oracle database administrator can run Database Dynamic Storage Tiering commands.

To use Database Dynamic Storage Tiering, the following tasks must be performed:

- Review the Database Dynamic Storage Tiering command requirements.
- Define the database parameters.
- Set up storage classes.
- Convert an existing VxFS database file system to a VxFS multi-volume file system for use with Database Dynamic Storage Tiering.
- Classify, or tag, volumes so that the tags indicate the quality of the underlying disk.
- Display the free space on each class.
- Add or remove volumes as necessary.

Database Dynamic Storage Tiering command requirements

Before defining your database parameters, review the following command requirements:

- Run the `dbed_update` command before running any of the Database Dynamic Storage Tiering commands. You should also run the `dbed_update` command if any of the database files change.

The repository must be up to date, since the Database Dynamic Storage Tiering commands retrieve database information from the repository.

- Define the `LD_LIBRARY_PATH` environment variable as follows:
For a Solaris SPARC system:

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/VRTSdbms3/lib; \  
export LD_LIBRARY_PATH
```

For a Solaris x64 system:

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/VRTSdbms3/lib32; \  
export LD_LIBRARY_PATH
```

- If you are using any of the Database Dynamic Storage Tiering commands in a high availability (HA) environment, the time on each system in the cluster must be synchronized.
- Create the volumes that you want to add to the multi-volume file system in the same disk group as the file system volume. As root, use the following command to change the owner of each volume:

```
# /opt/VRTS/bin/vxedit -g disk_group \  
set user=oracle volume
```
- Change the owner of the mount point on which you want to implement Database Dynamic Storage Tiering to oracle.

Defining database parameters

Running the `dbdst_admin` command defines parameters for the entire database. You must run this command at least once to define the database parameters for Database Dynamic Storage Tiering. Three pre-defined storage classes will be created (PRIMARY, SECONDARY, and BALANCE). Parameter values are stored in the SFDB repository.

Set at least one of the parameters in `maxclass`, `minclass`, `statinterval`, `sweepinterval`, `purgetime`, or `purgeinterval`, to enable default values. Add at least one class to enable the default classes.

[Table 7-3](#) lists the options for the `dbdst_admin` command:

Table 7-3 dbdst_admin command options

Command option	Description
<code>-S \$ORACLE_SID</code>	Specifies the <code>ORACLE_SID</code> , which is the name of the Oracle instance.
<code>list</code>	Lists all of the Database Dynamic Storage Tiering parameters of the database, including class name and description. This option should be used exclusively from the other options.
<code>maxclass=</code>	Maximum number of storage classes allowed in the database. The default value is 4.
<code>minclass=</code>	Minimum number of storage classes allowed in the database. The default value is 2.

Table 7-3 dbdst_admin command options (continued)

Command option	Description
sweepinterval=	Interval for file sweeping for file relocation. Default value is 1, which means one per day. If this value is set to 0, all scheduled sweep tasks will become unscheduled.
sweeptime=	Time per day for the file sweep to take place. Times are entered in 24-hour periods and should list hour: minute. For example, 8:30 AM is represented as 08:30 and 10:00 PM is represented as 22:00. Default value is 22:00.
statinterval=	Interval in minutes for gathering file statistics. Default value is 30, which represents every 30 minutes. If this value is set to 0, all scheduled tasks will become unscheduled.
purgeinterval=	Number of days after which the file statistics in the repository will be summarized and purged. Default value is 30. It is recommended that you set your purge interval sooner because you will not be able to view any statistics until the first 30-day interval is over, if you use the default.
purgetime=	Time per day for the file purge to take place. Times are entered in 24-hour periods and should list hour: minute. For example, 8:30 AM is represented as 08:30 and 8:00 PM is represented as 20:00. Default value is 20:00.
addclass=	Parameter that allows you to add a class to a database. The information should be entered as class:"description", where the class represents the class name and description is a string of up to 64 characters enclosed by double quotes used to describe the class.
rmclass=	Parameter that allows you to remove a class from a database. Enter the class name as it appears in the database.

Note: If you do not want to change specific default values, you can omit those parameters when you run the dbdst_admin command. You only need to enter the parameters that need to be changed.

To define database parameters

- Use the `dbdst_admin` command as follows:

```
$ /opt/VRTS/bin/dbdst_admin -S $ORACLE_SID -o list,maxclass=number,\
minclass=number,sweepinterval=interval,sweeptime=hh:mm,\
statinterval=interval,purgeinterval=interval,purgetime=hh:mm,\
addclass=class:"description",rmclass=class
```

For example, to add a class called `tier1` for database `PROD`, and to set up a purge interval of one, meaning that the file statistics will be gathered for one day and then summarized and purged, use the `dbdst_admin` command as follows:

```
$ /opt/VRTS/bin/dbdst_admin -S PROD -o addclass=tier1:"Fast Storage",\
purgeinterval=1
```

Setting up storage classes

When you define your database parameters, three pre-defined storage classes are created. You will need to add or remove storage classes to meet your needs.

Adding storage classes

In addition to the default storage classes, you can add storage classes to better manage your data.

Before adding a storage class, review the following information:

- Use the `dbdst_admin` command as follows:

```
$ /opt/VRTS/bin/dbdst_admin -S $ORACLE_SID -o addclass=class:\
"description"
```

For example, to create a storage class named `"FAST"` for an EMC array, use the `dbdst_admin` command as follows:

```
$ /opt/VRTS/bin/dbdst_admin -S $ORACLE_SID -o addclass=FAST:\
"fast EMC array"
```

Removing storage classes

If you no longer require a specific storage class, you can remove it.

Note: You cannot remove the pre-defined storage classes (PRIMARY, SECONDARY, and BALANCE).

Before removing a storage class, review the following information:

- Use the `dbdst_admin` command as follows:

```
$ /opt/VRTS/bin/dbdst_admin -S $ORACLE_SID rmclass=class
```

For example, to remove a storage class called "SLOW," use the `dbdst_admin` command as follows:

```
$ /opt/VRTS/bin/dbdst_admin -S $ORACLE_SID rmclass=SLOW
```

Displaying storage classes

You can display a list of Database Dynamic Storage Tiering properties and storage classes using the `dbdst_admin` command.

Before displaying your storage classes, review the following information:

- Use the `dbdst_admin` command as follows:

```
$ /opt/VRTS/bin/dbdst_admin -S $ORACLE_SID -o list
```

Converting a VxFS file system to a VxFS multi-volume file system

To convert your existing VxFS file system to a VxFS multi-volume file system, you must convert a single volume to a volume set.

Converting a single volume to a volume set

When you convert to a volume set using the `dbdst_convert` command, the original volume will be renamed to a new volume name. The mount device name will become the new volume set name. Creating the new volume set name with the mount device name nullifies the need to rename the mount device in various locations.

Before converting to a volume set, make sure the following conditions have been met:

Prerequisites

- The Db database must not be active, and must be down.
- Create at least one additional volume.

Usage Notes

- You must convert the single-volume file system on which you plan to implement Database Dynamic Storage Tiering.
- The file system has to be unmounted when you run the `dbdst_convert` command.
- If the file system has *n* volumes, volumes 1 through *n*-1 will be placed in the storage class "PRIMARY" and volume *n* will be placed in the storage class "SECONDARY."
- The volumes specified when running the conversion must be in the same disk group as the mount device.

To convert a mount device from a single volume device to a volume set

- 1 Use the `dbdst_convert` command as follows:

```
$ /opt/VRTS/bin/dbdst_convert -S $ORACLE_SID -M mount_device -v \  
volume_name,volume_name
```

- 2 Bring the database objects online.

For example, to convert a volume-based `oradata` file system to a Database Dynamic Storage Tiering-ready volume set file system on mount device `/dev/vx/dsk/oradg/oradata`, use the `dbdst_convert` command as follows:

```
$ /opt/VRTS/bin/dbdst_convert -S PROD -M /dev/vx/dsk/oradg/oradata -v \  
new_vol1,new_vol2
```

After conversion, you will have a volume set named `oradata` containing three volumes (`oradata_b4vset`, `new_vol1`, and `new_vol2`). The file system will have two storage classes defined as `PRIMARY` and `SECONDARY`. The volumes will be assigned as follows:

- `PRIMARY` storage class will contain volumes `oradata_b4vset` and `new_vol1`.
- `SECONDARY` storage class will contain volume `new_vol2`.

Classifying volumes into a storage class

Before creating a DST policy or manually moving data, assign classes to your volumes.

Before assigning classes to volumes, review the following information:

Usage notes

- You must convert your VxFS file system to a multi-volume file system first.
- Storage classes must be registered using the `dbdst_admin` command before assigning classes to volumes.
- The database can be online or offline.

To classify a volume

- Use the `dbdst_classify` command as follows:

```
$ /opt/VRTS/bin/dbdst_classify -S $ORACLE_SID -M mount_device \  
-v volume_name:class[,volume_name:class]
```

For example, to assign the class "FAST" to volume `new_vol1`, use the `dbdst_classify` command as follows

```
$ /opt/VRTS/bin/dbdst_classify -S $ORACLE_SID -M /dev/vx/dsk/oradg/oradata \  
-v new_vol1:FAST
```

Displaying free space on your storage class

To see the free space, class information, and volume information on your storage classes, use the `dbdst_show_fs` command.

Table 7-4 shows the `dbdst_show_fs` command options.

Table 7-4 dbdst_show_fs command options

Command options	Description
-S \$ORACLE_SID	Specifies the <code>ORACLE_SID</code> , which is the name of the Oracle instance.
-o volume	Displays the free space on volumes in each class.
-m	Specifies the mount point.

Before displaying the free space on a storage class, review the following information:

Prerequisites

- Make sure the file system is mounted.
- See the `dbdst_show_sf` (1M) manual page.

To display the free space on a storage class

- Use the `dbdst_show_fs` command as follows:

```
$ /opt/VRTS/bin/dbdst_show_fs -S $ORACLE_SID -o volume \  
-m mount_point
```

Adding new volumes to a storage class

Use the `dbdst_addvol` command to add volumes to a volume set.

Before adding a volume, review the following information:

Usage notes

- The database must be inactive when adding volumes to a storage class.
- The database file system has to be mounted.

To add a volume to a volume set

- Use the `dbdst_addvol` command as follows:

```
$ /opt/VRTS/bin/dbdst_addvol -S $ORACLE_SID -M mount_device \  
-v volume_name:class[,volume_name:class]
```

Removing volumes from a storage class

You may need to remove a volume from a volume set. To remove a volume, use the `dbdst_rmvol` command.

Before removing a volume, review the following information:

Usage notes

- The database must be inactive when removing volumes from a storage class.
- Only a volume that does not contain any file system data can be removed.

To remove a volume from a volume set

Use the `dbdst_rmvol` command as follows:

```
$ /opt/VRTS/bin/dbdst_rmvol -S $ORACLE_SID -M mount_device \  
-v volume_name[,volume_name]
```

Extent balancing in a database environment

To obtain better performance in a database environment, you would normally use a volume striped over several disks. As the amount of data stored in the file system increases over time, additional space in the form of new disks must be added.

To increase space, you could perform a volume relayout using the `vxrelayout` command. However, changing a large volume from a four-way striped volume to six-way striped volume involves moving old block information into temporary space and writing those blocks from the temporary space to a new volume, which would require an extended amount of time. To solve this problem, Veritas Storage Foundation for Db provides the Extent Balanced File System or EBFS .

An Extent Balanced File System is created on a multi-volume file system where individual volumes are not striped over individual disks. For data-availability, these individual volumes can be mirrored. The file system on the EBFS has a special placement policy called a balance policy. When the balance policy is applied, all the files are divided into small "chunks" and the chunks are laid out on volumes so that adjacent chunks are on different volumes. The default chunk size is 1MB and can be modified. Since every file contains chunks on all available volumes, it is important that individual volumes that make up the EBFS and volume set be of same size and same access properties.

Setting up the file system in this way provides the same benefit as striping your volumes.

Note: You cannot convert an existing file system to an EBFS file system.

Extent balancing file system

You can define allocation policies with a balance allocation order and "chunk" size to files or a file system, known as extent balancing. The chunk size is the maximum size of any extent that files or a file system with this assigned policy can have. The chunk size can only be specified for allocation policies with a balance allocation order.

An extent balancing policy specifies the balance allocation order and a non-zero chunk size. The balance allocation order distributes allocations randomly across the volumes specified in the policy and limits each allocation to a maximum size equal to the specified chunk size.

Extent balancing extends the behavior of policy enforcement by rebalancing extent allocations such that each volume in the policy is as equally used as possible. Policy enforcement handles the following cases:

- New volumes are added to the policy, and the extents associated with a file need rebalancing across all volumes, including the new ones.
- Volumes are removed from the volume set or from the policy, and the extents for a file residing on a removed volume need to be moved to other volumes in the policy.
- An extent balancing policy is assigned to a file and its extents have to be reorganized to meet the chunk size requirements defined in the policy.

The extent balancing policy is intended for balancing data extents belonging to files across volumes defined in the policy. However, there is no restriction imposed in assigning extent balancing policy for metadata.

Note: If the fixed extent size is less than the chunk size, then the extent size will be limited to the largest multiple of the fixed extent size that is less than the chunk size. If the fixed extent size is greater than the chunk size, then the extent size will be the fixed extent size.

Creating an extent balanced file system

Any MultiVolume File System (MVFS) can become an extent balanced file system, if the storage tier has a chunk size associated with the class. The `dbdst_admin` command permits the user to define a chunk size for the class.

For example, the following `dbdst_admin` commands define chunk sizes for the gold and silver storage classes:

```
$ /opt/VRTS/bin/dbdst_admin -S $ORACLE_SID -o definechunk gold:256K
$ /opt/VRTS/bin/dbdst_admin -S $ORACLE_SID -o definechunk silver:128K
```

The above commands make storage class gold as extent balanced.

Once the chunksize is defined for a storage tier, we can classify any MVFS into this storage tier.

For example, assume that `/oradata` is the filesystem created on volume-set `/dev/vx/dsk/oradg/ora_vset`, and contains database datafiles. Let us further assume that datafile names end with extension `*.dbf`. To define storage class in this MVFS, the following `dbdst_classify` command is used:

```
$ /opt/VRTS/bin/dbdst_classify -S $ORACLE_SID -M /dev/vx/dsk/oradg/ora_vset -v \
vol1:GOLD,vol2:GOLD,vol3:GOLD
```

It is important to note that, an MVFS can have multiple storage tiers and that each tier may have a different chunk size. For example, for the same MVFS in the

above example, we can define another storage tier using the `dbdst_classify` command:

```
$ /opt/VRTS/bin/dbdst_classify -S $ORACLE_SID -M /dev/vx/dsk/oradg/ora_vset -v \
vol4:silver,vol5:silver
```

At this point we have two storage tiers in MVFS /oradata each having different chunksizes. To create the real extent balance, we need to assign a DST policy and to enforce it.

To define and enforce the policy, you could use the following `dbdst_preset_policy` command:

```
$ /opt/VRTS/bin/dbdst_preset_policy -S $ORACLE_SID -d /oradata -P GOLD=*.dbf,SILVER=*.inx
```

The above example creates a DST policy, assigns the policy to /oradata and enforces the policy. All datafiles of the form *.dbf will be extent balanced in GOLD tier with chunksize 256K and all index files of the form *.inx will be extent balanced in SILVER tier with chunk size 128K.

To view the space usage in the /oradata MVFS use the `dbdst_show_fs` command. For example:

```
$ /opt/VRTS/bin/dbdst_show_fs -S $ORACLE_SID -m /oradata
```

When the GOLD or SILVER tier requires more space, we could add extra space by adding new volumes to the respective storage tier using the `dbdst_addvol` command. For example:

```
$ /opt/VRTS/bin/dbdst_addvol -S $ORACLE_SID -M /dev/vx/dsk/oradg/ora_vset -v vol7:GOLD
```

As soon as you add a new volume, the DST policy is enforced and the extents are balanced over the new volume too. This can be viewed by using the `dbdst_show_fs` command again.

To view detailed extent information about a given file, you can use the `fsmmap` command. For example:

```
$ /opt/VRTS/bin/fsmmap -a /oradata/emp.dbf
```

Running Database Dynamic Storage Tiering reports

You can create a report that lists all updated allocation policies or you can view an audit report, which lists recent relocation changes for a specific date range resulting from your policies.

Viewing modified allocation policies

To create a list of modified allocation policies, use the `dbdst_report` command with the `policy` option.

To list allocation policies

- Use the `dbdst_report` command as follows:

```
$ /opt/VRTS/bin/dbdst_report -S $ORACLE_SID -o policy
```

For example to view a list of modified allocation policies, use the `dbdst_report` command as follows:

```
$ /opt/VRTS/bin/dbdst_report -S $ORACLE_SID -o policy
```

Viewing audit reports

To view an audit report, which lists recent file relocation changes within a specific date range, use the `dbdst_report` command with the `audit` option.

To view an audit report

- Use the `dbdst_report` command as follows:

```
$ /opt/VRTS/bin/dbdst_report -S $ORACLE_SID -o audit \
startdate=yyyy-mm-dd,enddate=yyyy-mm-dd
```

For example, to view an audit report of changes from January 1, 2007 through March 1, 2007, use the `dbdst_report` command as follows:

```
$ /opt/VRTS/bin/dbdst_report -S $ORACLE_SID -o audit \
startdate=2007-01-01,enddate=2007-03-01
```

Oracle Database Dynamic Storage Tiering use cases

This section discusses Oracle use cases for Dynamic Storage Tiering.

Migrating partitioned data and tablespaces

Perhaps the simplest application of multi-tier storage to databases is relocation of individual table partitions between different placement classes as usage requirements change. If exact relocation times are unpredictable, or if relocation

is infrequent, administrators may wish to relocate table partitions when necessary rather than defining strict periodic relocation schedules.

Ad hoc relocation of table partitions can be useful, for example, with databases that track sales and inventory for seasonal businesses such as sports equipment or outdoor furniture retailing. As the selling season for one type of inventory (for example, summer equipment or furniture) approaches, database table partitions that represent in-season goods can be relocated to high-performance storage, since they will be accessed frequently during the coming months. Similarly, partitions that represent out-of-season goods can be relocated to lower-cost storage, since activity against them is likely to be infrequent.

For example, sales are mostly catalog-driven for a large retailer specializing in sports equipment. Product details are saved in a large database and the product table is partitioned based on type of activity. Some of the products are seasonal and do not sell well at other times. For example, very few snow skis are sold during the summer. To achieve season-based migration, see the following example. Assume the table `product_tab` has two partitions, `summer` and `winter`. Each of these partitions is mapped to a separate data file.

First, you must set up your system to use Database Dynamic Storage Tiering.

To add the `fast_storage` and `slow_storage` storage classes

- ◆ Use the `dbdst_admin` command as follows:

```
$ /opt/VRTS/bin/dbdst_admin -S PROD -o addclass=\
fast_storage:"Fast Storage for Production DB"

$ /opt/VRTS/bin/dbdst_admin -S PROD -o addclass=\
slow_storage:"Slow Storage for Production DB"
```

To convert the database's file system and add volumes for use with Database Dynamic Storage Tiering

- ◆ Use the `dbdst_convert` command as follows:

```
$ /opt/VRTS/bin/dbdst_convert -S PROD \
-M /dev/vx/dsk/oradg/oradata -v new_vol1,new_vol2,new_vol3
```

To classify volumes into storage classes

- ◆ Use the `dbdst_classify` command as follows:

```
$ /opt/VRTS/bin/dbdst_classify -S PROD \  
-M /dev/vx/dsk/oradg/oradata -v new_vol1:fast_storage  
  
$ /opt/VRTS/bin/dbdst_classify -S PROD \  
-M /dev/vx/dsk/oradg/oradata -v new_vol2:slow_storage,\  
new_vol3:slow_storage
```

Once the volumes are configured, an administrator can define file placement policy rules that specify seasonal relocation of selected tablespaces and partitions and assign them to the database's file system.

To move summer data to slower storage and winter data to faster storage at the beginning of winter

- ◆ Use the `dbdst_partition_move` command as follows:

```
$ /opt/VRTS/bin/dbdst_partition_move -S PROD -T product_tab \  
-p winter -c fast_storage  
  
$ /opt/VRTS/bin/dbdst_partition_move -S PROD -T product_tab \  
-p summer -c slow_storage
```

These commands relocate the files that comprise the winter partition of the `product_tab` table to placement class `fast_storage`, and the files that comprise the summer partition to placement class `slow_storage`. Database Dynamic Storage Tiering determines which files comprise the winter and summer partitions of `product_tab`, and uses underlying DST services to immediately relocate those files to the `fast_storage` and `slow_storage` placement classes respectively.

To move winter data to slower storage and summer data to faster storage at the beginning of summer

- ◆ Use the `dbdst_partition_move` command as follows:

```
$ /opt/VRTS/bin/dbdst_partition_move -S PROD -T product_tab \  
-p summer -c fast_storage  
  
$ /opt/VRTS/bin/dbdst_partition_move -S PROD -T product_tab \  
-p winter -c slow_storage
```

Database Dynamic Storage Tiering formulates DST policy rules that unconditionally relocate the files containing the target partitions to the destination placement classes. It merges these rules into the database file system's active

policy, assigns the resulting composite policy to the file system, and enforces it immediately to relocate the subject files. Because the added policy rules precede any other rules in the active policy, the subject files remain in place until the `dbdst_partition_move` command is next executed, at which time the rules are removed and replaced with others.

Scheduling the relocation of archive and Flashback logs

Because they are the primary mechanism for recovering from data corruption, database logs are normally kept on premium storage, both for I/O performance and data reliability reasons. Even after they have been archived, logs are normally kept online for fast recovery, but the likelihood of referring to an archived log decreases significantly as its age increases. This suggests that archived database logs might be relocated to lower-cost volumes after a certain period of inactivity.

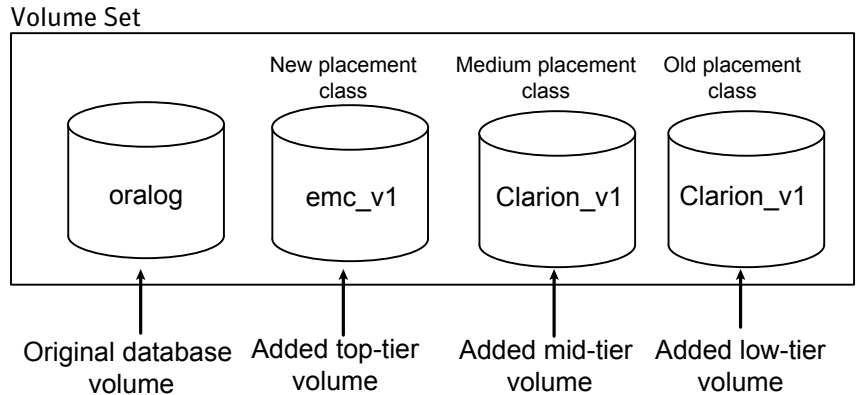
Similarly, Veritas Storage Foundation for DB Flashback technology creates logs that can be used for quick recovery from database corruption by restoring a database to its state at a previous time. Flashback logs are normally kept for a shorter period than archived database logs, if used at all, they are typically used within a few hours of creation. Two or three days are a typical Flashback log lifetime.

The rapidly decaying probability of use for archive and Flashback logs suggests that regular enforcement of a placement policy that relocates them to lower-cost storage after a period of inactivity can reduce an enterprise's average cost of online storage.

For example, a customer could be using a large OLTP Oracle database with thousands of active sessions, which needs to be up and running 24 hours a day and seven days a week with uptime of over 99%, and the database uses Flashback technology to correct any accidental errors quickly. The database generates large number of archive logs per day. If the database goes down for any reason, there is business requirement to bring the database back online and functional within 15 minutes. To prevent Oracle log switch delays during transactions, the archive logs need to be created in a fast EMC array. Archive logs older than a week can be moved to a mid-range Clarion array. Archive logs older than 15 days can be moved to slow JBOD disks. Archive logs are purged after 30 days. Current Flashback logs are created manually by the database administrator on fast EMC storage and can be moved to Clarion storage after two days. The database administrator then deletes the Flashback logs after a week. To set up a system like this, see the following example. Assume that archive logs and Flashback logs are created on the same file system, `/oralog`. On the file system, `/oralog/archive1` contains archive logs and `/oralog/flashback` contains Flashback logs.

Figure 7-1 illustrates a three-tier volume configuration that is suitable for automatic relocation and deletion of archive logs and Flashback logs.

Figure 7-1 Database storage configuration suitable for automatic relocation of archive and Flashback logs



The file system used by the production database in this example originally resides on the single volume oralog, which must be prepared by adding volumes and placement classes assigned to the volumes.

To add the NEW, MEDIUM, and OLD storage classes

- ◆ Use the `dbdst_admin` command as follows:

```
$ /opt/VRTS/bin/dbdst_admin -S PROD -o addclass=\
NEW:"EMC Storage for Production DB"

$ /opt/VRTS/bin/dbdst_admin -S PROD -o addclass=\
MEDIUM:"Clarion Storage for Production DB"

$ /opt/VRTS/bin/dbdst_admin -S PROD -o addclass=\
OLD:"JBOD Storage for Production DB"
```

To convert the database's file system and add volumes for use with Database Dynamic Storage Tiering

- ◆ Use the `dbdst_convert` command as follows:

```
$ /opt/VRTS/bin/dbdst_convert -S PROD \
-M /dev/vx/dsk/oradg/oralog -v emc_v1,clarion_v1,jbod_v1
```

To classify volumes into storage classes

- ◆ Use the `dbdst_classify` command as follows:

```
$ /opt/VRTS/bin/dbdst_classify -S PROD \
-M /dev/vx/dsk/oradg/oralog -v emc_v1:NEW

$ /opt/VRTS/bin/dbdst_classify -S PROD \
-M /dev/vx/dsk/oradg/oralog -v clarion_v1:MEDIUM

$ /opt/VRTS/bin/dbdst_classify -S PROD \
-M /dev/vx/dsk/oradg/oralog -v jbod_v1:OLD
```

Once the volumes are configured, an administrator can define file placement policy rules that specify access age-based relocation of selected files and assign them to the database's file system.

To define rules that periodically relocate Flashback and archive logs

- ◆ Use the `dbdst_file_move` command as follows:

```
$ /opt/VRTS/bin/dbdst_file_move -S PROD -o flashback -c MEDIUM:2
```

This command relocates files in the Flashback directory that have not been accessed for two days to the MEDIUM volume.

```
$ /opt/VRTS/bin/dbdst_file_move -S PROD -o archive1 -c MEDIUM:7 \
-c OLD:15
```

This command relocates files in the archive1 directory that have not been accessed for seven days to the MEDIUM volume, and files that have not been accessed for 15 days to the OLD volume.

Database Dynamic Storage Tiering translates these commands into DST access age-based policy rules, merges them with the file system's placement policy, and assigns the resulting policy to the file system. By default, Database Dynamic Storage Tiering enforces the active policy daily. During enforcement, the new rules relocate qualifying files to the destination storage tiers specified in the `dbdst_file_move` commands used to create the policies.

Performance and troubleshooting

- [Investigating I/O performance using storage mapping](#)
- [Troubleshooting SF Oracle RAC](#)
- [Prevention and recovery strategies](#)
- [Tunable parameters](#)

Investigating I/O performance using storage mapping

This chapter includes the following topics:

- [About Storage Mapping in SF Oracle RAC](#)
- [Understanding Storage Mapping](#)
- [Verifying Veritas Storage Mapping set up](#)
- [Using the vxstorage_stats command](#)
- [Using the dbed_analyzer command](#)
- [About arrays for Storage Mapping and statistics](#)
- [Oracle File Mapping \(ORAMAP\)](#)

About Storage Mapping in SF Oracle RAC

The storage mapping feature is available with SF Oracle RAC and enables you to map datafiles to physical devices. You can obtain and view detailed storage topology information using the `vxstorage_stats` and `dbed_analyzer` commands.

Understanding Storage Mapping

Access to mapping information is important since it allows for a detailed understanding of the storage hierarchy in which files reside, information that is critical for effectively evaluating I/O performance.

Mapping files to their underlying device is straightforward when datafiles are created directly on a raw device. With the introduction of host-based volume managers and sophisticated storage subsystems that provide RAID features, however, mapping files to physical devices has become more difficult.

With the SF Oracle RAC Storage Mapping option, you can map datafiles to physical devices. Veritas Storage Mapping relies on Veritas Mapping Service (VxMS), a library that assists in the development of distributed SAN applications that must share information about the physical location of files and volumes on a disk.

The Veritas Storage Mapping option supports Oracle's set of storage APIs called Oracle Mapping ("ORAMAP" for short) that lets Oracle determine the mapping information for files and devices.

Oracle provides a set of dynamic performance views (v\$ views) that shows the complete mapping of a file to intermediate layers of logical volumes and physical devices. These views enable you to locate the exact disk on which any specific block of a file resides. You can use these mappings, along with device statistics, to evaluate I/O performance.

The Veritas Storage Mapping option supports a wide range of storage devices and allows for "deep mapping" into EMC, Hitachi, and IBM Enterprise Storage Server ("Shark") arrays. Before proceeding with storage mapping, consult the current compatibility list in the Veritas Technical Support website to confirm the compatibility of your hardware:

<http://entsupport.symantec.com/docs/283161>

Deep mapping information identifies the physical disks that comprise each LUN and the hardware RAID information for the LUNs. You can view storage mapping topology information and I/O statistics using the following commands:

<code>vxstorage_stats</code> command	This command displays the complete I/O topology mapping of specific datafiles through intermediate layers like logical volumes down to actual physical devices.
<code>dbed_analyzer</code> command	This command retrieves tablespace-to-physical disk mapping information for all the datafiles in a specified database. It also provides information about the amount of disk space being used by a tablespace.

For information on the command line options, see details on storage mapping in the *Veritas Storage Foundation for Oracle Administrator's Guide*.

Note: For Solaris users, if you use Veritas FlashSnap Agent for Symmetrix, you cannot use the mapping functionality for non-Symmetrix arrays.

Verifying Veritas Storage Mapping set up

Before using the Veritas Storage Mapping option, verify that the features are set up correctly.

To verify that your system is using the Veritas Storage Mapping option

- 1 Verify that you have a license key for the storage mapping option.

```
# /opt/VRTS/bin/vxlictest -n "VERITAS Mapping Services" -f \
"Found_Edi_map"
```

```
Found_Edi_map feature is licensed
```

- 2 Verify that the VRTSvxmsa package is installed.

```
# pkginfo -l VRTSvxmsa
```

Output similar to the following is displayed:

```
PKGINST:  VRTSvxmsa
NAME:      VxMS - Veritas Mapping Service, Application Libraries.
CATEGORY:  system,utilities
ARCH:      sparc
VERSION:   4.2.1-REV=build218_2004.10.29
BASEDIR:   /opt
PSTAMP:    oigpsol0920041029112628
INSTDATE:  Nov 02 2004 15:22
STATUS:    completely installed
FILES:     33 installed pathnames
           8 shared pathnames
           14 directories
           15 executables
           2931 blocks used (approx)
```

Using the vxstorage_stats command

The `vxstorage_stats` command displays detailed storage mapping information and I/O statistics about one or more VxFS files. The mapping information and I/O statistics are recorded only for VxFS files and VxVM volumes.

In `vxstorage_stats` command output, I/O topology information appears first followed by summary statistics for each object.

The command syntax is as follows:

```
$ /opt/VRTSdbed/bin/vxstorage_stats -m -s [-I interval -c count]
-f filename
```

- Prerequisites
- You must log in as the database administrator (typically, the user ID oracle) or superuser.
- Usage Notes
- The `-s` option displays the file statistics for the specified file.
 - The `-c count` option specifies the number of times to display statistics within the interval specified by `-I interval`.
 - The `-I interval` option specifies the interval frequency for displaying updated I/O statistics.
 - The `-f filename` option specifies the file to display I/O mapping and statistics for.
 - The `-m` option displays the I/O topology for the specified file.
 - For more information, see the `vxstorage_stats(1m)` online manual page.
- Note:** The `vxstorage_stats` command process can at times take up to 30 minutes to complete.

Displaying Storage Mapping information

Review the procedure to display Storage Mapping information.

To display storage mapping information

- ◆ Use the `vxstorage_stats` command with the `-m` option to display storage mapping information:

```
$ /opt/VRTSdbed/bin/vxstorage_stats -m -f file_name
```

For example, as oracle user enter the following command:

```
$ /opt/VRTSdbed/bin/vxstorage_stats -m -f /oradata/system01.dbf
```

Output similar to the following is displayed.

da	EMC000184502242:32:0c:02	0	DISK	143113019	0
rd	/dev/vx/rdmp/c3t1d2s4	1	PARTITION	5593088	0
sd	/dev/rdsk/c3t1d2s4	1	PARTITION	17671680	3840
sd	c3t1d2	2	MIRROR	17677440	0
da	EMC000184502242:16:0c:02	0	DISK	143113019	0
da	EMC000184502242:17:0c:02	0	DISK	143113019	0
TY	NAME	NSUB	DESCRIPTION	SIZE (sectors)	OFFSET (sectors) PROPERTIES

```

fi /oradata/system01.dbf      1      FILE      2621442048 (B) 4718592 (B)
Extents:3 Sparse Extents:0
v myindex                    1      MIRROR      16777216      0
pl vxvm:mydb/myindex-01      3      STRIPE      16779264      0
Stripe_size:2048
rd /dev/vx/rdmp/c3t1d3s3      1      PARTITION    5593088      0
sd /dev/rdsk/c3t1d3s3          1      PARTITION    17674560     960
sd c3t1d3                      2      MIRROR      17677440     0
da EMC000184502242:02:0c:02  0      DISK        143113019    0
da EMC000184502242:31:0c:02  0      DISK        143113019    0
rd /dev/vx/rdmp/c3t1d15s4      1      PARTITION    5593088      0
sd /dev/rdsk/c3t1d15s4          1      PARTITION    17669760     5760
sd c3t1d15                     2      MIRROR      17677440     0
da EMC000184502242:01:0c:02  0      DISK        143113019    0
da EMC000184502242:32:0c:02  0      DISK        143113019    0
rd /dev/vx/rdmp/c3t1d2s4      1      PARTITION    5593088      0
sd /dev/rdsk/c3t1d2s4          1      PARTITION    17671680     3840
sd c3t1d2                      2      MIRROR      17677440     0
da EMC000184502242:16:0c:02  0      DISK        143113019    0
da EMC000184502242:17:0c:02  0      DISK        143113019    0

```

Note: For file type (fi), the SIZE column is number of bytes; for volume (v), plex (pl), dmp device (rd), sub-disk (sd), and physical disk (da), the SIZE column is in 512-byte blocks. Stripe sizes are given in sectors.

Displaying I/O statistics information

Review the procedure to display the I/O statistics information.

To display I/O statistics information

- ◆ Use the vxstorage_stats command with the -s option to display I/O statistics information:

```
$ /opt/VRTSdbed/bin/vxstorage_stats -s -f file_name
```

For example, as oracle user enter the following command:

```
$ /opt/VRTSdbed/bin/vxstorage_stats -s -f \
/data/system01.dbf
```

Output similar to the following is displayed.

OBJECT	I/O OPERATIONS		I/O BLOCKS (512 byte)		AVG TIME (ms)	
	READ	WRITE	B_READ	B_WRITE	AVG_RD	AVG_WR
/data/system01.dbf	2	2479	8	5068810	0.00	53.28
/dev/vx/rdisk/mydb/myindex	101	2497	1592	5069056	12.18	52.78
vxvm:mydb/myindex-01	101	2497	1592	5069056	12.18	52.76
/dev/rdisk/c3t1d3s3	131	1656	2096	1689696	14.43	39.09
c3t1d3	131	1656	2096	1689696	14.43	39.09
EMC000184502242:02:0c:02	8480	231019	275952	23296162	-	-
EMC000184502242:31:0c:02	3244	232131	54808	23451325	-	-
/dev/rdisk/c3t1d15s4	0	1652	0	1689606	0.00	46.47
c3t1d15	0	1652	0	1689606	0.00	46.47
EMC000184502242:01:0c:02	23824	1188997	1038336	32407727	-	-
EMC000184502242:32:0c:02	5085	852384	135672	29956179	-	-
/dev/rdisk/c3t1d2s4	14	1668	200	1689834	18.57	34.19
c3t1d2	14	1668	200	1689834	18.57	34.19
EMC000184502242:16:0c:02	4406	271155	121368	23463948	-	-
EMC000184502242:17:0c:02	3290	269281	55432	23304619	-	-

To display Storage Mapping and I/O statistics information at repeated intervals

- ◆ Use the `vxstorage_stats` command with the `-I interval` and `-c count` options. The `-I interval` option specifies the interval frequency for displaying updated I/O statistics and the `-c count` option specifies the number of times to display statistics:

```
$ /opt/VRTSdbed/bin/vxstorage_stats [-m] [-s] \
[-I interval -c count] -f file_name
```

For example, enter the following command to display statistics two times with a time interval of two seconds:

```
$ /opt/VRTSdbed/bin/vxstorage_stats -s -i2 -c2 \-f /data/system01.dbf
```

Using the `dbed_analyzer` command

You must have an understanding of which tablespaces reside on which disks to effectively perform a parallel backup. If two tablespaces reside on the same disk, for example, backing them up in parallel will not reduce their backup time.

The `dbed_analyzer` command provides tablespace-to-physical disk mapping information for all the datafiles in a specified tablespace, list of tablespaces, or an entire database. (In contrast, the `vxstorage_stats` command provides this

information on a per-file basis only.) In addition, `dbed_analyzer` provides information about the amount of disk space they are using.

Prerequisites	■ You must log in as the database administrator (typically, the user ID oracle).
Usage Notes	■ The <code>-o sort=tbs</code> option provides the layout of the specified tablespaces on the physical disk as well as the amount of disk space they are using. ■ The <code>-o sort=disk</code> option provides the name of the disks containing the specified tablespaces as well as the amount of disk space the tablespaces are using. ■ The <code>-f filename</code> option specifies the name of a file containing a list of the tablespaces for which to obtain mapping information. ■ The <code>-t tablespace</code> option specifies the name of a tablespace for which to obtain mapping information. ■ If <code>-f filename</code> or <code>-t tablespace</code> is not specified then all the tablespaces in the database will be analyzed. ■ For more information, see the <code>dbed_analyzer(1M)</code> online manual page.

Note: The `dbed_analyzer` command process may take a significant amount of time to complete.

Obtaining Storage Mapping information for a list of tablespaces

Review the procedure to obtain Storage Mapping information for a list of tablespaces.

To obtain Storage Mapping information sorted by tablespace

- ◆ Use the dbed_analyzer command with the -f filename and -o sort=tbs options:

```
$ /opt/VRTSdbed/bin/dbed_analyzer -S $ORACLE_SID \
-H $ORACLE_HOME -o sort=tbs -f filename
```

For example, as oracle user enter the following command:

```
$ /opt/VRTSdbed/bin/dbed_analyzer -S PROD -H /usr1/oracle \
-o sort=tbs -f /tmp/tbsfile
```

For a Solaris OS system, output similar to the following is displayed in the file tbsfile:

TBSNAME	DATAFILE	DEVICE	SIZE(sectors)
SYSTEM	/usr1/oracle/rw/DATA/PROD.dbf	c3t21000020379DBD5Fd0	819216
TEMP	/usr1/oracle/rw/DATA/temp_20000	c3t21000020379DBD5Fd0	1021968
TEMP	/usr1/oracle/rw/DATA/temp_20001	c3t21000020379DBD5Fd0	2048016
SYS_AUX	/usr1/oracle/rw/DATA/sysaux.dbf	c3t21000020379DBD5Fd0	819216
ITEM	/usr1/oracle/rw/DATA/item_1000	c3t21000020379DBD5Fd0	1021968
ITM_IDX	/usr1/oracle/rw/DATA/itm_idx_2000	c3t21000020379DBD5Fd0	1021968
PRODIG_IDX	/usr1/oracle/rw/DATA/prodig_idx_3000	c3t21000020379DBD5Fd0	1021968
QTY_IDX	/usr1/oracle/rw/DATA/qty_idx_7000	c3t21000020379DBD5Fd0	1021968
ROLL_1	/usr1/oracle/rw/DATA/roll_1_5000	c3t21000020379DBD5Fd0	1021968
ROLL_2	/usr1/oracle/rw/DATA/roll_2_6000	c3t21000020379DBD5Fd0	1021968
ORDERS	/usr1/oracle/rw/DATA/orders_4000	c3t21000020379DBD5Fd0	1021968
ORD_IDX	/usr1/oracle/rw/DATA/ord_idx_10000	c3t21000020379DBD5Fd0	1021968
QTY_IDX	/usr1/oracle/rw/DATA/qty_idx_7001	c3t21000020379DBD5Fd0	1024016
ITM_IDX	/usr1/oracle/rw/DATA/itm_idx_2001	c3t21000020379DBD5Fd0	1024016
ROLL_1	/usr1/oracle/rw/DATA/roll_1_5001	c3t21000020379DBD5Fd0	1024016
QTY_IDX	/usr1/oracle/rw/DATA/qty_idx_7002	c3t21000020379DBD5Fd0	1024016
ROLL_2	/usr1/oracle/rw/DATA/roll_2_6001	c3t21000020379DBD5Fd0	1024016
ITEM	/usr1/oracle/rw/DATA/item_1001	c3t21000020379DBD5Fd0	4096016

To obtain Storage Mapping information sorted by disk

- ◆ Use the `dbed_analyzer` command with the `-f filename` and `-o sort=disk` options:

```
$ /opt/VRTSdbed/bin/dbed_analyzer -S $ORACLE_SID \
-H $ORACLE_HOME -o sort=disk -f filename
```

For example, as oracle user enter the following command:

```
$ /opt/VRTSdbed/bin/dbed_analyzer -S PROD -H /usr1/oracle \
-o sort=disk -f /tmp/tbsfile
```

For a Solaris OS system, output similar to the following is displayed in the file `tbsfile`:

DEVICE	TBSNAME	DATAFILE	SIZE(sectors)
c3t21000020379DBD5Fd0	SYSTEM	/usr1/oracle/rw/DATA/PROD.dbf	819216
c3t21000020379DBD5Fd0	TEMP	/usr1/oracle/rw/DATA/temp_20000	1021968
c3t21000020379DBD5Fd0	TEMP	/usr1/oracle/rw/DATA/temp_20001	2048016
c3t21000020379DBD5Fd0	SYS_AUX	/usr1/oracle/rw/DATA/sysaux.dbf	819216
c3t21000020379DBD5Fd0	ITEM	/usr1/oracle/rw/DATA/item_1000	1021968
c3t21000020379DBD5Fd0	ITM_IDX	/usr1/oracle/rw/DATA/itm_idx_2000	1021968
c3t21000020379DBD5Fd0	PROID_IDX	/usr1/oracle/rw/DATA/prodid_idx_3000	1021968
c3t21000020379DBD5Fd0	QTY_IDX	/usr1/oracle/rw/DATA/qty_idx_7000	1021968
c3t21000020379DBD5Fd0	ROLL_1	/usr1/oracle/rw/DATA/roll_1_5000	1021968
c3t21000020379DBD5Fd0	ROLL_2	/usr1/oracle/rw/DATA/roll_2_6000	1021968
c3t21000020379DBD5Fd0	ORDERS	/usr1/oracle/rw/DATA/orders_4000	1021968
c3t21000020379DBD5Fd0	ORD_IDX	/usr1/oracle/rw/DATA/ord_idx_10000	1021968
c3t21000020379DBD5Fd0	QTY_IDX	/usr1/oracle/rw/DATA/qty_idx_7001	1024016
c3t21000020379DBD5Fd0	ITM_IDX	/usr1/oracle/rw/DATA/itm_idx_2001	1024016
c3t21000020379DBD5Fd0	ROLL_1	/usr1/oracle/rw/DATA/roll_1_5001	1024016
c3t21000020379DBD5Fd0	QTY_IDX	/usr1/oracle/rw/DATA/qty_idx_7002	1024016
c3t21000020379DBD5Fd0	ROLL_2	/usr1/oracle/rw/DATA/roll_2_6001	1024016
c3t21000020379DBD5Fd0	ITEM	/usr1/oracle/rw/DATA/item_1001	4096016

About arrays for Storage Mapping and statistics

SF Oracle RAC provides "deep" mapping information and performance statistics for supported storage arrays. Deep mapping information consists of identifying the physical disks that comprise each LUN and the hardware RAID information for the LUNs.

Note: To use deep mapping, you must have Oracle 9.2.0.3. or later installed, but deep mapping is not supported for Oracle RAC 11g.

Veritas Array Integration Layer (VAIL) software interfaces third party hardware storage arrays with Veritas storage software. VAIL providers are software modules that enable Veritas applications to discover, query, and manage third party storage arrays.

For the most up-to-date array support information, see the appropriate hardware compatibility list (HCL) on the Veritas Technical Support Web page at:

<http://entsupport.symantec.com/docs/283161>

If you want to use storage array information accessible through the VAIL providers, install VAIL and perform any required configuration for the storage arrays and VAIL providers. To use deep mapping services and performance statistics for supported storage arrays, you must install both VAIL and Veritas Mapping Services (VxMS).

You will need to install required third party array CLIs and APIs on the host where you are going to install VAIL before you install VAIL. If you install any required CLI or API after you install VAIL, rescan the arrays so that SF Oracle RAC can discover them.

For details on supported array models, see the *Veritas Array Integration Layer Array Configuration Guide*.

Oracle File Mapping (ORAMAP)

With Veritas Storage Mapping option, you can view the complete I/O topology mapping of datafiles through intermediate layers like logical volumes down to actual physical devices. You can use this information to determine the exact location of an Oracle data block on a physical device and to help identify hot spots.

Note: Mapping functionality requires Oracle 9.2.0.3 or a later version, but is not supported for Oracle RAC 11g.

Mapping components

Review the mapping components in your Oracle documentation. You will need an understanding of these components to interpret the mapping information in Oracle's dynamic performance views.

The mapping information in Oracle's dynamic performance views consists of the following components:

- File components

A mapping file component is a mapping structure describing a file. It provides a set of attributes for a file, including the file's size, number of extents, and type. File components are exported to the user through V\$MAP_FILE.

- File extent components

A mapping file extent component describes a contiguous group of blocks residing on one element. The description specifies the device offset, the extent size, the file offset, the extent type (Data or Parity), and the name of the element where the extent resides.

- Element components

A mapping element component is a mapping structure that describes a storage component within the I/O stack. Elements can be mirrors, stripes, partitions, RAID5, concatenated elements, and disks.

This component contains information about the element's mapping structure, such as the element's size, type, number of subelements, and a brief description. Element components are exported to the user through V\$MAP_ELEMENT.

- Subelement components

A mapping subelement component describes the link between an element and the next element in the I/O stack. The subelement component contains the subelement number, size, the element name for the subelement, and the element offset. Subelement components are exported to the user through V\$MAP_SUBELEMENT.

These four types of mapping components completely describe the mapping information for an Oracle instance.

Storage Mapping views

The mapping information that is captured is presented in Oracle's dynamic performance views.

[Table 8-1](#) provides brief descriptions of these views.

For more detailed information, refer to your Oracle documentation.

Table 8-1 Storage Mapping views

View	Description
V\$MAP_LIBRARY	Contains a list of all the mapping libraries that have been dynamically loaded by the external process.
V\$MAP_FILE	Contains a list of all the file mapping structures in the shared memory of the instance.

Table 8-1 Storage Mapping views (continued)

View	Description
V\$MAP_FILE_EXTENT	Contains a list of all the file extent mapping structures in the shared memory of the instance.
V\$MAP_ELEMENT	Contains a list of all the element mapping structures in the SGA of the instance.
V\$MAP_EXT_ELEMENT	Contains supplementary information for all element mapping structures.
V\$MAP_SUBELEMENT	Contains a list of all subelement mapping structures in the shared memory of the instance.
V\$MAP_COMP_LIST	Describes the component list associated with the element name.
V\$MAP_FILE_IO_STACK	Contains the hierarchical arrangement of storage containers for the file. This information is displayed as a series of rows. Each row represents a level in the hierarchy.

Verifying Oracle file mapping set up

Verify the set up of Oracle file mapping.

To verify that \$ORACLE_HOME is set up for Oracle file mapping (ORAMAP)

1 Verify whether \$ORACLE_HOME is ready for Oracle file mapping (ORAMAP):

```
# cd $ORACLE_HOME/rdbms/filemap/bin
# ls -l
-r-xr-x--- 1 root system 900616 Apr 08 19:16 fmpu1
-r-sr-xr-x 1 root system 14614 Apr 08 19:16 fmpu1hp
```

2 Confirm the following items and make the appropriate corrections:

- root owns fmpu1hp and the setuid bit is set.
- The permissions for fmpu1hp are -r-sr-xr-x.
- The permissions for fmpu1 are -r-xr-x---

3 If any of these items is not set as specified, make the appropriate corrections.

Enabling Oracle file mapping

After verifying the Oracle file mapping set up, enable Oracle file mapping.

To enable Oracle file mapping with the Veritas Storage Mapping option

- 1 Ensure that the file filemap.ora exists, and contains a valid entry for the Veritas mapping library for Oracle storage mapping.

Enter the following commands:

```
# cd $ORACLE_HOME/rdbms/filemap/etc  
  
# cat filemap.ora
```

For the Solaris OS for 32-bit Oracle, the filemap.ora file should contain the following setting:

```
lib=Veritas:/opt/VRTSdbed/lib/libvxoramap_32.so
```

For the Solaris OS for 64-bit Oracle, the filemap.ora file should contain the following setting:

```
lib=Veritas:/opt/VRTSdbed/lib/libvxoramap_64.so
```

- 2 After verifying that the system is using the Veritas library for Oracle storage mapping, set the file_mapping initialization parameter to true.

For example:

```
SQL> alter system set file_mapping=true;
```

The file_mapping initialization parameter is set to false by default. You do not need to shut down the instance to set this parameter. Setting file_mapping=true starts the FMON background process.

If you want storage mapping to be enabled whenever you start up an instance, set the file_mapping initialization parameter to true in the init.ora file.

Accessing dynamic performance views

Review the procedure to access dynamic performance views.

To access dynamic performance views

- 1 Confirm that the Veritas mapping library for Oracle file mapping has been enabled.

For example, enter the following SQL command:

```
SQL> select lib_idx idx, lib_name name, vendor_name vname, \
path_name path from v$map_library;
```

- 2 After storage mapping has been enabled, Oracle datafiles can be mapped using the DBMS_STORAGE_MAP package.

For more information about various features and capabilities of the DBMS_STORAGE_MAP package, see your Oracle documentation.

- 3 Use SQL commands to display the mapping information that is captured in Oracle’s dynamic performance views.

- To display the contents of v\$map_file for a Quick I/O file, enter the following SQL command:

```
SQL> select file_name name, file_map_idx idx, \
file_status status, file_type type, file_structure str, \
file_size fsize, file_nexts nexts from v$map_file;
```

- To display the contents of v\$map_file_extent, enter the following SQL command:

```
SQL> select * from v$map_file_extent;
```

- To display the contents of v\$map_element, enter the following SQL command:

```
SQL> select elem_idx idx, elem_name, elem_type type, \
elem_size, elem_nsubelem nsub, elem_descr, stripe_size \
from v$map_element;
```

IDX	ELEM_NAME	TYPE	ELEM_SIZE	NSUB	ELEM_DESCR	STRIPE_SIZE
0	/dev/vx/rdsk/PROD_dg/ora92MIRROR			12582912	MIRROR	0
1	vxvm:PROD_dg/ora92-01	CONCATENATED	12586455	1	VERITAS VOLUME	0
2	/dev/vx/rdmp/c2t5d0s4	PARTITION	17674902	1	HOST DEVICE	0
3	/dev/rdsk/c2t5d0s4	PARTITION	17674902	1	DEVICE	0
4	c2t5d0	DISK	17682084	0	DISK	0

- To display the contents of v\$map_subelement, enter the following SQL command:

```
SQL> select * from v$map_subelement;
```

CHILD_IDX	PARENT_IDX	SUB_NUM	SUB_SIZE	ELEM_OFFSET	SUB_FLAGS
1	0	0	12586455	0	0
2	1	0	12586455	0	0
3	2	0	17674902	0	0
4	3	0	17682084	7182	0

- To display all the elements within the I/O stack for a specific file, enter the following SQL command:

```
SQL> with fv as
2 (select file_map_idx, file_name from v$map_file
3 where file_name = '/ora92/dbs/system01.dbf')
4 select
5 fv.file_name, lpad(' ', 4 * (level - 1)) || \
    el.elem_name elem_name, el.elem_size, el.elem_type, \
    el.elem_descr
6 from
7 v$map_subelement sb, v$map_element el, fv,
8 (select unique elem_idx from v$map_file_io_stack io, fv
9 where io.file_map_idx = fv.file_map_idx) fs
10 where el.elem_idx = sb.child_idx
11 and fs.elem_idx = el.elem_idx
12 start with sb.parent_idx in
13 (select distinct elem_idx
14 from v$map_file_extent fe, fv
15 where fv.file_map_idx = fe.file_map_idx)
16 connect by prior sb.child_idx = sb.parent_idx;
```

FILE_NAME	ELEM_NAME	ELEM_SIZE	ELEM_TYPE	ELEM_DESCR
/ora92/dbs/system01.dbf	vxvm:PROD_dg/ora92-01	12586455	CONCATENATED	VERITAS VOLUME
/ora92/dbs/system01.dbf	/dev/vx/rdmp/c2t5d0s4	17674902	PARTITION	HOST DEVICE
/ora92/dbs/system01.dbf	/dev/rdsk/c2t5d0s4	17674902	PARTITION	DEVICE
/ora92/dbs/system01.dbf	c2t5d0	17682084	DISK	DISK

Troubleshooting SF Oracle RAC

This chapter includes the following topics:

- [About troubleshooting SF Oracle RAC](#)
- [Troubleshooting I/O fencing](#)
- [Troubleshooting CVM](#)
- [Troubleshooting the repository database](#)
- [Troubleshooting Database Dynamic Storage Tiering commands](#)
- [Troubleshooting CFS](#)
- [Troubleshooting VCSIPC](#)
- [Troubleshooting interconnects](#)
- [Troubleshooting Oracle](#)
- [Troubleshooting ODM](#)

About troubleshooting SF Oracle RAC

SF Oracle RAC contains several component products, and as a result can be affected by any issues with component products. The first step in case of trouble should be to identify the source of the problem. It is rare to encounter problems in SF Oracle RAC itself; more commonly the problem can be traced to setup issues or problems in component products.

Use the information in this chapter to diagnose the source of problems. Indications may point to SF Oracle RAC set up or configuration issues, in which case solutions

are provided wherever possible. In cases where indications point to a component product or to Oracle as the source of a problem, it may be necessary to refer to the appropriate documentation to resolve it.

Running scripts for engineering support analysis

Troubleshooting scripts gather information about the configuration and status of your cluster and its modules. The scripts identify package information, debugging messages, console messages, and information about disk groups and volumes. Forwarding the output of these scripts to Symantec Tech Support can assist with analyzing and solving any problems.

getdbac

The getdbac script gathers information about the SF Oracle RAC modules. The file `/tmp/vcsopslog.time_stamp.sys_name.tar.Z` contains the script's output.

To use the getdbac script, on each system enter:

```
# /opt/VRTSvcs/bin/getdbac -local
```

getcomms

The getcomms script gathers information about the GAB and LLT modules. The file `/tmp/commslog.time_stamp.tar` contains the script's output.

To use the getcomms script, on each system enter:

```
# /opt/VRTSgab/getcomms -local
```

hagetcf

The hagetcf script gathers information about the VCS cluster and the status of resources. The output from this script is placed in a tar file, `/tmp/vcsconf.sys_name.tar.gz`, on each cluster system.

To use the hagetcf script, on each system enter:

```
# /opt/VRTSvcs/bin/hagetcf
```

Troubleshooting tips

The following files and command output may be required to determine the source of a problem:

- [Oracle installation error log](#)
- [Veritas log file](#)

- OS system log
- GAB port membership

Oracle installation error log

This file contains errors that occurred during installation. It clarifies the nature of the error and exactly when it occurred during the installation.

To check the Oracle installation error log

- 1 Access the following file:

```
$ORACLE_BASE/oraInventory/logs/installActions<date_time>.log
```

- 2 Verify if there are any installation errors logged in this file, since they may prove to be critical errors.
- 3 If there are any installation problems, send this file to Tech Support. It is required for debugging the issue.

Veritas log file

The Veritas log file contains all actions performed by HAD.

To check the Symantec log files

- 1 Access the following file:

```
/var/VRTSvcS/log/engine_A.log
```

- 2 Verify if there are any CVM or PrivNIC errors logged in this file, since they may prove to be critical errors.
- 3 You can access the vxconfigd log file at:

```
/var/adm/ras/vxconfigd.log
```

There are additional log files pertaining to the agents for SF Oracle RAC components such as CVM and CFS in the /var/VRTSvcS/log directory, which are also helpful in diagnosing issues.

To check the agent log files for CVM:

```
# /var/VRTSvcS/log/CVMVolDg_A.log
```

To check the agent log files for CFS:

```
# /var/VRTSvcS/log/CFSMount_A.log
```

To check the agent log files for Oracle:

```
# /var/VRTSvcS/log/Oracle_A.log
```

OS system log

OS syslog files located at /var/adm/messages can provide valuable information for diagnosing problems.

GAB port membership

To check GAB port membership

Enter the following command:

```
# /sbin/gabconfig -a
```

Port b must exist on the local system.

The output resembles this information:

```
GAB Port Memberships
=====
Port a gen 4a1c0001 membership 01
Port b gen ada40d01 membership 01
Port d gen 40100001 membership 01
Port f gen f1990002 membership 01
Port h gen d8850002 membership 01
Port o gen f1100002 membership 01
Port v gen 1fc60002 membership 01
Port w gen 15ba0002 membership 01
```

Table 9-1 defines each GAB port's function.
For illustration of different GAB ports, See Figure 1-6 on page 29.

Table 9-1 GAB port function

Port	Function
a	This port is used for GAB.
b	This port is used for I/O fencing communications.
d	ODM uses this port when communicating with other ODM instances to support the file management features of Oracle Managed Files (OMF) cluster wide.

Table 9-1 GAB port function (*continued*)

Port	Function
f	CFS uses this port for GLM lock and metadata communication.
h	VCS uses this port. VCS communicates the status of resources running on each system to all systems in the cluster.
o	This port is used by the VCSMM driver or VCS Membership Module .
v	CVM uses this port for kernel-to-kernel communication.
w	vxconfigd configuration daemon (module for CVM) uses this port for messaging.

Troubleshooting I/O fencing

The following sections discuss troubleshooting the I/O fencing problems. Review the symptoms and recommended solutions.

SCSI reservation errors during bootup

When restarting a node of an SF Oracle RAC cluster, SCSI reservation errors may be observed such as:

```
date system name kernel: scsi3 (0,0,6) : RESERVATION CONFLICT
```

This message is printed for each disk that is a member of any shared disk group which is protected by SCSI-3 I/O fencing. This message may be safely ignored.

The vxfsthddw utility fails when SCSI TEST UNIT READY command fails

If you 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.

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 the new node or attempt to restart vxfen driver with the command:

```
# /etc/init.d/vxfen start
```

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.
```

See [“How vxfen driver checks for pre-existing split-brain condition”](#) on page 262.

How vxfen driver checks for pre-existing split-brain condition

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 coordinator disks.

To resolve actual and apparent potential split brain conditions

◆ Depending on the split brain condition that you encountered, do the following:

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

- 1 Physically verify that system 2 is down.
Verify the systems currently registered with the coordinator disks. Use the following command:

```
# vxfenadm -g 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 using the `vxfcntlclearpre` command.

See [“Clearing keys after split brain using vxfcntlclearpre command”](#) on page 264.

- 3 Make any necessary repairs to system 2.
- 4 Restart system 2.

Clearing keys after split brain using vxfcntlclearpre command

When you have encountered a split brain condition, use the `vxfcntlclearpre` command to remove SCSI-3 registrations and reservations on the coordinator disks as well as on the data disks in all shared disk groups.

See [“About vxfcntlclearpre utility”](#) on page 82.

Registered keys are lost on the coordinator disks

If the coordinator disks lose the keys that are registered, the cluster might panic when a split-brain occurs.

To refresh the missing keys

- ◆ Use the `vxfcntlswap` utility to replace the coordinator disks with the same disks. The `vxfcntlswap` utility registers the missing keys during the disk replacement.

See [“Refreshing lost keys on coordinator disks”](#) on page 90.

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 `vxfcntlswap` utility.

See [“About vxfcntlswap utility”](#) on page 84.

Review the following information to replace coordinator disk in the coordinator disk group, or to destroy a coordinator disk group.

Note: Replacing or removing coordinator disks requires all services be shut down.

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:

```
# /etc/init.d/vcsmm stop
```

- 4 Stop I/O fencing on each node:

```
# /etc/init.d/vxfen stop
```

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 vxfencoordg | grep flags: | grep coordinator
```

- If the coordinator attribute value is set to on, you must turn off this attribute for the coordinator disk group.

```
# vxdg -g vxfencoordg set coordinator=off
```

- Destroy the disk group.

```
# vxdg destroy vxfencoordg
```

- 7 Add the new disk to the node, initialize it as a VxVM disk, and add it to the `vxfencoordg` disk group.

See the *Veritas Storage Foundation for Oracle RAC Installation and Configuration Guide* for detailed instructions.

- 8 Test the recreated disk group for SCSI-3 persistent reservations compliance.

See [“Testing the coordinator disk group using `vxfcntlsthdw -c` option”](#) on page 74.

- 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:

```
# /etc/init.d/vxfen start
```

- 11 On each node, start the VCSMM driver:

```
# /etc/init.d/vcsmm start
```

- 12 If necessary, restart VCS on each node:

```
# hstart
```

The vxfenswap utility faults when echo or cat is used in .bashrc file

The vxfenswap utility faults when you use echo or cat to print messages in the .bashrc file for the nodes.

To recover the vxfenswap utility fault

- ◆ Verify whether the rcp or scp functions properly.

If the vxfenswap operation is unsuccessful, use the `vxfenswap -cancel` command if required to roll back any changes that the utility made.

See [“About vxfenswap utility”](#) on page 84.

Troubleshooting CVM

This section discusses troubleshooting CVM problems.

Shared disk group cannot be imported

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
galaxy	1	0	Joined: Master
nebula	0	1	Joined: Slave

This above output shows that CVM is healthy, with system galaxy as the CVM master.

If CVM is functioning correctly, then this message 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:

```
# /sbin/gabconfig -a
```

Port b must exist on the local system.

Error importing shared disk groups

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:

See [“Removing preexisting keys”](#) on page 83.

Unable to start CVM

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:

See [“Removing preexisting keys”](#) on page 83.

CVMVolDg not online even though CVMCluster is online

When the CVMCluster resource goes online, the shared disk groups are automatically imported. If the disk group import fails for some reason, the CVMVolDg resources fault. Clearing and offlining the CVMVolDg type resources does not fix the problem.

To resolve the resource issue

- 1 Fix the problem causing the import of the shared disk group to fail.
- 2 Offline the service group containing the resource of type CVMVolDg as well as the service group containing the CVMCluster resource type.
- 3 Bring the service group containing the CVMCluster resource online.
- 4 Bring the service group containing the CVMVolDg resource online.

Shared disks not visible

If the shared disks in `/proc/scsi/scsi` 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 looking at `/proc/scsi/fibre_channel_driver/*` files.

For example:

```

/proc/scsi/qla2xxx/2 contains...
...
SCSI LUN Information:
(Id:Lun) * - indicates lun is not registered with the OS.
( 0: 0): Total reqs 74, Pending reqs 0, flags 0x0, 0:0:84 00
( 0: 1): Total reqs 0, Pending reqs 0, flags 0x0*, 0:0:84 00
( 0: 2): Total reqs 0, Pending reqs 0, flags 0x0*, 0:0:84 00
( 0: 3): Total reqs 0, Pending reqs 0, flags 0x0*, 0:0:84 00
( 0: 4): Total reqs 0, Pending reqs 0, flags 0x0*, 0:0:84 00
...

```

The above example indicates that not all LUNs are discovered by SCSI. The problem might be corrected by specifying `dev_flags` or `default_dev_flags` and *max_luns* parameters for SCSI driver.

If the LUNs are not visible in `/proc/scsi/fibre_channel_driver/*` files, it may indicate a problem with SAN configuration or zoning.

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 if the disk is seen, using the following command:

```
# prtvtoc /dev/rdisk/cntndnsn
```

Troubleshooting the repository database

This section describes the `sfua_db_config` script commands that can be used to troubleshoot repository database issues.

sfua_db_config script command options

[Table 9-2](#) describes the `sfua_db_config` script command options. Use these command options when troubleshooting repository database issues.

Table 9-2 sfua_db_config script command options

Command option	Description
-ssh	Use this command option with the <code>sfua_db_config</code> command in a high availability (HA) configuration. This command option indicates that ssh and scp are to be used for communication between systems. Note: Either ssh or rsh should be preconfigured so that you can execute the commands without being prompted for passwords or confirmations. The default is rsh.
-o dropdb	The <code>sfua_db_config -o dropdb</code> command drops the repository database.
-o unconfig_cluster	The <code>sfua_db_config -o unconfig_cluster</code> command unconfigures the repository database from the VCS cluster.
-o dbstatus	The <code>sfua_db_config -o dbstatus</code> command verifies the status of the database and database server.
-o stopserver	The <code>sfua_db_config -o stopserver</code> command stops the database server.
-o startserver	The <code>sfua_db_config -o startserver</code> command starts the database server.
-o serverstatus	The <code>sfua_db_config -o serverstatus</code> command reports the status of the database server.
-o stopdb	The <code>sfua_db_config -o stopdb</code> command detaches the repository database from the database server.
-o startdb	The <code>sfua_db_config -o startdb</code> command attaches the repository database to the database server.

Switching the repository database from one node to another

Sometimes, it is helpful to switch the high availability repository database from one node to another.

One way of switching the repository is as follows:

Take the repository database offline on one node:

```
# hagrps -offline Sfua_Base -sys nebula
```

Bring the repository database online on the other node:

```
# hagrps -online Sfua_Base -sys galaxy
```

Troubleshooting Database Dynamic Storage Tiering commands

If the Database Dynamic Storage Tiering commands fail as in the following example, then review the `tsdb_debug.log`.

The `tsdb_debug.log` is located at:

```
/var/vx/vxdba/logs/tsdb_debug.log
```

For example, when the following message appears after issuing a `dbdst_addvol` command review the `tsdb_debug.log`:

```
$ /opt/VRTS/bin/dbdst_addvol -S BLM21 -M /dev/vx/dsk/nobody_data1/\
data1vol -v new_vol1:fast_storage,new_vol2:slow_storage,new_vol3:slow_storage
```

```
SFORA dbdst_addvol ERROR V-81-6222 Could not add volume new_vol1 to vset
```

After accessing and reviewing the `tsdb_debug.log`, the following information appears:

```
snip /var/vx/vxdba/logs/tsdb_debug.log>
1216606 Tue May 13 10:11:05 2008
/opt/VRTS/bin/dbdst_addvol -S BLM21 -M /dev/vx/dsk/nobody_data1/data1vol -v
new_vol1:fast_storage,new_vol2:slow_storage,
new_vol3:slow_storage
1216606 Tue May 13 10:11:09 2008
RACmaster = editor
1216606 Tue May 13 10:11:09 2008
editor:/opt/VRTSdbcom/.dba/tsdb_setup.sh -g nobody_data1 -o addvol -d\
data1vol -v new_vol1 -m /oradata1 -t vxfs.placement_class.FAST_STORAGE
1216606 Tue May 13 10:11:09 2008
command failed, ret=1

1216606 Tue May 13 10:11:09 2008
tsdb_setup.sh arguments -g nobody_data1 -o addvol -d data1vol -v
new_vol1 -m /or
adata1 -t vxfs.placement_class.FAST_STORAGE
05/13/08@17:08:11
size of volume new_vol1 is 204800
VxVM vxvset ERROR V-5-1-10035 Volume set data1vol contains volume(s)
in snapshot chain.
```

This can cause inconsistencies in the snapshot hierarchy. Specify "-f" option to force the operation.

^^^ NOTE: here is the reason for the failure, barried in this log file:

^^^ /var/vx/vxdba/logs/tsdb_debug.log

Can not add to data1vol, ERR 1

ERROR:1

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, /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
```

Troubleshooting VCSIPC

This section discusses troubleshooting VCSIPC problems.

VCSIPC errors in Oracle trace/log files

If you see any VCSIPC errors in the Oracle trace/log files, check the /var/adm/messages file for any LMX error messages.

If you see messages that contain any of the following:

```
. . . out of buffers
. . . out of ports
. . . no minors available
```

See [“About LMX tunable parameters”](#) on page 290.

If you see any VCSIPC warning messages in Oracle trace/log files that resemble:

```
connection invalid
```

or,

```
Reporting communication error with node
```

Check whether the Oracle Real Application Cluster instance on the other system is still running or has been restarted. The warning message indicates that the VCSIPC/LMX connection is no longer valid.

Troubleshooting interconnects

This section discusses troubleshooting interconnect problems.

Restoring communication between host and disks after cable disconnection

If a Fibre cable is inadvertently disconnected between the host and a disk, you can restore communication between the host and the disk without restarting.

To restore lost cable communication between host and disk

- 1 Reconnect the cable.
- 2 Use the `format` command to verify that the host sees the disks.
It may take a few minutes before the host is capable of seeing the disk.
- 3 Issue the following `vxdctl` command to force the VxVM configuration daemon `vxconfigd` to rescan the disks:

```
# vxdctl enable
```

Troubleshooting Oracle

This section discusses troubleshooting Oracle.

Oracle user must be able to read /etc/littab File

Check the permissions of the file `/etc/littab`. Oracle must be allowed to read it.

Error when starting an Oracle instance

If the VCSMM driver (the membership module) is not configured, an error displays while starting the Oracle instance that resembles:

```
ORA-29702: error occurred in Cluster Group Operation
```

To start the VCSMM driver, enter the following command:

```
# /etc/init.d/vcsmm start
```

The command included in the `/etc/vcsmmtab` file enables the VCSMM driver to be started at system boot.

Oracle log files show shutdown called even when not shutdown manually

The Oracle enterprise agent calls shutdown if monitoring of the Oracle/Netlsnr resources fails. On all cluster nodes, review the following VCS and Oracle agent log files for any errors or status:

```
/var/VRTSvcs/log/engine_A.log
/var/VRTSvcs/log/Oracle_A.log
```

DBCA fails while creating database

Verify that the `hostname -i` command returns the public IP address of the current node. This command is used by the installer and the output is stored in the OCR. If `hostname -i` returns 127.0.0.1, it causes the DBCA to fail.

CRS processes fail to startup

Verify that the correct private IP address is configured on the private link using the `PrivNIC` agent. Check the CSS log files to learn more. You can find the CSS log files at `$CRS_HOME/log/node_name/cssd/*`

Consult the Oracle CRS documentation for more information.

CRS fails after restart

If the CRS fails to start after boot up, check for the occurrence of the following strings in the `/var/log/` messages file.

String value in the file:

```
Oracle CSSD failure. Rebooting for cluster integrity
```

CRS may fail due to Oracle CSSD failure. The Oracle CSSD failure may be caused by one of the following events:

- Communication failure occurred and CRS fenced out the node.
- OCR and Vote disk became unavailable.
- `ocssd` was killed manually and on restarting from `inittab` it rebooted the cluster.
- Killing the `init.cssd` script.

String value in the file:

```
Waiting for file system containing
```

The CRS installation is on a shared disk and the `init` script is waiting for that file system to be made available.

String value in the file:

```
Oracle Cluster Ready Services disabled by corrupt install
```

The following file is not available or has corrupt entries:

```
/etc/oracle/scls_scr/hostname/root/crsstart.
```

String value in the file:

```
OCR initialization failed accessing OCR device
```

The shared file system containing the OCR is not available and CRS is waiting for it to become available.

Removing Oracle CRS if installation fails

The following procedure provides instructions to remove Oracle CRS. Make sure that you consult the Oracle CRS documentation for complete steps.

To remove Oracle CRS

- 1 Run the `rootdelete.sh` script (in this example, `$CRS_HOME` is `'/crshome'`):

```
# cd /crshome/install
# ./rootdelete.sh
```

Run the `rootdeinstall.sh` script:

```
# cd /crshome/install
# ./rootdeinstall.sh
```

- 2 Stop the applications on all nodes:

```
# srvctl stop nodeapps -n node_name
```

- 3 Copy the file `inittab.orig` back to the name and remove other `init` files:

```
# cd /etc

# cp inittab.orig inittab

# rm init.crs init.crsd init.cssd init.evmd

# rm /etc/rc.d/rc2.d/K96init.crs

# rm /etc/rc.d/rc2.d/S96init.crs
```

- 4 Remove `ora*` files from the `/etc` directory:

```
# cd /etc

# rm -r oraInst.loc ora_save* oracle oratab
```

- 5 Remove file from `$CRS_HOME` and Oracle Inventory after taking a backup of your current files.

If your `$CRS_HOME` is located at `'/crshome'`, perform the following steps:

```
# cd /crshome

# mkdir crs.old

# mv * crs.old
```

If your `$CRS_HOME` is located at `'/crshome/crs'`, perform the following steps:

```
# cd /crshome

# mv crs crs.old
```

You can remove the `'crs.old'` directory at a later time.

- 6 Remove files from the OCR and Voting disk directories. For our example:

```
# rm /ocrvote/ocr
# rm /ocrvote/vote-disk
```

If OCR and Voting disk storage are on raw volumes, use command resembling:

```
# dd if=/dev/zero of=/dev/vx/rdisk/ocrvotedg/ocrvol bs=8192 \
count=18000

# dd if=/dev/zero of=/dev/vx/rdisk/ocrvotedg/votvol bs=8192 \
count=3000
```

- 7 Reboot the systems to make sure no CRS daemons are running.

For instructions on reinstalling CRS at a later time, see the *Veritas Storage Foundation for Oracle RAC Installation and Configuration Guide*.

Troubleshooting the Virtual IP (VIP) Configuration

When troubleshooting issues with the VIP configuration, use the following commands and files:

- Check for network problems on all nodes:
`/etc/ifconfig -a`
- Make sure the virtual host name is registered with the DNS server:
`/usr/bin/nslookup virtual_host_name`
- Verify the `/etc/hosts` file on each node.
- Check the output of the following command:
`$CRS_HOME/bin/crs_stat`
- On the problem node, use the command:
`$ srvctl start nodeapps -n node_name`
This command works only if the virtual IP address is plumbed.

OCR and Vote disk related issues

Verify that the permissions are set appropriately as given in the Oracle installation guide.

See [“CRS fails after restart”](#) on page 275.

Troubleshooting ODM

This section discusses troubleshooting ODM.

File System configured incorrectly for ODM shuts down Oracle

Review the instructions on creating the link and confirming that Oracle uses the libraries. Shared file systems in RAC clusters without ODM libraries linked to Oracle RAC 9i may exhibit slow performance and are not supported.

If ODM cannot find the resources it needs to provide support for cluster file systems, it does not allow Oracle to identify cluster files and causes Oracle to fail at startup.

To verify cluster status, run the following command and review the output:

```
# cat /dev/odm/cluster

cluster status: enabled
```

If the status is "enabled," ODM is supporting cluster files. Any other cluster status indicates that ODM is not supporting cluster files. Other possible values include:

pending	ODM cannot yet communicate with its peers, but anticipates being able to eventually.
failed	ODM cluster support has failed to initialize properly. Check console logs.
disabled	ODM is not supporting cluster files. If you think ODM should be supporting the cluster files, check: ■ /dev/odm mount options in /etc/vfstab. If the <code>nocluster</code> option is being used, it can force the disabled cluster support state. ■ Make sure the <code>VRTSgms</code> (group messaging service) package is installed.

If /dev/odm is not mounted, no status can be reported.

Prevention and recovery strategies

This chapter includes the following topics:

- [Prevention and recovery strategies](#)

Prevention and recovery strategies

The following topics are useful diagnostic tools and strategies for preventing and recovering from the various problems that can occur in the SF Oracle RAC environment.

Verification of GAB ports in SF Oracle RAC cluster

The following 8 ports need to be up on all the nodes of SF Oracle RAC cluster:

- GAB
- I/O fencing
- ODM
- CFS
- VCS ('had')
- vcsmm (membership module for SF Oracle RAC)
- CVM (kernel messaging)
- CVM (vxconfigd)

The following command can be used to verify the state of GAB ports:

```
# gabconfig -a
```

GAB Port Memberships

```
Port a gen 7e6e7e05 membership 01
Port b gen 58039502 membership 01
Port d gen 588a7d02 membership 01
Port f gen 1ea84702 membership 01
Port h gen cf430b02 membership 01
Port o gen de8f0202 membership 01
Port v gen db411702 membership 01
Port w gen cf430b02 membership 01
```

The data indicates that all the GAB ports are up on the cluster having nodes 0 and 1.

For more information on the GAB ports in SF Oracle RAC cluster, see the *Veritas Storage Foundation for Oracle RAC Installation and Configuration Guide*.

Examining GAB seed membership

The number of systems that participate in the cluster is specified as an argument to the `gabconfig` command in `/etc/gabtab`. In the following example, two nodes are expected to form a cluster:

```
# cat /etc/gabtab

/sbin/gabconfig -c -n2
```

GAB waits until the specified number of nodes becomes available to automatically create the port “a” membership. Port “a” indicates GAB membership for an SF Oracle RAC cluster node. Every GAB reconfiguration, such as a node joining or leaving increments this seed membership in every cluster member node.

A sample port ‘a’ membership as seen in `gabconfig -a` is shown:

```
Port a gen 7e6e7e01 membership 01
```

In this case, 7e6e7e01 indicates the “membership generation number” and 01 corresponds to the cluster’s “node map”. All nodes present in the node map reflects the same membership ID as seen by the following command:

```
# gabconfig -a | grep "Port a"
```

The semi-colon is used as a placeholder for a node that has left the cluster. In the following example, node 0 has left the cluster:

```
# gabconfig -a | grep "Port a"

Port a gen 7e6e7e04 membership ;1
```

When the last node exits the port “a” membership, there are no other nodes to increment the membership ID. Thus the port “a” membership ceases to exist on any node in the cluster.

When the last and the final system is brought back up from a complete cluster cold shutdown state, the cluster will seed automatically and form port “a” membership on all systems. Systems can then be brought down and restarted in any combination so long as at least one node remains active at any given time.

The fact that all nodes share the same membership ID and node map certifies that all nodes in the node map participates in the same port “a” membership. This consistency check is used to detect “split-brain” and “pre-existing split-brain” scenarios.

Split-brain occurs when a running cluster is segregated into two or more partitions that have no knowledge of the other partitions. The pre-existing network partition is detected when the “cold” nodes (not previously participating in cluster) start and are allowed to form a membership that might not include all nodes (multiple sub-clusters), thus resulting in a potential split-brain.

Manual GAB membership seeding

It is possible that one of the nodes does not come up when all the nodes in the cluster are restarted, due to the “minimum seed requirement” safety that is enforced by GAB. Human intervention is needed to safely determine that the other node is in fact not participating in its own mini-cluster.

The following should be carefully validated before manual seeding, to prevent introducing split-brain and subsequent data corruption:

- Verify that none of the other nodes in the cluster have a port “a” membership
- Verify that none of the other nodes have any shared disk groups imported
- Determine why any node that is still running does not have a port “a” membership

Run the following command to manually seed GAB membership:

```
# gabconfig -cx
```

Refer to `gabconfig (1M)` for more details.

Evaluating VCS I/O fencing ports

I/O Fencing (VxFEN) uses a dedicated port that GAB provides for communication across nodes in the cluster. You can see this port as port ‘b’ when `gabconfig -a` runs on any node in the cluster. The entry corresponding to port ‘b’ in this

membership indicates the existing members in the cluster as viewed by I/O Fencing.

GAB uses port "a" for maintaining the cluster membership and must be active for I/O Fencing to start.

To check whether fencing is enabled in a cluster, the '-d' option can be used with `vxfenadm (1M)` to display the I/O Fencing mode on each cluster node. Port "b" membership should be present in the output of `gabconfig -a` and the output should list all the nodes in the cluster.

If the GAB ports that are needed for I/O fencing are not up, that is, if port "a" is not visible in the output of `gabconfig -a` command, LLT and GAB must be started on the node.

The following commands can be used to start LLT and GAB respectively:

To start LLT on each node:

```
# /etc/init.d/llt start
```

If LLT is configured correctly on each node, the console output displays:

```
LLT INFO V-14-1-10009 LLT Protocol available
```

To start GAB, on each node:

```
# /etc/init.d/gab start
```

If GAB is configured correctly on each node, the console output displays:

```
GAB INFO V-15-1-20021 GAB available
```

```
GAB INFO V-15-1-20026 Port a registration waiting for seed port  
membership
```

Verifying normal functioning of VCS I/O fencing

It is mandatory to have VCS I/O fencing enabled in SF Oracle RAC cluster to protect against split-brain scenarios. VCS I/O fencing can be assumed to be running normally in the following cases:

- Fencing port 'b' enabled on both nodes

```
# gabconfig -a
```

- Registered keys present on the coordinator disks

```
# vxfenadm -g all -f /etc/vxfentab
```

Managing SCSI-3 PGR keys in SF Oracle RAC cluster

I/O Fencing places the SCSI-3 PGR keys on coordinator LUNs. The format of the key follows the naming convention wherein ASCII “A” is prefixed to the LLT ID of the system that is followed by 7 dash characters.

For example:

node 0 uses A-----

node 1 uses B-----

In an SF Oracle RAC/SF CFS/SF HA environment, VxVM/CVM registers the keys on data disks, the format of which is ASCII “A” prefixed to the LLT ID of the system followed by the characters “PGRxxxx” where ‘xxxx’ = i such that the disk group is the ith shared group to be imported.

For example: node 0 uses APGR0001 (for the first imported shared group).

In addition to the registration keys, VCS/CVM also installs a reservation key on the data LUN. There is one reservation key per cluster as only one node can reserve the LUN.

See [“About SCSI-3 Persistent Reservations”](#) on page 41.

The following command lists the keys on a data disk group:

```
# vxdbg list |grep data
```

```
galaxy_data1 enabled,shared,cds 1201715530.28.pushover
```

Select the data disk belonging to galaxy_data1:

```
# vxdisk -o alldgs list |grep galaxy_data1
```

```
clt2d0s2 auto:cdsdisk    clt2d0s2          galaxy_data1 online shared
clt2d1s2 auto:cdsdisk    clt2d1s2          galaxy_data1 online shared
clt2d2s2 auto:cdsdisk    clt2d2s2          galaxy_data1 online shared
```

The following command lists the PGR keys:

```
# vxdisk -o listreserve list clt2d0s2
```

```
.....
```

```
.....
```

```
Multipathing information:
```

```
numpaths: 1
```

```
hdisk6 state=enabled
```

```
Reservations:
```

```
BPGR0000 (type: Write Exclusive Registrants Only, scope: LUN(0x0))
```

```
2 registered pgr keys
```

```
BPGR0004
APGR0004
```

Alternatively, the PGR keys can be listed using `vxfenadm` command:

```
# echo "/dev/vx/dmp/clt2d0s2" > /tmp/disk71

# vxfenadm -g all -f /tmp/disk71

Device Name: /dev/vx/dmp/clt2d0s2
Total Number Of Keys: 2
key[0]:
    Key Value [Numeric Format]: 66,80,71,82,48,48,48,52
    Key Value [Character Format]: BPGR0004
key[1]:
    Key Value [Numeric Format]: 65,80,71,82,48,48,48,52
    Key Value [Character Format]: APGR0004
```

Evaluating the number of SCSI-3 PGR keys on a coordinator LUN, if there are multiple paths to the LUN from the hosts

The utility `vxfenadm` (1M) can be used to display the keys on the coordinator LUN. The key value identifies the node that corresponds to each key. Each node installs a registration key on all the available paths to the LUN. Thus, the total number of registration keys is the sum of the keys that are installed by each node in the above manner.

See [“About vxfenadm utility”](#) on page 80.

Detecting accidental SCSI-3 PGR key removal from coordinator LUNs

The keys currently installed on the coordinator disks can be read using the following command:

```
# vxfenadm -g all -f /etc/vxfentab
```

There should be a key for each node in the operating cluster on each of the coordinator disks for normal cluster operation.

Identifying a faulty coordinator LUN

The utility `vxfsentshdw` (1M) provided with I/O Fencing can be used to identify faulty coordinator LUNs. This utility must be run from any two nodes in the cluster. The coordinator LUN, which needs to be checked, should be supplied to the utility.

See “[About vxfcntl utility](#)” on page 71.

Replacing a faulty coordinator disk without stopping I/O Fencing

To replace a disk in a coordinator disk group when the cluster is online

- 1 Import the coordinator disk group using vxldg command.

```
# vxldg -tfc import `cat /etc/vxfendg`
```

- 2 Remove the faulty disk (c1t2d1s2 in this example) from the coordinator disk group and add the new disk (c1t2d3s2 in this example).

```
# vxfcntl -l
```

I/O Fencing Configuration Information:

```
=====
```

```
Single Disk Flag      : 0
Count                 : 3
Disk List
Disk Name              Major Minor  Serial Number      Policy
/dev/vx/rdmp/c1t2d0s2   45     8      50002AC0006E028C    dmp
/dev/vx/rdmp/c1t2d1s2   45    10      50002AC00072028C    dmp
/dev/vx/rdmp/c1t2d2s2   45    12      50002AC0006D028C    dmp
```

- 3 Verify that the coordinator flag is set:

```
# vxldg list vxfcntlddg |grep flags | grep coordinator \
flags: cds coordinator
#
```

- 4 If the disk group was created with the coordinator flag set, the flag must be temporarily reset before going to the next step:

```
# vxldg -g vxfcntlddg set coordinator=off
```

- 5 Now verify that the coordinator flag is off:

```
# vxldg list vxfcntlddg | grep flags: | grep coordinator
```

- 6 Remove the existing old disks and add new disks:

```
# vxldg -g vxfcntlddg rmdisk c1t2d1s2
```

```
# vxldg -g vxfcntlddg adddisk c1t2d3s2
```

- 7 From one of the nodes, run the vxfcntl utility:

```
# /opt/VRTSvcs/vxfen/bin/vxfencfg -g vxfcntlddg
```

- 8 If the coordinator flag was reset in the second step, set it using:

```
# vxdg -g vxfencoorddg set coordinator=on
```

- 9 Verify that the coordinator flag is set:

```
# vxdg list vxfencoorddg |grep flags | grep coordinator \
flags: cds coordinator
#
```

- 10 The output of `vxfenconfig -l` confirms that the replacement of the disk is complete.

```
# vxfenconfig -l
```

I/O Fencing Configuration Information:

```
Single Disk Flag      : 0
Count                 : 3
Disk List
Disk Name              Major   Minor   Serial Number Policy
/dev/vx/rdmp/c1t2d0s2  45      8      50002AC0006E028C dmp
/dev/vx/rdmp/c1t2d3s2  45     10     50002AC00033028C dmp
/dev/vx/rdmp/c1t2d2s2  45     12     50002AC0006D028C dmp
```

- 11 Deport the coordinator disk group:

```
# vxdg deport `cat /etc/vxfendg`
```

Collecting I/O Fencing kernel logs

I/O Fencing kernel logs contain useful information to troubleshoot intricate I/O fencing issues. The logs can be collected using the following command:

```
# /opt/VRTSvcs/vxfen/bin/vxfendebg -p
```


Tunable parameters

This chapter includes the following topics:

- [About SF Oracle RAC tunable parameters](#)
- [About LMX tunable parameters](#)
- [About VXFEN tunable parameters](#)
- [Configuring the VXFEN module parameters](#)

About SF Oracle RAC tunable parameters

Tunable parameters can be configured to enhance the performance of specific SF for Oracle RAC features. This chapter discusses how to configure the following SF Oracle RAC tunables:

- LMX
- VXFEN

Symantec recommends that the user not change the tunable kernel parameters without assistance from Veritas support personnel. Several of the tunable parameters preallocate memory for critical data structures, and a change in their values could increase memory use or degrade performance.

Warning: Do not adjust the SF Oracle RAC tunable parameters for LMX and VXFEN as described below to enhance performance without assistance from Veritas support personnel.

About LMX tunable parameters

The section describes the LMX tunable parameters and how to reconfigure the LMX module.

Solaris OS LMX tunable parameters

To change the tunable global parameter values for the LMX driver on a node, edit the file `/kernel/drv/lmx.conf`.

Table 11-1 describes the LMX driver tunable parameters.

Table 11-1 LMX tunable parameters

LMX parameter	Default value	Maximum value	Description
contexts (minors)	8192	65535	Specifies the maximum number of contexts system-wide. Each Oracle process typically has two LMX contexts. "Contexts" and "minors" are used interchangeably in the documentation; "context" is an Oracle-specific term and should be used to specify the value in the <code>lmx.conf</code> file.
ports	4096	65535	Specifies the number of communication endpoints for transferring messages from the sender to the receiver in a uni-directional manner.
buffers	4096	65535	Specifies the number of addressable regions in memory to which LMX data can be copied.

If you see the message "no minors available" on one node, you can edit the file `/kernel/drv/lmx.conf` and add a configuration parameter increasing the value for the maximum number of contexts.

Note: The error message may contain the term "minors," but you must use the term "contexts" when changing the parameter value.

Warning: Increasing the number of contexts on a specific system has some impact on the resources of that system.

To reconfigure the LMX module

This section discusses how to reconfigure the LMX module on a node. In this example, configuring `contexts=16384` allows a maximum of 8192 Oracle processes ($8192 * 2 = 16384$). Note in the following output that double-quotes are not used to specify an integer value.

```
#
# LMX configuration file
#
name="lmx" parent="pseudo" contexts=16384 instance=0;
```

For the changes in the `/kernel/drv/lmx.conf` file to take effect, either restart the system or reconfigure the LMX module as described in the following procedure.

To reconfigure the LMX modules

- 1 Shut down all Oracle service groups on the system:

```
# hagr -offline oragrp -sys galaxy
```

- 2 Stop all Oracle client processes on the system, such as `sqlplus` and `svrmgrl`.
- 3 Unconfigure the LMX module:

```
# /sbin/lmxconfig -U
```

- 4 Determine the LMX module ID:

```
# /usr/sbin/modinfo | grep -I lmx
```

The module ID is the number in the first column of the output.

- 5 Unload the LMX module, using the module ID you determined:
- # /usr/sbin/modunload -I *module_ID*
- 6 For a system running Solaris 10, run the `update_drv` command to re-read the `/kernel/drv/lmx.conf` file.
- # /usr/sbin/update_drv lmx

Note: The `modunload` command has often been used on driver modules to force the system to reread the associated driver configuration file. While this procedure and command works in Solaris 9, this behavior may fail in later releases. The supported method for rereading the driver configuration file for systems running Solaris 10 is through the `update_drv` command. For additional information, refer to `update_drv(1M)`.

- 7 Configure the LMX module:
- # /sbin/lmxconfig -c
- 8 Bring the service groups back online:
- # hagrps -online oragrp -sys galaxy

About VXFEN tunable parameters

The section describes the VXFEN tunable parameters and how to reconfigure the VXFEN module.

[Table 11-2](#) describes the tunable parameters for the VXFEN driver.

Table 11-2 VXFEN tunable parameters

vxfen Parameter	Description and Values: Default, Minimum, and Maximum
vxfen_debug_sz	Size of debug log in bytes ■ Values Default: 65536 Minimum: 65536 Maximum: 256K

Table 11-2 VXFEN tunable parameters (*continued*)

vxfen Parameter	Description and Values: Default, Minimum, and Maximum
vxfen_max_delay	Specifies the maximum number of seconds that the smaller sub-cluster waits before racing with larger sub-clusters for control of the coordinator disks when a split brain occurs. This value must be greater than the vxfen_min_delay value. ■ Values Default: 60 Minimum: 0 Maximum: 600
vxfen_min_delay	Specifies the minimum number of seconds that the smaller sub-cluster waits before racing with larger sub-clusters for control of the coordinator disks when a split brain occurs. This value must be smaller than the vxfen_max_delay value. ■ Values Default: 1 Minimum: 0 Maximum: 600

In the event of a network partition, the smaller sub-cluster delays before racing for the coordinator disks. The time delayed allows a larger sub-cluster to win the race for the coordinator disks. The vxfen_max_delay and vxfen_min_delay parameters define the delay in seconds.

Configuring the VXFEN module parameters

The following section discusses the procedures for configuring and reconfiguring the VXFEN modules, after adjusting the tunable kernel driver parameters.

To reconfigure the VXFEN module

On each node, edit the file /kernel/drv/vxfen.conf to change the value of the vxfen driver tunable global parameters, vxfen_max_delay and vxfen_min_delay.

Note: You must restart the system to put change into effect.

The following VXFEN example displays the content of the default file /kernel/drv/vxfen.conf before changing the vxfen_max_delay parameter:

```
#
# VXFEN configuration file
#
name="vxfen" parent="pseudo" instance=0 dbg_log_size=65536
vxfen_max_delay=60 vxfen_min_delay=1;
```

After editing the file to change the min_delay value to 30, the default file /kernel/drv/vxfen.conf contains the following values:.

```
#
# VXFEN configuration file
#
name="vxfen" parent="pseudo" instance=0 dbg_log_size=65536
vxfen_max_delay=60 vxfen_min_delay=30;
```

After reviewing the default file edits, close and save the file.

For the changes to take effect, either restart the system, or reconfigure the VXFEN module as described in the following procedure.

To reconfigure the VXFEN module

- 1 Shut down all Oracle service groups on the node:

```
# hagrps -offline oragrp -sys galaxy
```

- 2 Stop all Oracle client processes, such as sqlplus, svrmgr1, and gsd, on the node.

- 3 Unconfigure the VXFEN module:

```
# /sbin/vxfenconfig -U
```

- 4 Determine the VXFEN module ID:

```
# /usr/sbin/modinfo | grep -I vxfen
```

The module ID is the number in the first column of the output.

- 5 Unload the VXFEN module, using the module ID you determined:

```
# /usr/sbin/modunload -I module_ID
```

- 6 For a system running Solaris 10, run the `update_drv` command to re-read the `/kernel/drv/vxfen.conf` file.

```
# /usr/sbin/update_drv vxfen
```

Note: The `modunload` command has often been used on driver modules to force the system to reread the associated driver configuration file. While this procedure and command works in Solaris 9, this behavior may fail in later releases. The supported method for rereading the driver configuration file for systems running Solaris 10 is through the `update_drv` command. For additional information, refer to `update_drv(1M)`.

- 7 Configure the VXFEN module:

```
# /sbin/vxfenconfig -c
```

- 8 Bring the service groups back online:

```
# hagrps -online oragrp -sys galaxy
```


Reference

- [Database FlashSnap status information](#)
- [Using third party software to back up files](#)
- [Error messages](#)

Database FlashSnap status information

This appendix includes the following topics:

- [About Database FlashSnap status information](#)
- [Database FlashSnap status information from the GUI](#)
- [Database FlashSnap Snapshot status information from the CLI](#)

About Database FlashSnap status information

Veritas Database FlashSnap functionality provides the following information for the various snapplan stages and snapshot procedures:

- Snapshot status information
- Snapshot database status information

You can view this information using either the command line interface (CLI) or the GUI.

For additional information about Database FlashSnap GUI functionality, see the *Veritas Storage Foundation for Database Graphical User Interface Guide*

Database FlashSnap status information from the GUI

You can obtain both the snapshot status and the database status from the GUI. The tables in this section provide detailed information regarding the various status values.

Snapshot status information from the GUI

To view snapshot status information from the GUI, click on a specific snapplan in the object tree. The snapshot status can be seen on the right side of the window in the Snapplan State field.

[Table A-1](#) displays information regarding the various snapshot status values.

Note: The values in the Snapshot status column below are displayed as seen in the Snapplan State field of the GUI.

Note: SF Oracle RAC does not support Database FlashSnap reverse resynchronization.

Table A-1 Snapshot status information from the GUI

Snapshot status	Complete operations	Allowed operations
init_full	■ Modify/Validate Snapplan (successful) ■ Resync Snapshot (successful)	Create snapshot
snapshot_start	Create Snapshot (failed)	If the Create Snapshot operation failed, contact your system administrator for help. You can use the VxVM utilities to create a snapshot and resynchronize the snapshot volumes, then use the Create Snapshot operation with the Force snapshot creation option for the subsequent snapshot.
snapshot_end	Create Snapshot (successful)	■ Resync Snapshot ■ Create Snapshot Database with the Create database option

Table A-1 Snapshot status information from the GUI (*continued*)

Snapshot status	Complete operations	Allowed operations
resync_start	Resync Snapshot (failed)	If the Resync Snapshot operation failed, contact your system administrator for help. You can use the VxVM utilities to resynchronize the snapshot volumes, then use the Create Snapshot operation with the Force snapshot creation option for the subsequent snapshot.
restartdb_start	Start Up Snapshot Database with the Restart database option (failed)	Try to start the snapshot database manually.
restartdb_end	Create Snapshot Database with the Restart database option (successful)	Shut Down Database with the unmount option
mountdb_start	dbed_vmclonedb -o mountdb command failed from the CLI Note: This option is not supported in the GUI.	Recover the snapshot database manually, then run the dbed_vmclonedb -o update_status command from the CLI Note: This option is not supported in the GUI.
mountdb_end	dbed_vmclonedb -o mountdb command from the CLI was successful Note: This option is not supported in the GUI.	■ Umount Database FlashSnap ■ dbed_vmclonedb -o update_status command from the CLI Note: This option is not supported in the GUI.
recoverdb_start	Create Snapshot Database with the Restart database option (failed).	Recover the snapshot database manually, then run the dbed_vmclonedb -o update_status command from the CLI Note: This option is not supported in the GUI.

Table A-1 Snapshot status information from the GUI (continued)

Snapshot status	Complete operations	Allowed operations
recoverdb_end	Create Snapshot Database with the Restart database option (successful)	Shut Down Database with the umount option
umount_start	dbed_vmclonedb -o umount command failed from the CLI	Verify that your file system(s) are not busy and retry the command.
umount_end	dbed_vmclonedb -o umount command from the CLI was successful Note: This option is not supported in the GUI.	■ Start Up Snapshot Database with the restart database option ■ Resync Snapshot

Snapshot database status information from the GUI

To view the Snapshot database status information from the GUI, click on a specific snapplan in the object tree. The Snapshot database status can be seen on the right side of the window in the Database Status field.

Table A-2 below displays information regarding the various status values.

Note: The values in the Snapshot database status column below are displayed as seen in the Snapplan State field of the GUI.

Note: SF Oracle RAC does not support Database FlashSnap reverse resynchronization.

Table A-2 Snapshot database status information from the GUI

Database status	Completed operations
init	■ Modify/Validate Snapplan (successful) ■ Create Snapshot (successful)
init Db	■ Modify/Validate Snapplan (successful) ■ Create Snapshot (successful)
database_recovered	Start up the Snapshot Database with the startup database option (successful)

Table A-2 Snapshot database status information from the GUI (continued)

Database status	Completed operations
database_recovered	Create Snapshot Database with the Create database option (successful)

Database FlashSnap Snapshot status information from the CLI

To view snapshot status information from the command line interface (CLI), use the `dbed_vmchecksnap` command with the `-o list` option to list all available snapplans for a specified database. Snapshot status information is displayed in the command output under the column heading `SNAP_STATUS`.

Note: The snapshot status and snapshot database status information may also appear in error messages.

Snapshot status information from the CLI

Table A-3 shows detailed information about each snapshot status (`SNAP_STATUS`) value.

Note: SF Oracle RAC does not support Database FlashSnap reverse resynchronization.

Table A-3 Snapshot status information from the CLI

SNAP_STATUS	Completed operations	Allowed operations
init_full	- dbed_vmchecksnap -o validate (successful) - dbed_vmsnap -o resync (successful)	dbed_vmsnap -o snapshot
init_db	dbed_vmchecksnap -o validate -f snapplan (failed)	Ensure that your storage configuration has been set up correctly.

Table A-3 Snapshot status information from the CLI (*continued*)

SNAP_STATUS	Completed operations	Allowed operations
snapshot_start	dbed_vmsnap -o snapshot (failed)	Contact your system administrator for help. Use Veritas Volume Manager commands to resynchronize the snapshot volumes, and use dbed_vmsnap -o snapshot -F to force snapshot creation.
snapshot_end	■ dbed_vmsnap -o snapshot (successful)	■ dbed_vmsnap -o resync ■ dbed_vmclonedb -o mount mountdb recoverdb
snapshot_vol_start snapshot_vol_end resync_dg_start resync_dg_end	dbed_vmsnap -o snapshot (failed)	Re-run dbed_vmsnap -o snapshot
resync_vol_start resync_vol_end snapshot_dg_start snapshot_dg_end	dbed_vmsnap -o resync (failed)	Re-run dbed_vmsnap -o resync
resync_start	dbed_vmsnap -o resync (failed)	Contact your system administrator for help. Use Veritas Volume Manager commands to resynchronize the snapshot volumes, and use dbed_vmsnap -o snapshot -F to force snapshot creation.
mount_start	dbed_vmclonedb -o mount (failed)	dbed_vmclonedb -o -umount
mount_end	dbed_vmclonedb -o mount (successful)	dbed_vmclonedb -o umount

Table A-3 Snapshot status information from the CLI (*continued*)

SNAP_STATUS	Completed operations	Allowed operations
restartdb_start	dbed_vmclonedb -o restartdb (failed)	■ dbed_vmclonedb -o umount ■ Start the snapshot database manually.
restartdb_end	dbed_vmclonedb -o restartdb (successful)	dbed_vmclonedb -o umount
mountdb_start	dbed_vmclonedb -o mountdb (failed)	Recover the snapshot database manually, then run dbed_vmclonedb -o update_status
mountdb_end	dbed_vmclonedb -o mountdb (successful)	■ dbed_vmclonedb -o update_status ■ dbed_vmclonedb -o umount
recoverdb_start	dbed_vmclonedb -o recoverdb (failed)	■ Recover the snapshot database manually, then run dbed_vmclonedb -o update_status ■ dbed_vmclonedb -o umount
recoverdb_end	dbed_vmclonedb -o recoverdb (successful)	dbed_vmclonedb -o umount
umount_start	dbed_vmclonedb -o umount (failed)	Verify that your file system(s) are not busy and retry the command.
umount_end	dbed_vmclonedb -o umount (successful)	■ dbed_vmclonedb -o mount ■ dbed_vmclonedb -o restartdb ■ dbed_vmsnap -o resync

Snapshot database status information from the CLI

To view snapshot database status information from the command line, use the `dbed_vmchecksnap` command with the `-o list` option to list all available snapplans for a specified database. Snapshot database status information is displayed in the command output under the column heading `DB_STATUS`.

Table A-4 shows detailed information about each database status (`DB_STATUS`) value.

Note: SF Oracle RAC does not support Database FlashSnap reverse resynchronization.

Table A-4 Snapshot database status information from the CLI

DB_STATUS	Completed operations
init	■ <code>dbed_vmchecksnap -o validate</code> (successful) ■ <code>dbed_vmsnap -o snapshot</code> (successful)
database_recovered	<code>dbed_vmclonedb -o recoverdb</code> (successful)

Using third party software to back up files

This appendix includes the following topics:

- [About using third party software to back up files](#)
- [Using third party software to back up files](#)

About using third party software to back up files

SF Oracle RAC supports the use of third party software for backing up files. However, Quick I/O is not supported for SF Oracle RAC.

Using third party software to back up files

If you are using third party backup software other than Veritas NetBackup, ensure that it can back up and restore VxFS extent attributes. This is important because restored Oracle Disk Manager (ODM) files rely on proper extent layouts for best performance.

Backing up and restoring Oracle Disk Manager files using Oracle RMAN

Oracle allocates Oracle Disk Manager files with contiguous extent layouts for good database performance. When you restore database files they are allocated using these extent attributes. If you are using Oracle RMAN's conventional backup method with any backup software, datafiles are also restored with the proper extent layouts.

If you are using RMAN's "proxy copy" backup method with a backup software other than NetBackup, the extent attributes may not be backed up. To ensure the

restored datafiles have proper extent layouts, preallocate the lost datafiles using the `odmmkfile` command. This command preallocates contiguous space for files prior to restoring them.

For example, to preallocate an Oracle datafile with size 100M, assuming the Oracle database block size is 8K, use the `odmmkfile` command and enter:

```
# /opt/VRTS/bin/odmmkfile -h 8k -s 100m filename
```

For additional information about the `odmmkfile` command, see the `odmmkfile(1)` manual page.

Error messages

This appendix includes the following topics:

- [About error messages](#)
- [LMX error messages](#)
- [VxVM error messages](#)
- [VXFEN driver error messages](#)

About error messages

Error messages can be generated by the following software modules:

- LLT Multiplexer (LMX)
- Veritas Volume Manager (VxVM)
- Veritas Fencing (VXFEN) driver

See [“Turning on the debug messages for the kernel drivers”](#) on page 61.

LMX error messages

There are two types of LMX error messages: critical and non-critical.

Gather information about systems and configurations for Symantec support personnel.

LMX critical error messages

The messages in [Table C-1](#) report critical errors seen when the system runs out of memory, when LMX is unable to communicate with LLT, or when you are unable to load or unload LMX.

Check the `/var/adm/messages` file for the error messages.

[Table C-1](#) lists the critical LMX kernel module error messages.

Table C-1 LMX critical error messages

Message ID	LMX Message
00001	lmxload packet header size incorrect (number)
00002	lmxload invalid lmx_llt_port number
00003	lmxload context memory alloc failed
00004	lmxload port memory alloc failed
00005	lmxload buffer memory alloc failed
00006	lmxload node memory alloc failed
00007	lmxload msgbuf memory alloc failed
00008	lmxload tmp msgbuf memory alloc failed
00009	lmxunload node number conngrp not NULL
00010	lmxopen return, minor non-zero
00011	lmxopen return, no minors available
00012	lmxconnect lmxlltopen(1) err= number
00013	lmxconnect new connection memory alloc failed
00014	lmxconnect kernel request memory alloc failed
00015	lmxconnect mblk memory alloc failed
00016	lmxconnect conn group memory alloc failed
00017	lmxlltfini: LLT unregister failed err = number
00018	lmxload contexts number > number, max contexts = system limit = number
00019	lmxload ports number > number, max ports = system limit = number
00020	lmxload buffers number > number, max buffers = system limit = number
00021	lmxload msgbuf number > number, max msgbuf size = system limit = number

LMX non-critical error messages

If the message displays in [Table C-2](#) creates errors while running an Oracle application, use the `lmxconfig` command to turn off the display. For example:

```
# /sbin/lmxconfig -e 0
```

To re-enable message displays, type:

```
# /sbin/lmxconfig -e 1
```

[Table C-2](#) contains LMX error messages that may appear during run-time.

Table C-2 LMX non-critical error messages

Message ID	LMX Message
06001	lmxreqlink duplicate kreq= 0xaddress, req= 0xaddress
06002	lmxreqlink duplicate ureq= 0xaddress kr1= 0xaddress, kr2= 0xaddress req type = number
06003	lmxrequnlink not found kreq= 0xaddress from= number
06004	lmxrequnlink_l not found kreq= 0xaddress from= number
06005	kreq was not found
06101	lmxpollreq not in doneq CONN kreq= 0xaddress
06201	lmxnewcontext lltinit fail err= number
06202	lmxnewcontext llregister fail err= number
06301	lmxrecvport port not found unode= number node= number ctx= number
06302	lmxrecvport port not found (no port) ctx= number
06303	lmxrecvport port not found ugen= number gen= number ctx= number
06304	lmxrecvport dup request detected
06401	lmxinitport out of ports
06501	lmxsendport lltsend node= number err= number
06601	lmxinitbuf out of buffers
06602	lmxinitbuf fail ctx= number ret= number
06701	lmxsendbuf lltsend node= number err= number

Table C-2 LMX non-critical error messages (continued)

Message ID	LMX Message
06801	lmxconfig insufficient privilege, uid= number
06901	lmxlltnodestat: LLT getnodeinfo failed err= number

VxVM error messages

Table C-3 contains VxVM error messages that are related to I/O fencing.

Table C-3 VxVM error messages for I/O fencing

Message	Explanation
vold_pgr_register(disk_path): failed to open the vxfen device. Please make sure that the vxfen driver is installed and configured.	The vxfen driver is not configured. Follow the instructions to set up these disks and start I/O fencing. You can then clear the faulted resources and bring the service groups online.
vold_pgr_register(disk_path): Probably incompatible vxfen driver.	Incompatible versions of VxVM and the vxfen driver are installed on the system. Install the proper version of SF Oracle RAC.

VXFEN driver error messages

Table C-4 contains VXFEN driver error messages. In addition to VXFEN driver error messages, informational messages can also be displayed.

See “VXFEN driver informational message” on page 313.

See “Node ejection informational messages” on page 313.

Table C-4 VXFEN driver error messages

Message	Explanation
Unable to register with coordinator disk with serial number: xxxx	This message appears when the vxfen driver is unable to register with one of the coordinator disks. The serial number of the coordinator disk that failed is displayed.

Table C-4 VXFEN driver error messages (*continued*)

Message	Explanation
Unable to register with a majority of the coordinator disks. Dropping out of cluster.	This message appears when the vxfen driver is unable to register with a majority of the coordinator disks. The problems with the coordinator disks must be cleared before fencing can be enabled. This message is preceded with the message "VXFEN: Unable to register with coordinator disk with serial number xxxx."
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!	This message appears when there is a preexisting split-brain in the cluster. In this case, the configuration of vxfen driver fails. Clear the split-brain using the instructions given in the chapter on Troubleshooting SF Oracle RAC before configuring vxfen driver.
Unable to join running cluster since cluster is currently fencing a node out of the cluster	This message appears while configuring the vxfen driver, if there is a fencing race going on in the cluster. The vxfen driver can be configured by retrying after some time (after the cluster completes the fencing).

VXFEN driver informational message

The following informational message appears when a node is ejected from the cluster to prevent data corruption when a split-brain occurs.

```
VXFEN CRITICAL V-11-1-20 Local cluster node ejected from cluster
to prevent potential data corruption
```

Node ejection informational messages

Informational messages may appear on the console of one of the cluster nodes when a node is ejected from a disk or LUN.

For example:

```
<date> <system name> scsi: WARNING:
/sbus@3,0/lpfs@0,0/sd@0,1(sd91):
<date> <system name> Error for Command: <undecoded cmd 0x5f>
Error Level: Informational
```

```
<date> <system name> scsi: Requested Block: 0 Error Block 0  
<date> <system name> scsi: Vendor: <vendor> Serial Number:  
0400759B006E  
<date> <system name> scsi: Sense Key: Unit Attention  
<date> <system name> scsi: ASC: 0x2a (<vendor unique code  
0x2a>), ASCQ: 0x4, FRU: 0x0
```

These informational messages can be ignored.

Glossary

Agent	A process that starts, stops, and monitors all configured resources of a type, and reports their status to VCS.
Authentication Broker	The VERITAS Security Services component that serves, one level beneath the root broker, as an intermediate registration authority and a certification authority. The authentication broker can authenticate clients, such as users or services, and grant them a certificate that will become part of the VERITAS credential. An authentication broker cannot, however, authenticate other brokers. That task must be performed by the root broker.
Cluster	A cluster is one or more computers that are linked together for the purpose of multiprocessing and high availability. The term is used synonymously with VCS cluster, meaning one or more computers that are part of the same GAB membership.
CVM (cluster volume manager)	The cluster functionality of Veritas Volume Manager.
Disaster Recovery	Administrators with clusters in physically disparate areas can set the policy for migrating applications from one location to another if clusters in one geographic area become unavailable due to an unforeseen event. Disaster recovery requires heartbeating and replication.
disk array	A collection of disks logically arranged into an object. Arrays tend to provide benefits such as redundancy or improved performance.
DMP (Dynamic Multipathing)	A feature of designed to provide greater reliability and performance by using path failover and load balancing for multiported disk arrays connected to host systems through multiple paths. DMP detects the various paths to a disk using a mechanism that is specific to each supported array type. DMP can also differentiate between different enclosures of a supported array type that are connected to the same host system.
DST (Dynamic Storage Tiering)	A feature with which administrators of multi-volume file systems can manage the placement of files on individual volumes in a volume set by defining placement policies that control both initial file location and the circumstances under which existing files are relocated. These placement policies cause the files to which they apply to be created and extended on specific subsets of a file system's volume set, known as placement classes. The files are relocated to volumes in other placement

classes when they meet specified naming, timing, access rate, and storage capacity-related conditions.

See also Veritas File System (VxFS)

Failover	A failover occurs when a service group faults and is migrated to another system.
GAB (Group Atomic Broadcast)	A communication mechanism of the VCS engine that manages cluster membership, monitors heartbeat communication, and distributes information throughout the cluster.
HA (high availability)	The concept of configuring the to be highly available against system failure on a clustered network using Veritas Cluster Server (VCS).
HAD (High Availability Daemon)	The core VCS process that runs on each system. The HAD process maintains and communicates information about the resources running on the local system and receives information about resources running on other systems in the cluster.
IP address	An identifier for a computer or other device on a TCP/IP network, written as four eight-bit numbers separated by periods. Messages and other data are routed on the network according to their destination IP addresses. See also virtual IP address
Jeopardy	A node is in jeopardy when it is missing one of the two required heartbeat connections. When a node is running with one heartbeat only (in jeopardy), VCS does not restart the applications on a new node. This action of disabling failover is a safety mechanism that prevents data corruption.
latency	For file systems, this typically refers to the amount of time it takes a given file system operation to return to the user.
LLT (Low Latency Transport)	A communication mechanism of the VCS engine that provides kernel-to-kernel communications and monitors network communications.
logical volume	A simple volume that resides on an extended partition on a basic disk and is limited to the space within the extended partitions. A logical volume can be formatted and assigned a drive letter, and it can be subdivided into logical drives. See also LUN
LUN	A LUN, or logical unit, can either correspond to a single physical disk, or to a collection of disks that are exported as a single logical entity, or virtual disk, by a device driver or by an intelligent disk array's hardware. VxVM and other software modules may be capable of automatically discovering the special characteristics of LUNs, or you can use disk tags to define new storage attributes. Disk tags are administered by using the <code>vxdisk</code> command or the graphical user interface.
main.cf	The file in which the cluster configuration is stored.

mirroring	A form of storage redundancy in which two or more identical copies of data are maintained on separate volumes. (Each duplicate copy is known as a mirror.) Also RAID Level 1.
Node	The physical host or system on which applications and service groups reside. When systems are linked by VCS, they become nodes in a cluster.
resources	Individual components that work together to provide application services to the public network. A resource may be a physical component such as a disk group or network interface card, a software component such as a database server or a Web server, or a configuration component such as an IP address or mounted file system.
Resource Dependency	A dependency between resources is indicated by the keyword "requires" between two resource names. This indicates the second resource (the child) must be online before the first resource (the parent) can be brought online. Conversely, the parent must be offline before the child can be taken offline. Also, faults of the children are propagated to the parent.
Resource Types	Each resource in a cluster is identified by a unique name and classified according to its type. VCS includes a set of pre-defined resource types for storage, networking, and application services.
root broker	The first authentication broker, which has a self-signed certificate. The root broker has a single private domain that holds only the names of brokers that shall be considered valid.
SAN (storage area network)	A networking paradigm that provides easily reconfigurable connectivity between any subset of computers, disk storage and interconnecting hardware such as switches, hubs and bridges.
Service Group	A service group is a collection of resources working together to provide application services to clients. It typically includes multiple resources, hardware- and software-based, working together to provide a single service.
Service Group Dependency	A service group dependency provides a mechanism by which two service groups can be linked by a dependency rule, similar to the way resources are linked.
Shared Storage	Storage devices that are connected to and used by two or more systems.
shared volume	A volume that belongs to a shared disk group and is open on more than one node at the same time.
SFCFS (Storage Foundation Cluster File System)	The Veritas Cluster File System is an extension of the industry standard Veritas File System (VxFS). CFS allows the same file system to be simultaneously mounted on multiple nodes. Unlike other clustered file systems, CFS is a true SAN file system. All I/O takes place over the storage area network.
SNMP Notification	Simple Network Management Protocol (SNMP) developed to manage nodes on an IP network.

State	The current activity status of a resource, group or system. Resource states are given relative to both systems.
Storage Checkpoint	A facility that provides a consistent and stable view of a file system or database image and keeps track of modified data blocks since the last Storage Checkpoint.
System	The physical system on which applications and service groups reside. When a system is linked by VCS, it becomes a node in a cluster. See Node
types.cf	A file that describes standard resource types to the VCS engine; specifically, the data required to control a specific resource.
VCS (Veritas Cluster Server)	An open systems clustering solution designed to eliminate planned and unplanned downtime, simplify server consolidation, and allow the effective management of a wide range of applications in multiplatform environments.
Virtual IP Address	A unique IP address associated with the cluster. It may be brought up on any system in the cluster, along with the other resources of the service group. This address, also known as the IP alias, should not be confused with the base IP address, which is the IP address that corresponds to the host name of a system.
VxFS (Veritas File System)	A component of the Veritas product suite that provides high performance and online management capabilities to facilitate the creation and maintenance of file systems. A file system is a collection of directories organized into a structure that enables you to locate and store files.
VxVM (Veritas Volume Manager)	A Symantec product installed on storage clients that enables management of physical disks as logical devices. enhances data storage management by controlling space allocation, performance, data availability, device installation, and system monitoring of private and shared systems.
VVR (Veritas Volume Replicator)	A data replication tool designed to contribute to an effective disaster recovery plan.

Index

Symbols

/sbin/vcsmmconfig
starting VCMM 274

A

arrays
configuring 249

B

backing up
 using Storage Checkpoints 126
 using Storage Checkpoints and Storage
 Rollback 122
backing up a database 197

C

clone 207
clone databases
 restarting 208
 shutting down 207
clone databases, creating 152
cluster
 Group membership services/Atomic Broadcast
 (GAB) 27
 interconnect communication channel 26
 low latency transport (LLT) 26
Cluster File System (CFS)
 architecture 31
 communication 32
 overview 31
Cluster Volume Manager (CVM)
 architecture 29
 communication 30
 overview 29
commands 243
 dbed_analyzer 246
 dbed_ckptcreate 134, 140
 dbed_ckptdisplay 142
 dbed_ckptremove 152
 dbed_ckptrollback 150

commands *(continued)*

 dbed_ckptumount 146
 dbed_clonedb 152
 dbed_update 140
 format (verify disks) 274
 vxctl enable (scan disks) 274
 vxstorage_stats 243

communication

 communication stack 25
 data flow 24
 GAB and processes port relationship 28
 Group membership services/Atomic Broadcast
 GAB 27
 interconnect communication channel 26
 requirements 25

coordinator disks

 DMP devices 66
 for I/O fencing 66

cron 144

 scheduling Storage Checkpoints 145

crontab file 145

CRS installation

 removing CRS if installation fails 276

D

data corruption

 preventing 40

data disks

 for I/O fencing 66

Database FlashSnap

 applications 158
 backing up
 databases 197
 commands 161
 copying a snapplan 191
 creating a snapshot mirror 163
 dbed_vmchecksnap 191
 dbed_vmclonedb 197
 dbed_vmsnap 194
 dbed_vmsnap -o resync 211

Database FlashSnap (*continued*)

- displaying a snapplan 191
- host and storage requirements 163
- node in the cluster configuration 162
- overview 157, 177
- planning considerations 162
- removing a snapplan 191
- removing a snapshot volumesnapshot volumes
 - removing 212
- resynchronizing 211
- selecting the snapshot mode 162
- setting up hosts 162

database snapshots

- creating 194

databases

- unmounting file systems 207
- dbed_analyzer command 246
- dbed_ckptcreate command 134, 140
- dbed_ckptdisplay command 142
- dbed_ckptremove command 152
- dbed_ckptrollback command 150
- dbed_ckptumount command 146
- dbed_clonedb command 152
- dbed_update command 140
- dbed_vmchecksnap command 191
- dbed_vmclonedb command 197
- dbed_vmsnap -o resync command 211
- dbed_vmsnap command 194

drivers

- tunable parameters 289

E**error messages**

- node ejection 313
- VxVM errors related to I/O fencing 312

example

- configuring LMX parameters 291

F**file**

- errors in Oracle trace/log files 273
- errors in trace/log files 273
- reading /etc/llttab file 274

format command 274**G****getcomms**

- troubleshooting 258

getdbac

- troubleshooting script 258

Global Cluster Option (GCO)

- overview 44

H**hagetcf** (troubleshooting script) 258**I****I/O**

- displaying Storage Mapping statistics 245

I/O fencing

- communication 42
- operations 42
- preventing data corruption 40
- testing and scenarios 67

IP address

- troubleshooting VIP configuration 278

K**kernel**

- tunable driver parameters 289

L**Listener**

- description 23

LLT multiplexer (LMX)

- overview 26

LMX

- error messages, non-critical 311
- tunable parameters 290

low latency transport (LLT)

- overview 26

M**messages**

- LMX error messages, non-critical 311
- node ejected 313
- VXFEN driver error messages 312

minors

- appearing in LMX error messages 273

O**Oracle Disk Manager**

- restoring files using NetBackup 307

Oracle Disk Manager (ODM)

- overview 34

- Oracle instance
 - definition 20
- Oracle tempfiles
 - recreating 209
- Oracle user
 - reading /etc/llttab file 274

R

- recovering
 - using Storage Checkpoints 126
- removing snapshot volumes 212
- reservations
 - description 41
- restoring
 - using Storage Checkpoints and Storage Rollback 122
- resynchronizing a snapshot 211

S

- SCSI-3 PR 41
- SF Oracle RAC
 - architecture 20
 - communication infrastructure 24
 - error messages 309
 - high-level functionality 20
 - overview of components 19, 23
 - Storage Mapping 241
 - tunable parameters of kernel drivers 289
 - using Storage Checkpoints 121
- SFRAC
 - tunable parameters 289
- SFRAC components
 - Cluster Volume Manager (CVM) 29
- snapplans
 - copying 191
 - displaying 191
 - removing 191
- snapshot volumes
 - backing up a database 197
 - creating
 - using the command line 166, 168
 - mounting 199
 - removing 212
 - resynchronizing 211
- snapshots
 - creating 194
- Storage Checkpoints 122
 - backing up and recovering 126

Storage Checkpoints *(continued)*

- backing up and recovering databases 126
- creating 140
- determining space requirements 124
- displaying 142
- performance 125
- removing 152
- unmounting 146
- using the CLI 137
- verifying 128

Storage Mapping

- configuring arrays 249
- dbed_analyzer command 246
- description 241
- displaying I/O statistics 245
- displaying information 244
- displaying information for a list of tablespaces 247
- enabling Oracle file mapping 252
- mapping components 250
- Oracle file mapping 250
- ORAMAP 250
- using the vxstorage_stats command 243
- verifying feature set up 243
- verifying Oracle file mapping set up 252
- verifying set up 243
- views 251, 253
- vxstorage_stats 243

- Storage Rollback 122, 150
 - guidelines for recovery 134

T

troubleshooting

- CVMVolDg 268
- error when starting Oracle instance 274
- File System Configured Incorrectly for ODM 279
- getcomms 258
 - troubleshooting script 258
- getdbac 258
- hagetcf 258
- Oracle log files 275
- overview of topics 267, 272–273, 279
- restoring communication after cable disconnection 274
- running scripts for analysis 258
- scripts 258
- SCSI reservation errors during bootup 261
- shared disk group cannot be imported 267

V**VCSIPC**

- errors in Oracle trace/log files 273
- errors in trace/log files 273
- overview 26, 39

VCSMM

- vcsmmconfig command 274

vxassist

- used to add DCOs to volumes 163

vxctl command 274**VXFEN driver error messages 312****VXFEN driver informational message 313****vxstorage_stat command 243****vxstorage_stats 243****vxstorage_stats command 243****VxVM**

- error messages related to I/O fencing 312

VxVM (Volume Manager)

- errors related to I/O fencing 312