Veritas Storage Foundation™ for Oracle RAC Installation and Configuration Guide

AIX

5.0

N18488J

symantec.

Veritas Storage Foundation for Oracle RAC Installation and Configuration Guide

Storage Foundation 5.0 for Oracle RAC

Symantec Corporation
20330 Stevens Creek Blvd.
Cupertino, CA 95014
www.symantec.com

Third-party legal notices

Third-party software may be recommended, distributed, embedded, or bundled with this Symantec product. Such third-party software is licensed separately by its copyright holder. All third-party copyrights associated with this product are listed in the accompanying release notes.

AIX is a registered trademark of IBM Corporation.
Oracle is a registered trademark of Oracle Corporation.

Licensing and registration

Veritas Storage Foundation for Oracle RAC is a licensed product. See the *Veritas Storage Foundation for Oracle RAC Installation and Configuration Guide* for license installation instructions.

Technical support

For technical assistance, visit http://support.veritas.com and select phone or email support. Use the Knowledge Base search feature to access resources such as TechNotes, product alerts, software downloads, hardware compatibility lists, and our customer email notification service.

Contents

Section 1 SF Oracle RAC Concepts

Chapter 1 Introducing SF Oracle RAC

Section 2 Installing and upgrading SF Oracle RAC

Chapter 2 Preparing to install SF Oracle RAC

Chapter 3 Installing and Configuring SF Oracle RAC Software

Section 6 Backup and recovery

Chapter 18 Configuring the repository database for Oracle

Chapter 19 Using Checkpoints and Storage Rollback with Storage Foundation for Oracle RAC

Chapter 20 Using database FlashSnap for backup and off-host processing

Section 7 Performance and troubleshooting

Chapter 21 Investigating I/O performance using storage mapping

Chapter 22 Troubleshooting SF Oracle RAC

Section 8 Reference information

Appendix A Sample VCS configuration files for SF Oracle RAC

Appendix B Creating a starter database

Appendix C Agent reference

Appendix D I/O fencing topics

Section I

SF Oracle RAC Concepts

Read this section to understand SF Oracle RAC product concepts.

Chapter 1

Introducing SF Oracle RAC

Veritas Storage Foundation for Oracle RAC (SF Oracle RAC) from 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.

About SF Oracle RAC

SF Oracle RAC is a storage management and clustering solution that enables you to:

- 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 Cluster Ready Services (CRS).
- Enhance scalability and availability with access to multiple 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.

- 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 simulator and firedrill 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:

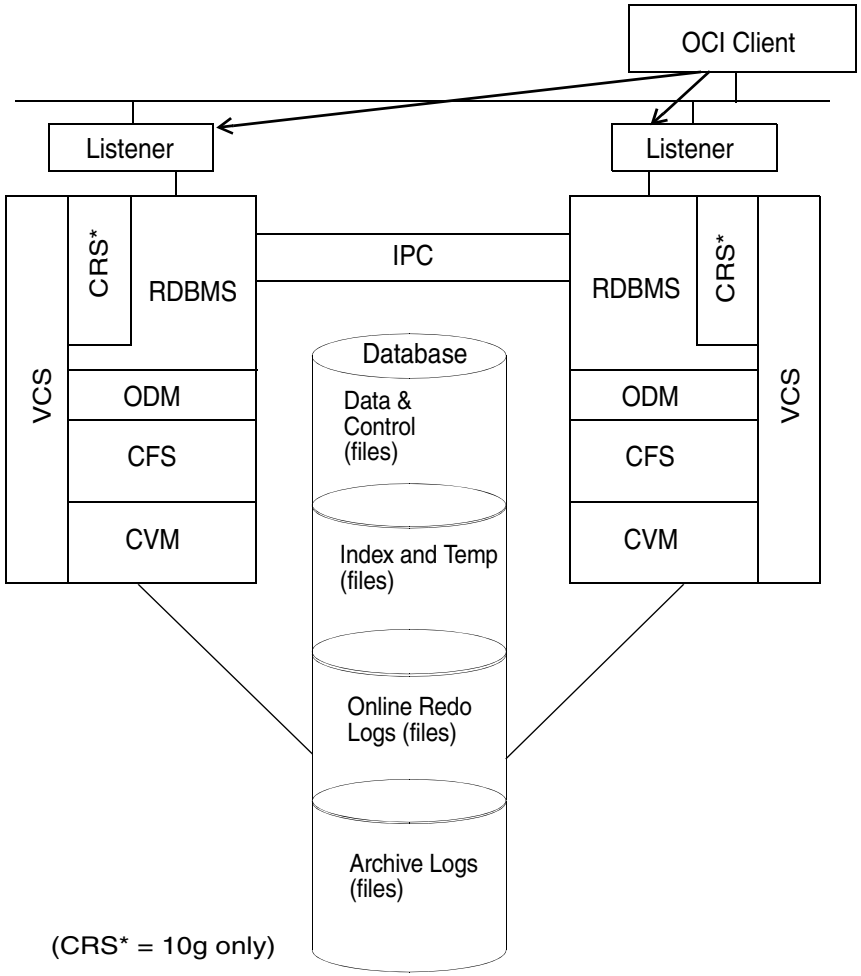
- Each node runs its own operating system.
- A cluster interconnect enables cluster communications.
- A public network connects each node to a LAN for client access.
- Shared storage is accessible by each node that needs to run the application.

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

- 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.

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 1-1 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. General highlights of the component products include:

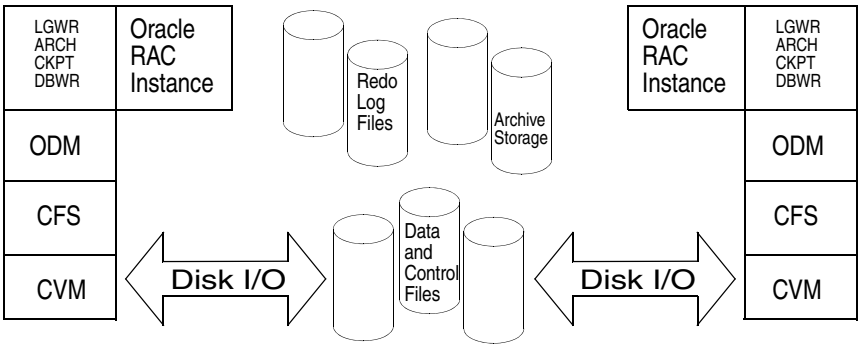
- Cluster Volume Manager (CVM) -- Enables simultaneous access to shared volumes based on technology from Veritas Volume Manager (VxVM).
- Cluster File System (CFS) -- Enables simultaneous access to shared file systems based on technology from Veritas File System (VxFS).
- Database Accelerator -- Provides the interface with the Oracle Disk Manager (ODM) API.
- Cluster Server (VCS) -- Uses technology from Veritas Cluster Server to manage Oracle RAC databases and infrastructure components.
- RAC Extensions -- Manages cluster membership and communications between cluster nodes.

Communication infrastructure

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

Data flow

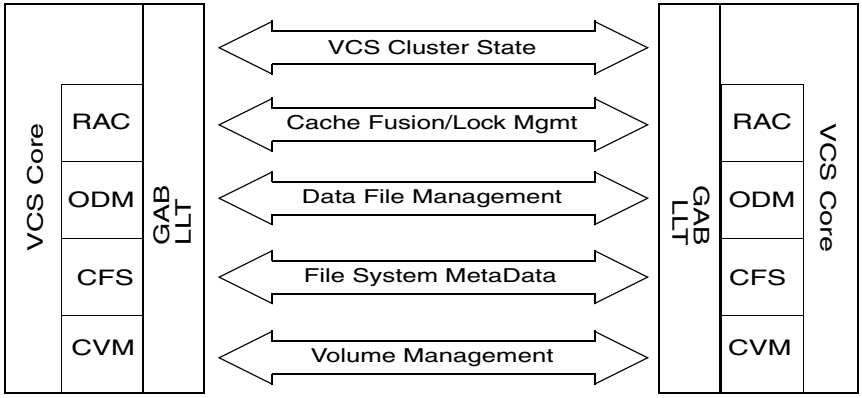
The CVM, CFS, ODM, and Oracle RAC elements reflect the overall data flow, or data stack, from an instance running on a server to the shared storage. The various Oracle processes composing an instance -- such as DB Writers, Log Writer, Checkpoint, Archiver, and Server -- read and write data to the storage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



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 communication 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.



Cluster interconnect communication channel

The cluster interconnect provides the communication channel for all system-to-system communication, in addition to 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 and Cache Fusion, 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

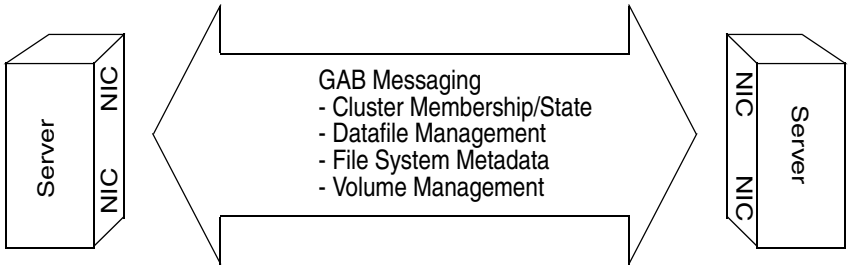
LLT is responsible for sending and receiving heartbeat traffic over network links. The Group Membership Services function of GAB uses heartbeats to determine cluster membership.

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.



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 processes 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.

Cluster Volume Manager

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`. (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.

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 is 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.

Cluster File System

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 some other file systems that 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.

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

Oracle Disk Manager

SF Oracle RAC requires Oracle Disk Manager (ODM), a standard API published by Oracle for support of database I/O. SF Oracle RAC 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 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 - Port d

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 database parameters, such as the `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.

Veritas Cluster Server

VCS directs SF Oracle RAC operations by controlling the startup and shutdown of component 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 with 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.

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` and `PrivNIC.cf`.

RAC extensions

Oracle RAC relies on several support services provided by VCS. Key features include Veritas Cluster Server Membership Manager (VCSMM) and Cluster 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/CRS 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.

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.

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 linked to Oracle at installation time to implement the `skgxp` functionality. This module communicates with the LLT Multiplexer (LMX) via `ioctl` calls.

LMX 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.

I/O fencing

I/O fencing is a mechanism to prevent uncoordinated access to the shared storage. This feature works even in the case of faulty cluster communications causing a split-brain condition.

Understanding Split Brain and the need for I/O fencing

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 and the remaining node takes corrective action. However, the failure of private interconnects (instead of the actual nodes) would present identical symptoms and cause each node to determine its peer has departed. This situation typically results in data corruption because both nodes attempt 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 nodes using 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 even though the system later returns and begins write operations.

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

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 simple 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.

Components of I/O fencing

Fencing in SF Oracle RAC involves coordinator disks 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.

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 added to a disk group are automatically fenced, as are new paths discovered to a device.

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 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 only register with them and do not need to reserve them.

These disks 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 coordinator disks before it can fence the peer from the data drives. This concept of racing for

control of the coordinator disks to gain the ability to fence data disks is key to understanding prevention of split brain through fencing.

Dynamic Multipathing devices with I/O fencing

DMP allows coordinator disks to take advantage of the path failover and the dynamic adding and removal capabilities of DMP. You can configure coordinator disks to use Veritas Volume Manager Dynamic Multipathing (DMP) feature. Veritas Volume Manager DMP uses the ASL model for communicating with storage. As a result, DMP can discover devices and failover paths, and can issue other SCSI commands suitable for the unique characteristics of each array.

For more information on using DMP, see the *Veritas Volume Manager Administrator's Guide*.

Also see "Enabling fencing in the VCS configuration" on page 112.

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 attempts 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 reboots the system.

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.

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.

Additional features of SF Oracle RAC

Additional SF Oracle RAC features include:

- The ability to transition from a local high-availability cluster into a disaster recovery environment. This environment ensures maximum data protection and availability in the event of large-scale disasters and involves global clustering with VCS and replication with Veritas Volume Replicator (VVR).
- The ability to back up and recover data at the volume and file system levels using Veritas Database Flashsnap and Veritas Storage Checkpoints.
- 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.

Section II

Installing and upgrading SF Oracle RAC

Install and configure SF Oracle RAC. After completing this process, proceed to the appropriate Oracle section for all Oracle-specific procedures.

Chapter 2

Preparing to install SF Oracle RAC

The following topics contain important planning information for installing SF Oracle RAC:

Supported SF Oracle RAC installations work with Oracle9i R2, Oracle 10g R1, and Oracle 10g R2. The *Veritas Storage Foundation 5.0 for Oracle RAC Release Notes* provides details on SF Oracle RAC requirements.

Overview of SF Oracle RAC installation and configuration tasks

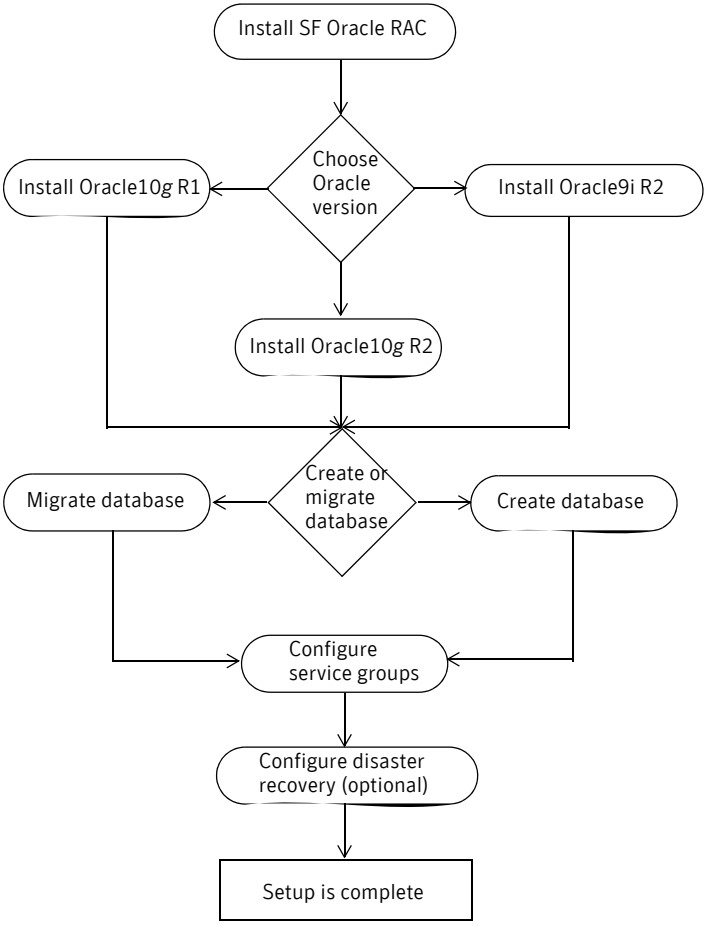
Phases involved in installing and configuring SF 5.0 Oracle RAC include:

- Preparing to install and configure SF Oracle RAC
- Installing SF Oracle RAC and configuring its components
- Installing Oracle RAC and creating Oracle RAC database
- Setting up VCS to manage RAC resources
- Setting up disaster recovery in SF Oracle RAC environment (optional)
- Setting up backup and recovery feature for SF Oracle RAC (optional)

For a high-level flow of the SF 5.0 Oracle RAC installation and configuration process:

See "High-level process of setting up SF Oracle RAC" on page 43.

Figure 2-1 High-level process of setting up SF Oracle RAC



Preparing to install and configure SF Oracle RAC

Before installing or upgrading SF Oracle RAC, you must:

- Make sure you meet the installation requirements.
  See "SF Oracle RAC prerequisites" on page 67.
- Set up the basic hardware and plan your configuration.
  Details about supported hardware are on the Veritas support web site: http://support.veritas.com.
  See "Typical SF Oracle RAC cluster setup" on page 54.
- Perform the SF Oracle RAC pre-installation and pre-configuration tasks. Gather the required information to install and configure SF Oracle RAC.
  See "Performing preinstallation tasks" on page 70.

Installing SF Oracle RAC and configuring its components

Install SF Oracle RAC on clusters of up to eight nodes; note that a global cluster environment only allows a maximum of two nodes per cluster. In addition to installing SF 5.0 Oracle RAC, you can also upgrade an existing cluster to SF 5.0 Oracle RAC.

See "Installing and Configuring SF Oracle RAC Software" on page 83.

See "Upgrading SF Oracle RAC" on page 115.

Installing and configuring SF Oracle RAC involves:

Installing SF Oracle RAC	Use the Veritas product installer or the installsfrac program. On each node, the interactive installer installs packages for: ■ Veritas Cluster Server (VCS) ■ Veritas Volume Manager (VxVM) ■ Veritas File System (VxFS) ■ Veritas High Availability Agent for Oracle ■ Veritas Volume Replicator (VVR) ■ Other SF Oracle RAC modules
Performing system checks to configure SF Oracle RAC	Use the Veritas product installer or `installsfrac -configure` option of installsfrac program. The installer guides you to perform basic system checks.

Configuring SF Oracle RAC stack	Use the Veritas product installer or `installsfrac -configure` option of installsfrac program. After you perform basic system checks, you can configure the SF Oracle RAC components: ■ VCS ■ CVM, Veritas Volume Manager enabled for clusters ■ CFS, Veritas File System enabled for clusters ■ VVR (optional) The installer also starts the SF Oracle RAC processes.
Setting up I/O fencing	Manually configure I/O fencing feature for SF Oracle RAC: ■ Verify whether shared storage can support I/O fencing using vxfentsthdw script. ■ Set up coordinator disks for the I/O fencing feature into a disk group. ■ Set the UseFence=SCSI3 attribute in the configuration file. ■ Restart the processes.

Installing Oracle RAC and creating Oracle RAC database

After installing and configuring components of Storage Foundation for Oracle RAC, proceed to install Oracle RAC. SF 5.0 Oracle RAC supports Oracle9i R2, Oracle10*g* R1 and R2.

- Prepare to install Oracle RAC
  See "Preparing to Install Oracle9i RAC" on page 129.
  See "Preparing to Install Oracle 10g RAC" on page 201.
- Install Oracle RAC.
  See "Installing Oracle9i RAC" on page 137.
  See "Installing Oracle 10g RAC" on page 213.
- Create a raw database on raw volumes within a VxVM disk group or on a Veritas cluster file system.
  Numerous procedures exist for creating a database. If you decide to use the Oracle dbca utility, review the procedure to create a database.
  See "Creating a starter database" on page 443.

Setting up VCS to manage RAC resources

SF Oracle RAC provides the capability to completely automate the RAC environment. This capability ranges from enabling automatic control of the entire database environment to having VCS mount cluster file systems or enable

CVM and CFS daemons. The user or DBA is free to choose the level of control and automation.

VCS uses the main.cf configuration file to manage resources in the cluster. The SF Oracle RAC installation process creates a basic VCS configuration file. After installing Oracle and creating the database, you can modify the main.cf file on one of the cluster nodes to reflect the new resources and their configuration.

You can configure VCS service groups using the configuration wizard or manually.

See "Configuring Oracle9i service groups" on page 147.

See "Configuring Oracle 10g service groups" on page 233.

See "Sample VCS configuration files for SF Oracle RAC" on page 425.

Setting up disaster recovery in SF Oracle RAC environment (optional)

You can create a global cluster environment with SF Oracle RAC and volume replication capability. VCS provides the Global Cluster Option (GCO) for wide-area failover and disaster recovery, and Veritas Volume Replicator provides the volume replication capability.

After installing SF Oracle RAC on each node in the cluster, you can choose to set up a global cluster environment for disaster recovery. The general process for setting up a global cluster involves:

- Creating a cluster on a secondary site with hardware set up
- Installing SF Oracle RAC
- Installing Oracle RAC
- Configuring VCS service groups

You do not need to create a database for the secondary site, since it will be replicated from the primary site.

See "Preparing for global clustering" on page 273.

See "Configuring global clustering" on page 281.

Setting up backup and recovery feature for SF Oracle RAC (optional)

You can configure the following SF Oracle RAC optional features to back up and recover data at the volume and file system levels:

- Veritas Storage Checkpoint
  Allows efficient backup and recovery of Oracle RAC databases. This feature is available with SF Oracle RAC as part of the Veritas File System.

See "Using Checkpoints and Storage Rollback with Storage Foundation for Oracle RAC" on page 319.

- Veritas Database FlashSnap
  Allows you to create a point-in-time copy of an Oracle RAC database for backup and off-host processing.
  See "Using database FlashSnap for backup and off-host processing" on page 335.
- Veritas Storage Mapping
  Allows you to evaluate or troubleshoot I/O performance. You can access mapping information that allows for a detailed understanding of the storage hierarchy in which files reside.
  Chapter 21, "Investigating I/O performance using storage mapping" on page 389

About SF Oracle RAC component features

Review the description of the optional features and decide the features that you want to configure with SF Oracle RAC:

- Symantec Product Authentication Service
- Veritas Cluster Management Console
- Notification for VCS events
- Global Clusters
- Veritas Volume Replicator
- Typical SF Oracle RAC cluster setup

Note: To configure the optional features of the SF Oracle RAC components, make sure to install all packages when the installation program prompts you.

Symantec Product Authentication Service

The Symantec Product Authentication Service is a common Veritas feature 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:

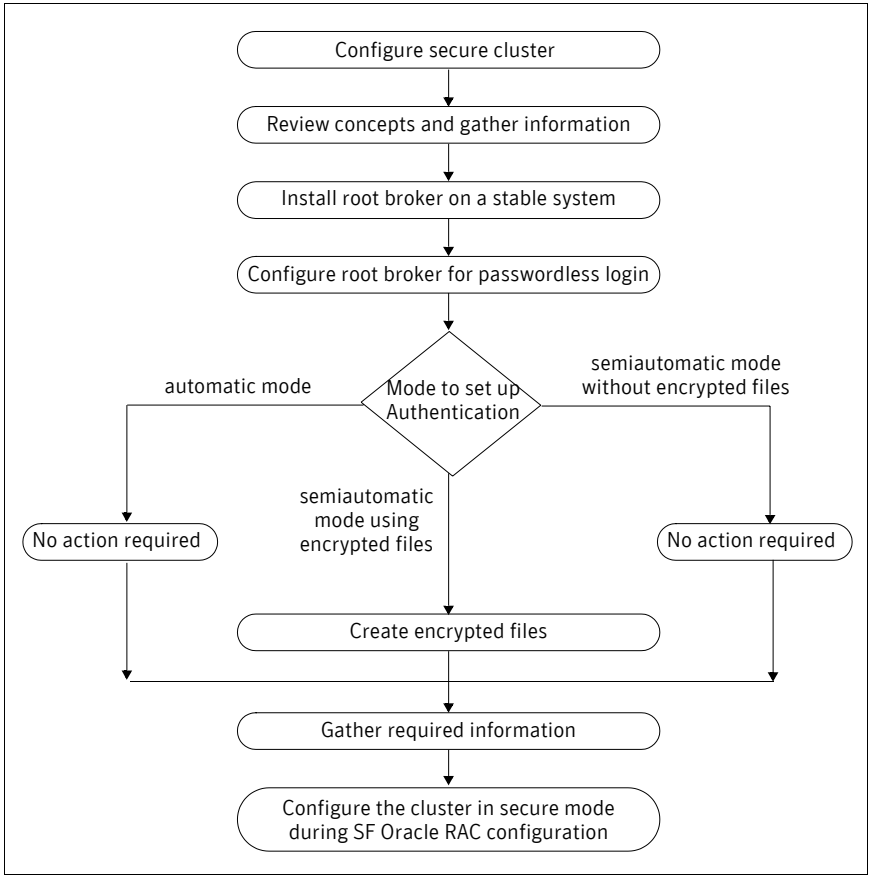
- Cluster nodes and clients, including the VCS Java and the Web consoles
  You can set up Authentication Service for the cluster during the SF Oracle RAC installation and configuration process. If you want to enable Authentication Service after installation, refer to the *Veritas Cluster Server User's Guide*.
  See "Configuring the cluster in secure mode" on page 92
- Veritas Cluster Management Console Management Server and the centrally managed SF Oracle RAC clusters
  See "Veritas Cluster Management Console" on page 51.
- Veritas Storage Foundation Management Server and the centrally managed hosts
  See "Typical SF Oracle RAC cluster setup" on page 54.

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.

Figure 2-2 depicts the flow of configuring SF Oracle RAC in secure mode.

Figure 2-2 Secure SF Oracle RAC cluster configuration flowchart



If you decide to enable the Authentication Service, the root broker administrator must perform the following preparatory tasks:

- Install the root broker on another stable system.
  The root broker is the main registration and certification authority and can serve multiple clusters. Symantec recommends that you install a single root broker on a utility computer such as an email server or domain controller, which can be highly available.
  See "Installing root broker for Symantec Product Authentication Service" on page 58.
- Configure the root broker system for a passwordless login when you want to use the automatic mode.

The installsfrac program provides the following modes to configure Symantec Product Authentication Service:

- In the automatic mode, the installer configures Authentication Service automatically without any user intervention.
  You must provide the name of the root broker system.
- In the semiautomatic modes, the installer provides you an option to use encrypted files or answer the installer prompts to enable security. The semiautomatic mode requires the root broker administrator to set up the basic authentication environment and create principals for authentication brokers. You must complete the following preparatory tasks to configure security in the semiautomatic mode:

With encrypted files	■ The root broker administrator must create an encrypted file for each node in the cluster. See "Creating encrypted files for Symantec Product Authentication Service" on page 59. ■ The root broker administrator must provide the encrypted files in a media or make it available on a shared location that you can access. ■ You must copy the encrypted files to a directory in the installation node. Make a note of the path of this encrypted files.
Without encrypted files	■ You must gather the following information from the root broker administrator: Root broker name Root broker domain name Root broker port (Default is 2821) Authentication broker principal name for each node Authentication broker password for each Authentication broker ■ The root broker administrator must provide the root_hash file in a media or make it available on a shared location that you can access. ■ You must copy the root_hash file to a directory in the installation node. Make a note of the path of this root_hash file.

Refer to the *Symantec Product Authentication Service Administrator's Guide* for more information.

Note: Make sure that the system clocks of the Rook Broker and Authentication Brokers systems are in sync.

Veritas Cluster Management Console

Veritas Cluster Management Console is a high availability management solution that enables monitoring and administering SF Oracle RAC clusters from a single web console.

You can configure Cluster Management Console to manage a single cluster, multiple clusters, or both.

- If you want to use Cluster Management Console to manage multiple clusters, you must set up a management server.
- If you want to use the Cluster Management Console to manage a single cluster, choose the option to install the Cluster Management Console during SF Oracle RAC installation and configuration.

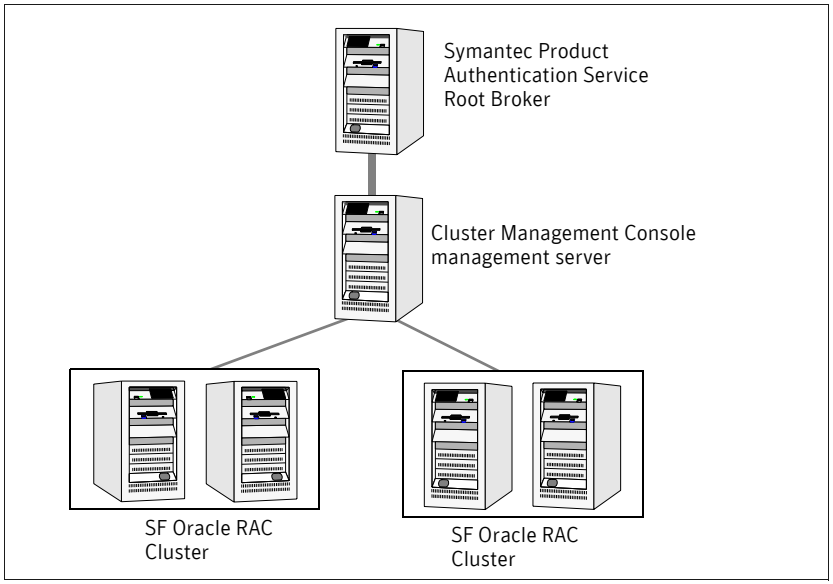
Operational mode	Configurational description
Local management of one cluster (single-cluster mode)	The Cluster Management Console is installed along with SF Oracle RAC on each node in the cluster and is configured for failover. It is integrated with SF Oracle RAC as part of the ClusterService service group. The Cluster Management Console offers robust cluster management capability and can be run from any supported Web browser on any system. See "Configuring the Cluster Management Console" on page 95.

Operational mode	Configurational description
Centralized, comprehensive, enterprise-wide administration of multiple clusters (multi-cluster mode)	One instance of the Cluster Management Console is installed outside all clusters on a standalone server. The console enables users to visually and intuitively input commands to the multi-cluster management engine, the *management server*. The management server initiates monitoring and management actions based upon those commands. The management server uses a database to store cluster configurations, cluster status, events, event policies, report jobs, report outputs, and more. See "Installing the management server for the Veritas Cluster Management Console" on page 61. If the management server and cluster nodes are separated by a firewall, a component called *cluster connector* is installed on each cluster node. Cluster connector enables communication with clusters through firewalls. Cluster connector also provides buffering for cluster data. If the console goes offline and then comes back online, it can retrieve data collected during the offline period from the cluster connector buffer. See "Configuring cluster connector" on page 94. The console offers additional capability for administering users, reports, events, and notification. If the cluster environment includes licensed VCS global clusters, disaster recovery (DR) capability is also available.

The configurational differences between the operational modes mean that you cannot switch a single Cluster Management Console installation from one mode to the other. The modes are also incompatible on the same system. Consequently, one system cannot offer both operational modes. However, the modes *can* co-exist in the same multi-cluster environment, with single-cluster-mode installations on SF Oracle RAC cluster nodes, and multi-cluster-mode installations on management server hosts. Such a deployment can be desirable if different IT administrators in your enterprise have different scopes of responsibility.

See *Veritas Cluster Server Centralized Management Guide* for more information.

Figure 2-3 Sample deployment for Veritas Cluster Management Console



Notification for VCS events

You have the option to configure SMTP email notification and SNMP trap notification of VCS events by the VCS Notifier component. Refer to the *Veritas Cluster Server User's Guide* for more information on SMTP and SNMP notification.

Global Clusters

Global clusters provide the ability to fail over applications between geographically distributed clusters when disaster occurs. You require a separate license to configure global clusters. You must add this license during the installation.

If you choose to configure global clusters, the installer enables you to choose whether or not to use the same NIC, virtual IP address, and netmask as are configured for the ClusterService group, which are the defaults. If you choose not to use the same networking information, you must specify appropriate values for the NIC, virtual IP address, and netmask when you are prompted.

Veritas Volume Replicator

Veritas Volume Replicator is an optional, separately-licensable feature of SF Oracle RAC. Volume Replicator 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.

Typical SF Oracle RAC cluster setup

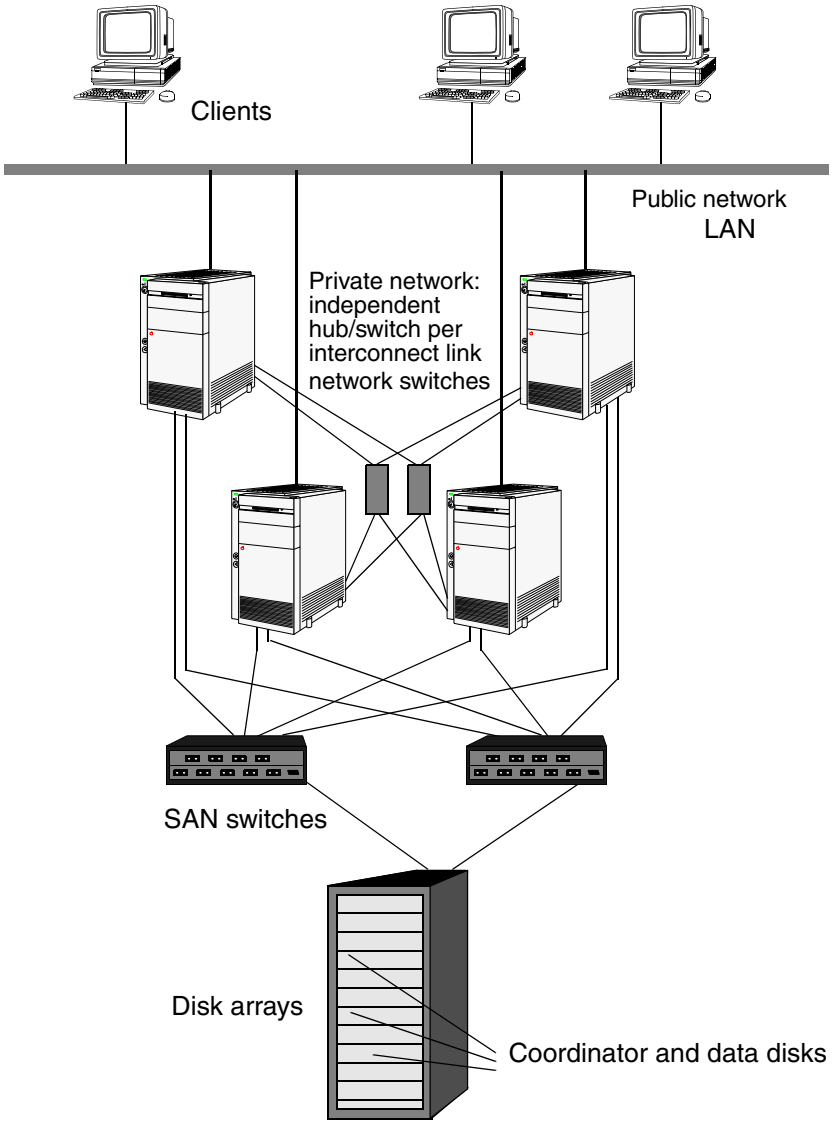
From a high-level, if you install SF Oracle RAC with Oracle9i or Oracle10*g* and create a database, the SF Oracle RAC cluster typically has the following characteristics:

- Nodes connected by at least two VCS private network links using 100 Base T or Gigabit. Ethernet controllers on each system.
  For two-node clusters, cross-over Ethernet cables are acceptable. For three or more nodes, Ethernet switches can be used. Symantec recommends Gigabit Ethernet using enterprise-class switches for the private links. In either case, use a minimum of two switches to provide necessary redundancy. If multiple links are present on a single switch, such as cases where three or four links are configured, a separate VLAN must exist for each link. The use of multiple links on a single hub is not supported.
  Nodes are connected to shared storage devices through Fibre Channel switch. Symantec does not support the use of shared SCSI with the SF Oracle RAC product. For a complete list of supported Fibre Channel storage devices, see the current hardware compatibility list on the Symantec Support Web site.
  http://support.veritas.com
- Nodes running Veritas Cluster Server (VCS), Veritas Volume Manager with cluster features (CVM), Veritas File System with cluster features (CFS), and Storage Foundation for Oracle RAC agents and components, including I/O fencing.
- Oracle RAC database is configured on the shared storage that is available to each node. The shared storage could be cluster file system or raw volumes. All shared storage, including coordinator disks, must support SCSI-3 PR.
- VCS is configured to enable agents to direct and manage the resources required by Oracle RAC. This configuration is recommended but not required. The resources run in parallel on each system.

For a high-level view of an SF Oracle RAC configuration for a four-node cluster:

See "View of the SF Oracle RAC cluster" on page 55.

Figure 2-4 View of the SF Oracle RAC cluster



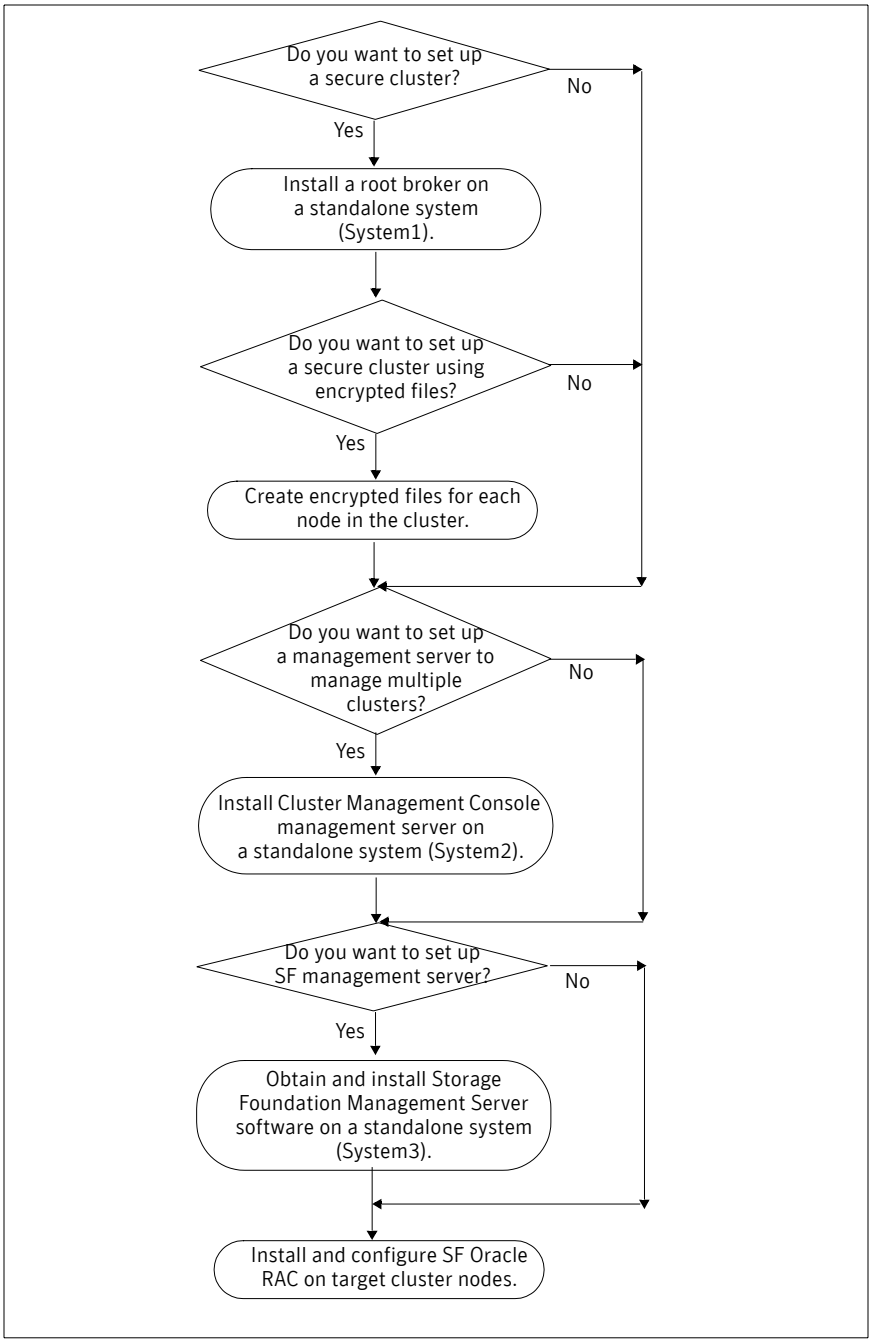
Preparing SF Oracle RAC cluster setup for optional features

After planning the SF Oracle RAC features that you want to configure, you must prepare to configure these features.

See "About SF Oracle RAC component features" on page 47.

Figure 2-5 represents the major tasks and decisions required to install and configure SF Oracle RAC.

Figure 2-5 Workflow for fresh install of SF 5.0 Oracle RAC



Complete the following preparatory tasks based on the SF Oracle RAC features you want to configure:

- "Installing root broker for Symantec Product Authentication Service" on page 58
- "Creating encrypted files for Symantec Product Authentication Service" on page 59
- "Installing the management server for the Veritas Cluster Management Console" on page 61
- "Installing Veritas Storage Foundation Management Server" on page 67

Installing root broker for Symantec Product Authentication Service

Install the root broker only if you plan on using Symantec Product Authentication Service. The root broker administrator must install and configure the root broker before you configure the Authentication Service for SF Oracle RAC. Symantec recommends that you install the root broker on a stable system that is outside the cluster. You can install the root broker on an AIX, HP-UX, Linux, or Solaris system. See *Symantec Product Authentication Service Installation Guide* for more information. You can configure the Authentication Service during or after SF Oracle RAC installation.

See "Symantec Product Authentication Service" on page 47.

To install the root broker

1. Change to the directory where you can start the installsfrac program:

   ```
   # cd cluster_server
   ```

2. Start the Root Broker installation program:

   ```
   # ./installsfrac -security
   ```

3. Select to install the Root Broker from the three choices that the installer presents:

   ```
   [3] Install Symantec Product Authentication Service Root Broker.
   ```

4. Enter the name of the system where you want to install the Root Broker.

   ```
   Enter the system name on which to install Symantec Product
   Authentication Service: venus
   ```

5. Review the output as the installer:
   - checks to make sure that the SF Oracle RAC supports the operating system
   - verifies that you are installing from the global zone (only on Solaris)
   - checks if the system already runs the security package

6 Review the output as the installsfrac program checks for the installed packages on the system.
The installsfrac program lists the packages that will be installed on the system. Press Enter to continue.

7 Review the output as the installer installs the root broker on the system.

8 Enter **y** when the installer prompts you to configure the Symantec Product Authentication Service.

9 Enter a password for the root broker. Make sure the password contains a minimum of five characters.

10 Enter a password for the authentication broker. Make sure the password contains a minimum of five characters.

11 Press Enter to start the Authentication Server processes.

```
Do you want to start Symantec Product Authentication Service
processes now? [y,n,q] y
```

12 Review the output as the installer starts the Authentication Service.

13 If you plan to configure the Authentication Service during SF Oracle RAC installation, choose to configure the cluster in secure mode when the installer prompts you.
See "Configuring SF Oracle RAC Components" on page 90.

Creating encrypted files for Symantec Product Authentication Service

Create encrypted files only if you plan on choosing the semiautomatic mode that uses an encrypted file to configure the Authentication Service. The encrypted files must be created by the administrator on the root broker node. The administrator must create encrypted files for each node that would be a part of the cluster before you configure the Authentication Service for SF Oracle RAC. See *Veritas Cluster Server User's Guide* for more information. You can configure the Authentication Service during or after SF Oracle RAC installation.

See "Symantec Product Authentication Service" on page 47.

The example procedure assumes venus as the root broker node. The example procedure creates encrypted files for nodes galaxy and nebula that would form the SF Oracle RAC cluster rac_cluster101.

To create encrypted files

1 Determine the root broker domain name. Enter the following command on the root broker system:

```
venus> # vssat showalltrustedcreds
```

For example, the domain name would resemble
"Domain Name: root@venus.symantecexample.com" in the output.

2 For each node in the cluster, make sure that you have created an account on root broker system.
For example, to verify on node galaxy:

```
venus> # vssat showprpl --pdrtype root \
--domain root@venus.symantecexample.com --prplname galaxy
```

- If the output displays the principal account on root broker for the authentication broker on the node, then delete the existing principal accounts. For example:

```
venus> # vssat deleteprpl --pdrtype root \
--domain root@venus.symantecexample.com \
--prplname galaxy --silent
```

- If the output displays an error similar to "Failed To Get Attributes For Principal," then the account for given authentication broker is not created on this root broker. Proceed to step 3.

3 Create a principal account for each authentication broker in the cluster. For example:

```
venus> # vssat addprpl --pdrtype root --domain \
root@venus.symantecexample.com --prplname galaxy \
--password password --prpltype service
```

You must use this password that you create in the input file for the encrypted file.

4 Make a note of the following information that is required for the input file for the encrypted file.

- hash - The root hash string that consists of 40 characters, as shown by the command:

```
venus> # vssat showbrokerhash
```

- identity - Authentication broker identity
The value that you provide for **--prplname** in step 3 (for example, galaxy).
- password - Authentication broker password
The value that you provide for **--password** in step 3.
- root_domain - Domain name of the root broker system
The value that you determined in step 1.
- broker_admin_password - Authentication broker password for Administrator account on the node
Provide a password of at least five characters long.

5 For each node in the cluster, create the input file for the encrypted file.

The installer presents the format of the input file for the encrypted file when you proceed to configure the Authentication Service using encrypted file. For example, the input file for authentication broker on galaxy would resemble:

```
[setuptrust]
broker=venus.symantecexample.com
hash=758a33dbd6fae751630058ace3dedb54e562fe98
securitylevel=high

[configab]
identity=galaxy
password=password
root_domain=vx:root@venus.symantecexample.com
root_broker=venus.symantecexample.com:2821
broker_admin_password=ab_admin_password
start_broker=true
enable_pbx=false
```

6 Back up these input files that you created for the authentication broker on each node in the cluster.
Note that for security purposes, the command to create the output file for the encrypted file deletes the input file.

7 For each node in the cluster, create the output file for the encrypted file from the root broker system using the following command.

```
RootBroker> # vssat createpkg --in /path/to/blob/input/file.txt
--out /path/to/encrypted/blob/file.txt --host_ctx AB-hostname
```

For example:

```
venus> # vssat createpkg --in /tmp/galaxy.blob.in \
--out /tmp/galaxy.blob.out --host_ctx galaxy
```

Note that this command creates a encrypted file even if you provide wrong password for “password=” entry, but the encrypted file will fail to install on authentication broker node.

8 After you complete creating output files for the encrypted file, you must copy these files to the installer node.

9 After you have created the encrypted file, you can start the SF Oracle RAC installation and choose to configure the cluster in secure mode.
See “Configuring SF Oracle RAC Components” on page 90.

Installing the management server for the Veritas Cluster Management Console

Install the Cluster Management Console management server only if you plan to centrally manage multiple clusters. Make sure you have a root broker in your domain. SF Oracle RAC clusters need not be secure to configure Cluster Management Console to manage multiple clusters.

See "Veritas Cluster Management Console" on page 51.

Install the Cluster Management Console management server and supporting components on a standalone system (outside any cluster but on the local network). Configure the management server to use a previously installed root broker or install and configure a root broker on the management server host.

You can install the management server on one of the following supported operating systems:

- Installing the management server on Solaris
- Installing the management server on Windows

Refer to the *Veritas Cluster Server Installation Guide* for supported software information for the Cluster Management Console.

Installing the management server on Solaris

You must install the management server on a system outside the cluster. This procedure follows a script of a successful installation. If at any step you experience a result other than the expected result that is documented here, you can click "n" to re-enter information. If you continue to have problems, click "q" to quit the installation and then verify the installation prerequisites.

To install the management server on Solaris

1. Insert the distribution media into the disc drive on the local system. At the command prompt, type the following command to run the setup program:
   ```
   ./installer -rsh
   ```
   The setup program (setup) presents copyright information followed by a menu titled, "Storage Foundation and High Availability Solutions 5.0".

2. Enter **i** to specify a task.
   ```
   Enter a Task: [I,C,L,P,U,D,Q,?] i
   ```
   Setup displays another menu that lists products that are available for installation.

3. Select the menu number that corresponds to Veritas Cluster Management Console.
   ```
   Select a product to install: [1-13,b,q]
   ```
   Setup presents a description of the product.

4. Enter **1** to select a product component.
   ```
   Enter '1' to install the Management Server, '2' to install the
   Cluster
   Connector: [1-2,q] (1) 1
   ```
   Setup presents a message stating that it will install the management server.

5. Enter **y** to verify that the information up to this point is correct.
   ```
   Is this information correct? [y,n,q] (y)
   ```

Setup performs an initial system check of the local system and checks for installed packages on the local system. If these checks are satisfactory, setup lists the packages to be installed.

```
Storage Foundation and High Availability Solutions 5.0
installer will install the following CMC packages:
VRTSat       Symantec Product Authentication Service
VRTSperl     Veritas Perl 5.8.8 Redistribution
VRTSdbms3    Symantec Shared DBMS
VRTSjre15    Veritas Java Runtime Environment Redistribution
VRTSweb      Veritas Java Web Server
VRTScmcm     Veritas Cluster Management Console
VRTScmcdc    Veritas Cluster Management Console Documentation
Press [Return] to continue:
```

6 Press Enter.

You may install Cluster Management Console packages without performing configuration. The setup program gives you the option to configure Cluster Management Console now, and provides instructions for configuring Cluster Management Console later.

7 Enter **y** to configure Cluster Management Console.

```
Are you ready to configure CMC? [y,n,q] (y)
```

8 Enter a unique management server display name, such as:

```
Enter a unique management server display name: [?]
mgmtserver1_sol9
```

9 Enter the network address used by the management server, such as:

```
Enter the network address used by the management server [b,?]
mgmtserver1.symantecexample.com
```

10 When prompted, enter a location for the management server database.

```
Enter the desired location of the database to be used by the
management server [b,?] (/opt/VRTScmc/db)
```

Setup repeats the management server display name, the management server network address, and the database location.

11 Enter **y** to verify that the information up to this point is correct.

```
Is this information correct? [y,n,q,b] (y)
```

Setup describes local user configuration and custom user configuration.

12 Configure a local user or a custom user as the initial management server administrator. This is the first user account that is enabled to log in to the Cluster Management Console.

Make your selection and then specify the following user authentication details:

- For a local user, setup assumes that the domain name is the name of the local system and that the domain type is unixpwd, or UNIX password.

When prompted for the initial management server user name, enter root or another administrator-level user for the local system.

- For a custom user, you must explicitly specify the domain name and the domain type along with the user name. Follow the three separate prompts to enter this user information.

```
Local User:
Configure a user on the local machine as the initial admin user.
Custom User:
Configure a user manually.
1)  Local User
2)  Custom User
Enter '1' to enter the name of a local user, '2' to set up a
custom user:
[1-2,q] (1) 1

Storage Foundation and High Availability Solutions 5.0
Local admin user selection:
To log in to the CMC Management Server, enter the name of a local
user to be set as the administrator. The domain and domain type
will be automatically selected for you.
Enter the initial management server user name: [b,?] (root)
Storage Foundation and High Availability Solutions 5.0
Management Server admin user verification:
Management Server User Name: root
```

13 Enter **y** to verify that the information up to this point is correct.

```
Is this information correct? [y,n,q,b] (y)
```

Setup describes a particular management server service account, which the management server uses for secure internal communications with cluster connector. This account is named CMC_CC@CMC_SERVICES.

14 Enter a password for the management server service account and confirm it at the next prompt.

```
Enter a password for the CMC service account:xxxxx
Confirm the password you entered for the CMC service
account:xxxxx
```

When you install and configure cluster connector, you must provide this same password for the CMC_CC@CMC_SERVICES account.

15 Specify whether or not you want the management server to use a remote root broker for user authentication.

If you have already configured a root broker in your network, Symantec recommends that you enter **y** to use that existing root. Specify the additional details for that remote root broker exactly as specified.

If you do not have a currently-configured root broker, enter **n** to install and configure a root broker on the management server host.

After you enter **y** or **n**, setup installs an authentication broker on the management server and configures it to use whichever root broker you selected. When finished, setup presents:

- Installation progress percentages
- Status for writing the management server configuration file
- Status for creating secure internal service accounts

16 Enter **y** to start Veritas Cluster Management Console processes now.

```
Do you want to start Veritas Cluster Management Console
processes now? [y,n,q,b] (y)
```

Setup presents startup progress percentages and, if successful, displays the following message:

```
Startup completed successfully on all systems.
```

17 Enter an encryption key of at least five characters.

```
Enter five or more characters to be used an encryption key: [b]
xxxxx
This key must be retained in a secure file and referenced using
the -enckeyfile option if the generated responsefile is to be
used again.
Press [Return] to continue:
```

18 Press Enter to continue.

Record the location that setup provides for the installation log files, summary file, and response file. Also ensure that you record the initial admin user information. You *must* use this account to log in to the Cluster Management Console for the first time.

Installing the management server on Windows

You must install the management server on a system outside all clusters. Windows Management Instrumentation (WMI) is a prerequisite for installing and using the management server and cluster connector.

To install WMI

1 Log on as a user that has administrator privileges on the system on which you want to install WMI.

2 On the **Start** menu, click **Settings**, and then click **Control Panel.**

3 In the **Control Panel** window, double-click **Add or Remove Programs**.

4 In the task pane, click **Add/Remove Windows Components**.

5 Click **Management and Monitoring Tools**, then click **Details**.

6 Ensure that the WMI Windows Installer Provider is checked, and then click **OK**.

7 Click **Next**.

8 If prompted, insert the Windows CD and click **OK**.

9 After installation is complete, click **Finish**.

10 Restart your computer.

To install the management server on Windows

1 On the distribution disc, locate the **\installer** directory.

2 Double-click the **setup** file.
Depending upon the operating system, you may or may not receive the following warning message:
```
The publisher could not be verified. Are you sure you want to
run this software?
```
If you receive this message, click **Run**.

3 In the Welcome to the Veritas Cluster Management Console Installation Manager dialog box, read the introduction and then click **Next**.

4 In the Installation and Configuration Options dialog box, click **Install a new management server on the local node**, and then click **Next**.

5 In the Management Server Installation Directory dialog box, leave the default installation path provided in the text box or click **Browse** to search for another installation location. Click **Next** to accept the path.

6 In the Management Server Information dialog box, enter the system name and IP address of the intended management server host.
You cannot change the port specification, 14145, but it is provided to help you to prevent port conflicts when configuring other software. The other ports used by the Cluster Management Console are 8181 (HTTP), 8443 (HTTPS), and 2994 (DBMS; this port can be shared with other Symantec products)

7 In the Database File Path box, leave the default database path provided or click **Browse** to search for another location for the database. Click **Next** to accept the path.

8 In the Services Account Password dialog box, enter a password for the user account that cluster connector uses for management server communications, and then click **Next**.
Record the password that you enter in a safe place. You must use it again whenever you install or configure cluster connector.

9 In the User Credential Confirmation dialog box, leave the automatically-detected user information provided or specify another user name, domain, and domain type.

This user becomes the initial management server user. You must provide the credentials entered at this step when logging in to the management server for the first time.

10 In the Summary dialog box, review the information you have specified and, if satisfactory, click **Next** to accept it and start the installation.
The Installing Veritas Cluster Management Console dialog box displays a progress bar and a status message window for the installation.

11 When you receive the following message, click **Next**:
`"Done deleting installation files from node...,"`

12 In the Completed the Symantec Veritas Cluster Management Console Installation Manager dialog box, review the information about how to connect to the management server and log in for the first time. Record this information in a safe place and then click **Finish**.

13 Note the log file locations. The installer creates log files at the following locations:
- Installation logs – `C:\Documents and Settings\All Users\Application Data\Veritas\Cluster Management Console`. The file names are Install_GUI_0.log and Install_MSI_0.log.
- Management server logs – `C:\Program Files\Veritas\Cluster Management Console\log`

Installing Veritas Storage Foundation Management Server

Obtain the Storage Foundation Management Server software and install SF Management software on a system outside the cluster. For information on ordering SF Management Server, visit:

www.symantec.com/enterprise/sfms

Refer to the Storage Foundation Management Server documentation for details.

SF Oracle RAC prerequisites

Verify the requirements for your configuration before installing or upgrading SF Oracle RAC.

System requirements

Make sure that you have the correct equipment to install SF Oracle RAC.

Table 2-1

Item	Description
SF Oracle RAC systems	Two to eight systems with two or more CPUs at 2GHz or higher.
RAM	Each SF Oracle RAC system requires 2 GB or more of physical memory.
Network links	Two or more private links and one public link. Links must be 100BaseT or Gigabit Ethernet directly linking each node to the other node to form a private network that handles direct inter-system communication. Symantec recommends Gigabit Ethernet using enterprise-class switches for the private links.
DVD drive	One drive that is accessible to all nodes in the cluster.
Fibre channel or SCSI host bus adapters	SF Oracle RAC requires at least one built-in SCSI adapter per system to access the operating system disks, and at least one additional SCSI or Fibre Channel Host Bus Adapter per system for shared data disks.
Disks	Typical SF Oracle RAC configurations require that shared disks support applications that migrate between systems in the cluster. The SF Oracle RAC I/O fencing feature requires that all disks used as data disks or as coordinator disks must support SCSI-3 Persistent Reservations (PR). **Note:** The coordinator disk does not store data, so configure the disk as the smallest possible LUN on a disk array to avoid wasting space. "Checking shared disks for SCSI-3 support" on page 89
Disk space	SF Oracle RAC space requirement: ■ total: 2.6 G ■ `/opt`: 1.4 G ■ `/usr`: 200 KB ■ `/tmp`: 512 MB ■ `/var`: 32 MB ■ `/var/tmp`: 700 MB
Swap space	Two times the main memory.

Review the current compatibility list to confirm compatibility of your hardware:
http://support.veritas.com/docs/283161

Checking for Current RAM, Swap Space, and /tmp

1 To check the current RAM installed on a system, enter:

```
# /usr/sbin/lsattr -E -l sys0 -a realmem
```

If the current RAM is insufficient, you must install more memory before proceeding with the installation.

2 To check the amount of swap space configured for a system, enter:

```
# /usr/sbin/lsps -a
```

Refer to the operating system documentation for information about configuring swap space. Oracle recommends 1 GB of swap space. See the *Oracle Database Administrator's Reference for UNIX* for more information.

a To check for the current space available in the `/tmp` directory, enter:

```
# df -k /tmp
```

If the space remaining in /tmp is insufficient, complete *one* of the following operations:

- Delete unnecessary files in `/tmp`, and recheck the available space.
- Extend the file system that contains the `/tmp` directory.
- Set the TEMP and TMPDIR environment variables when setting the Oracle user's environment.

Software requirements

Software versions that SF 5.0 Oracle RAC supports include:

Oracle RAC	■ Oracle9i Release 2 ■ Oracle10*g* Release 1 ■ Oracle10*g* Release 2
AIX operating system	See "Supported operating systems" on page 69.
VCS, VxVM, VxFS, VVR	Use only versions of VCS, VxVM, VxFS, and VVR provided on the software disc. Remove other versions before you install the software from the SF Oracle RAC product disc.

To verify the latest information on support for Oracle database versions, see the Technical Support TechNote:

http://support.veritas.com/docs/282024

Supported operating systems

Within a cluster, all nodes must use the same operating system version and patch level. Run SF 5.0 Oracle RAC on these operating systems at the suggested patch levels.

- AIX 5.2 ML6 (legacy) or later
- AIX 5.3 TL4 with SP 4
  SP 4 was not available at the time of this release. Veritas 5.0 products also operate on AIX 5.3 with Service Pack 3, but you must install an AIX interim fix. See the following TechNote for information on downloads, service pack availability, and other important issues related to this release:
  http://support.veritas.com/docs/282024

APAR and Fileset Information

AIX filesets required for Oracle installation are:

- bos.adt.libm
- perfagent.tools
- bos.perf
- Java14
- rsct.core
- rsct.compat.basic
- rsct.compat.clients
- rsct.basic
- rsct.clients

The Oracle Agent on AIX requires a certain runtime fileset level for the xlC.rte and xlC.aix50.rte filesets:

xlC.rte	7.0.0.1	COMMITTED	C Set	++	Runtime
xlC.aix50.rte	7.0.0.4	COMMITTED	C Set	++	Runtime for AIX 5.0

Performing preinstallation tasks

Complete these tasks before installing SF Oracle RAC:

- ✔ Obtaining license keys
- ✔ Synchronizing cluster nodes
- ✔ Setting up inter-system communication
- ✔ Setting up shared storage
- ✔ Setting up environment variables
- ✔ Preparing information for the configuration phase of installsfrac.

Obtaining license keys

SF Oracle RAC includes a License Key certificate. The certificate specifies the product keys and the number of product licenses purchased. A single key enables you to install the product on the number and type of systems for which you purchased the license. A key may enable the operation of more products than are specified on the certificate; however, you are legally limited to the number of product licenses purchased.

The product installation procedure includes instructions on how to activate the key. If you encounter problems while licensing this product, visit the Symantec licensing support website at:

http://www.veritas.com/buy/vLicense/vLicenseHome.jhtml

The VRTSvlic package enables product licensing. After the VRTSvlic is installed, the following commands and their manual pages are available on the system:

vxlicinst	Installs a license key for a Symantec product
vxlicrep	Displays currently installed licenses
vxlictest	Retrieves features and their descriptions encoded in a license key

Even though other products are included on the enclosed software discs, you can only install the Symantec software products for which you have purchased a license.

Synchronizing cluster nodes

Symantec requires all cluster nodes have the same time. If you do not run the Network Time Protocol (NTP) daemon, make sure to synchronize the time settings on each node.

Setting up inter-system communication

If you configured `ssh` (SSH client) for the cluster nodes, the installation program can install SF Oracle RAC as long as `ssh` commands between nodes can execute without password prompting and confirmation.

If you did not configure `ssh`, enable each node to have remote `rsh` access to the other nodes during installation and disk verification.

On each node, placing a “+” character in the first line of the `/.rhosts` file gives remote access to the system running the install program. You can limit the remote access to specific nodes. Refer to the manual page for the `/.rhosts` file for more information.

Remove the remote `rsh` access permissions after the installation and disk verification process.

Setting up shared storage for I/O fencing

You need to set up shared storage so that it is visible to the SCSI layer from all the nodes in the cluster. The shared storage that you add for use with SF Oracle RAC software must support SCSI-3 persistent reservations, a functionality that enables the use of I/O fencing.

For troubleshooting, see "Shared disks not visible" on page 420.

Setting up environment variables

Set up the PATH and MANPATH variables prior to installing SF Oracle RAC.

Setting the PATH Variable

The installation and other commands are located in various directories. If necessary, add these directories to your PATH environment variable on each system:

For Bourne Shell (`sh` or `ksh`), type:

```
# PATH=/usr/sbin:/sbin:/usr/bin:/usr/lib/vxvm/bin:\
/opt/VRTSvxfs/sbin:/opt/VRTSvcs/bin:/opt/VRTS/bin:\
/opt/VRTSvcs/rac/bin:/opt/VRTSob/bin:$PATH; export PATH
```

For C Shell (`csh`), type:

```
# setenv PATH /usr/sbin:/sbin:/usr/bin:/usr/lib/vxvm/bin:\
/opt/VRTSvxfs/sbin:/opt/VRTSvcs/bin:/opt/VRTS/bin:\
/opt/VRTSvcs/rac/bin:/opt/VRTSob/bin:$PATH
```

Do not define paths for the `root` user to a cluster file system in the `LIBPATH` variable. You can define `$ORACLE_HOME/lib` in *LIBPATH* for the `oracle` user.

The path defined as `/opt/VRTSob/bin` is optional unless you choose to install Veritas Enterprise Administrator.

Setting the MANPATH Variable

Set the MANPATH variable to enable viewing manual pages.

For the Bourne Shell (`bash`, `sh` or `ksh`), type:

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

For the C Shell (`csh`), type:

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

Gathering information to install and configure SF Oracle RAC

The SF Oracle RAC installation and configuration program prompts you for information about some SF Oracle RAC components. The program provides default values for some information, which you can choose to use. Keep the following required information at hand.

Information to install SF Oracle RAC filesets

- System names on which to install SF Oracle RAC
  Example: **galaxy**, **nebula**
- License keys
  License keys could be one of the following types:
  - Valid license keys for each system in the cluster
  - Valid site license key
  - Valid demo license key

  If you want to configure Veritas Volume Replicator to enable disaster recovery, you must enter appropriate license keys.
  See "Obtaining license keys" on page 71.
- Do you want to install required SF Oracle RAC filesets or all SF Oracle RAC filesets?
  Install only the required filesets if you do not want to configure any optional components or features.
  Default option is to install all filesets.

Information to configure Veritas Cluster Server component

- Name of the cluster
  The name must begin with a letter of the alphabet (a-z, A-Z) and contain only the characters a through z, A through Z, and 1 through 0, hyphen (-), and underscore (_).
  Example: **rac_cluster101**
- Unique ID number for the cluster
  Number in the range of 0-65535. Within the site containing the cluster, each cluster must have a unique ID.
  Example: **101**
- Device names of the NICs used by the private networks among systems
  Do not enter the network interface card that is used for the public network.

Example: **en1, en2**
The interface names associated with each NIC for each network link must be the same on all nodes.

Information to configure SF Oracle RAC clusters in secure mode

- Which mode do you want to choose to configure Authentication Service?
  The installer provides you the following three modes to configure Authentication Service in the SF Oracle RAC clusters:
  - automatic mode
  - semiautomatic mode using encrypted files
  - semiautomatic mode without using encrypted files

  Default option is automatic mode.
  See "Symantec Product Authentication Service" on page 47.
- Host name of the Symantec Product Authentication Service Root Broker System
  Example: **venus**

Information to add SF Oracle RAC users

You need add SF Oracle RAC users now if you configured SF Oracle RAC cluster in secure mode.

- User name
  Example: **smith**
- User password
  Enter the password at the prompt.
- User privilege
  Users have three levels of privileges: A=Administrator, O=Operator, or G=Guest.
  Example: **A**

Information to configure Cluster Management Console cluster connector

- Management Server network address for Cluster Management Console
  Example: **mgmtserver1.symantecexample.com**
- Cluster Management Console service account password
  You must have set this account password while installing the management server.

- Root hash of the management server
  You can use `vssat showbrokerhash` command and copy the root hash of the management server.

Information to configure Cluster Mamagement Console

- Name of the public NIC for each node in the cluster
  The device name for the NIC that provides public network access.
  Example: **en0**
- Virtual IP address of the NIC for Cluster Management Console (CMC)
  This virtual IP address becomes a resource for use by the ClusterService group that includes the CMC. The “Cluster Virtual IP address” can fail over to another cluster system, making the Web Console highly available.
  Example: **10.10.12.1**
- Netmask for the virtual IP address
  The subnet used with the virtual address.
  Example: **255.255.240.0**

Information to configure SMTP email notification

- Domain-based address of the SMTP server
  The SMTP server sends notification email about the events within the cluster.
  Example: **smtp.symantecexample.com**
- Email address of each SMTP recipient to be notified
  Example: **john@symantecexample.com**
- Minimum severity of events for SMTP email notification
  Events have four levels of severity: I=Information, W=Warning, E=Error, and S=SevereError.
  Example: **E**

Information to configure SNMP trap notification

- Port number for the SNMP trap daemon
  Default port number is 162.
- Machine name for each SNMP console
  Example: **saturn**
- Minimum severity of events for SNMP trap notification
  Events have four levels of severity: I=Information, W=Warning, E=Error, and S=SevereError.

Example: **E**

Information to configure global clusters

- Name of the public NIC
  You may use the same NIC that you configured for the ClusterService group. Otherwise, you must specify appropriate values for the NIC when you are prompted.
  Example: **en0**
- Virtual IP address of the NIC
  You may use the same virtual IP address that you configured for the ClusterService group. Otherwise, you must specify appropriate values for the virtual IP address when you are prompted.
  Example: **10.10.12.1**
- Netmask for the virtual IP address
  You may use the same netmask as configured for the ClusterService group. Otherwise, you must specify appropriate values for the netmask when you are prompted.
  Example: **255.255.240.0**
- [HP-UX only] NetworkHosts IP addreses
  You may use the same NetworkHosts IP address as configured for the ClusterService group. Otherwise, you must specify appropriate values for the netmask when you are prompted..
  Example: **10.10.12.2**

Information to configure Cluster Volume Manager

- CVM cluster reconfiguration timeout in seconds
  Default is 200.

Information to configure Veritas Volume Replicator

- Frequency of VVR statistics collection
  Default is 10 seconds.
- Number of days to preserve the collected statistics
  Default is 3 days.

Information to configure I/O fencing

- Name of three disks that will form the coordinator disk group
  Example: **rhdisk75, rhdisk76, rhdisk77**

- DMP nodes names for each disk in the coordinator disk group (if using DMP)
  Example: **/dev/vx/dmp**

Starting SF Oracle RAC processes

You have the option of starting the SF Oracle RAC processes during the installation procedure.

Default disk group

As some VxVM commands require a disk group be specified, the installation enables you to register the name of the default VxVM disk group (which can be created later) on each eligible cluster system. Setting up a default disk group is optional.

In VxVM 4,0 and higher, commands that affect the contents of a disk group require you to specify a disk group using the `-g` option. If you specify a disk group in the *VXVM_DEFAULTDG* environment variable, or you configure the default disk group, you do not need to use the `-g` option for disk group operations on that disk group.

Setting umask for root user

Set umask for root user to “022” before installing SF Oracle RAC. Type:

```
# umask 0022
```

About CVM and CFS in an SF Oracle RAC environment

You can review concepts on CVM, CFS, and Oracle before installing SF Oracle RAC to better understand the overall setup and configuration of the product.

About CVM

Review CVM configuration differences from VxVM and CVM recovery operations. For introductory information on CVM, see "Cluster Volume Manager" on page 28.

CVM configuration differences

CVM configuration differs from VxVM configuration in these 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.

CVM recovery

When a node leaves a cluster, it can leave some mirrors in an inconsistent state. The membership change is communicated through GAB to the `vxconfigd` daemon, which automatically calls the `vxrecover` utility with the `-c` option when necessary.

CVM supports both the FastResync option and dirty region logging (DRL) as optional features to improve resynchronization performance. FastResync improves performance when reorganizing volumes (moving, splitting, and joining disk groups). This is useful when performing off-host processing. DRL speeds up resynchronization after a node failure.

Special considerations exist when using the DRL in an SF Oracle RAC environment. As in a non-clustered environment, the DRL in clusters exists on a log subdisk in a mirrored volume. The size of the DRL in clusters is typically larger than in non-clustered systems. The log size depends on the volume size and the number of nodes. The `vxassist` command automatically imports a sufficiently large DRL.

You can reimport a private disk group as a shared disk group but the DRL for any mirrored volume in the disk group is probably too small to accommodate maps for all the cluster nodes. Adding nodes to the cluster can also result in too

small a log size. In this situation, VxVM marks the log invalid and performs full volume recovery instead of using DRL.

About CFS

Review CFS File System benefits, CFS configuration differences from VxFS and CFS recovery operations. For introductory information on CFS, see "Cluster File System" on page 30.

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 these 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.
- 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. `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.

Coordinating CVM and CFS configurations

After installing SF Oracle RAC, a VCS cluster attribute (`HacliUserLevel`) is set to give root the ability to run commands on remote systems by way of the cluster interconnect. CFS takes advantage of this mechanism to enable you to perform file system operations requiring the primary node be initiated on secondary nodes and carried out on the primary node transparently.

If you reset this attribute, be aware of which node is the primary for certain file system operations and perform those tasks from that node. Unlike a non-RAC environment, you cannot run a sequence of VxVM and VxFS commands, such as resizing a volume and a file system, on the same node unless it is both the CVM master and CFS primary node.

About shared disk groups

This section highlights general information to refer to when dealing with disk groups and volumes. Refer to the Veritas Volume Manager documentation for complete details on creating and managing shared disk groups.

Viewing information on a disk group

To display information about a specific disk group, type:

```
vxdg list disk_group
```

Checking the connectivity policy on a shared disk group

By default, the connectivity policy for a shared disk group is set to “global.” This setting protects against possible data corruption and causes all nodes to detach from the disk group when any node reports a disk failure for that disk group.

The output of the `vxdg list` *`shared_disk_group`* command includes the following line:

```
detach-policy: global
```

To change the connectivity policy for a disk group from “local” to “global,” type:

```
# vxedit -g shared_disk_group set diskdetpolicy=global
shared_disk_group
```

Determining whether a node is CVM master or slave

To determine whether a node is the CVM master or slave, type:

```
# vxdctl -c mode
```

On `nebula`, which is the slave, the output shows:

```
mode: enabled: cluster active - SLAVE
master: galaxy
```

On `galaxy`, which is the master, the output shows:

```
mode: enabled: cluster active - MASTER
master:galaxy
```

Deporting and importing shared disk groups

Shared disk groups in an SF Oracle RAC environment are configured for “Autoimport” at the time of CVM startup. If the user manually deports the shared disk group on the CVM master, the disk group is deported on all nodes. To reimport the disk group, the user must import the disk group as a shared group from the CVM master.

To deport a shared disk group, use the following command on the CVM master:

```
vxdg deport shared_disk_group
```

To import a shared disk group, use the following command on the CVM master:

```
vxdg -s import shared_disk_group
```

To import a disk group as a standalone disk group, deport it from the CVM master and use the following command on any node:

```
vxdg -C import shared_disk_group
```

To reimport a disk group as a shared disk group, deport it from the standalone node and use the following command on the CVM master node:

```
vxdg -C -s import shared_disk_group
```

Reviewing limitations of shared disk groups

The cluster functionality of VxVM (CVM) does not support RAID-5 volumes or task monitoring for shared disk groups in a cluster. These features can function in private disk groups attached to specific nodes of a cluster. Online relayout is available provided it does not involve RAID-5 volumes.

CVM only provides access to raw device; it does not support shared access to file systems in shared volumes unless you install and configure the appropriate software, such as CFS.

About raw volumes versus 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=dba owner=dba 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.

Chapter 3

Installing and Configuring SF Oracle RAC Software

After reviewing the requirements and planning information, use this chapter to install and configure SF Oracle RAC on clean systems. For planning information:

See "Preparing to install SF Oracle RAC" on page 41.

High-level objectives and required tasks to complete each objective:

Installing the software

To install the SF Oracle RAC software, you may use the Symantec common product installer with the `-installonly` option or the `installsfrac` script with `-installonly` option.

The common product installer offers a high-level approach to installing multiple products along with Symantec Product Authentication Service, Veritas Cluster Management Console, and Veritas Central Management Server. Each of these products and features are covered in depth in their respective product guides.

The common product installer is the recommended method to license and install the product. The installer also enables you to configure the product, verify preinstallation requirements, and view the product's description. At most points

during an installation, you can type **b** ("`back`") to return to a previous section of the installation procedure. The back feature of the installation scripts is context-sensitive, so it returns to the beginning of a grouped section of questions. If an installation procedure hangs, use **Control–c** to stop and exit the program. There is a short delay before the script exits.

The `installsfrac -installonly` script offers a more direct approach to specifically installing SF Oracle RAC. The script takes the user only through the installation of packages.

Note: If you have obtained a Veritas product from an electronic download site, the single product download files do not contain the `installer` installation script, so you must use the product installation script to install the product. For example, if you download Veritas Storage Foundation for Oracle RAC, use the `installsfrac` script instead of the `installer` script.

Note: Configuring the software with the `installsfrac -configure` script occurs *after* you install the product and run required system checks.

To install SF Oracle RAC

1. Insert the disc containing the Veritas SF Oracle RAC software in a disc drive connected to one of the nodes for installation.
2. Log in as `root` user.
3. Mount the software disc:

   ```
   # mkdir -p /dvdrom
   # mount -V cdrfs -o ro /dev/cd0 /dvdrom
   ```

4. Navigate to the directory containing the installation program:

   ```
   # cd /cdrom/dvd1/sfrac
   ```

5. Start the `installsfrac` script:

   ```
   # ./installsfrac -installonly
   ```

   By default, the installsfrac program uses SSH for remote communication. However, to use RSH, specify the -rsh option with the installsfrac program.

   ```
   # ./installsfrac -rsh -installonly
   ```

6. Enter the names of the nodes separate by spaces where you want to install the software:

   ```
   Enter the system names separated by spaces on which to
   install SFRAC: galaxy nebula
   ```

7 After the script verifies that the local node running the script can communicate with remote nodes and that `VRTScpi` and `VRTSvlic` are present on each node, enter the license key for SF Oracle RAC.

You can also enter keys for other products:

```
Enter a SFRAC license key for galaxy:
XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXX
XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXX successsully registered on
on galaxy
SFRAC license registered on galaxy

Do you want to enter another license key for galaxy?
[y,n,q,?] (n)
```

Note: At this time, you can add another license. If you desire the Veritas Volume Replicator option for global clustering and disaster recovery, you can add it now.

8 Respond to the script as it verifies system requirements and installs the software. If requirements for installation are not met, the utility stops and indicates the actions required to proceed with the process.

- The script determines if any filesets are already installed.
- The script checks whether the required operating system patches are installed. The installer notes missing patches and recommends to stop the installation and install them.
  See "Software requirements" on page 69.
- The script checks for the required file system space.
- The script checks for the presence of processes that could conflict with the installation.

9 At the conclusion of the installation, the installer displays information about where to find installation log files:

```
...
Installing SFRAC: 100%

Installation completed successfully on all systems Installation
log files, summary file, and response file are saved at:

        /opt/VRTS/install/logs/installsfrac-lB2coI
```

Performing basic system checks

While some system checks are required prior to configuring SF Oracle RAC, other checks are optional or needed only when troubleshooting an issue. Use the SF Oracle RAC configuration program initially to check:

- Setup for LLT (optional): verifies the private interfaces on all nodes have the same settings for media speed and jumbo frames on the link.
- Shared disks for I/O fencing (required at this point)

Running an optional system check for LLT

Run this check to ensure the proper setup of LLT.

To run an optional system check for LLT

1. Log in as `root` user.
2. Navigate to the directory containing the installation program:
   ```
   # cd /cdrom/dvd1/sfrac
   ```
3. Launch the SF Oracle RAC configuration menu:
   ```
   # ./installsfrac -configure
   ```
   By default, the installsfrac program uses SSH for remote communication. However, to use RSH, specify the -rsh option with the installsfrac program.
   ```
   # ./installsfrac -rsh -configure
   ```
4. Enter the system names.
   ```
   Enter the system names separated by spaces on whcih to configure
   SFRAC: galaxy nebula
   ```
5. From the main menu, select **Check systems for SFRAC **INSTRUCTIONS ONLY ****.
6. Select **Check LLT links **INSTRUCTIONS ONLY****.
7. The installer lists the conditions for LLT links for each of the cluster systems and shows example commands for checking and changing settings. You must log in to each cluster system to make the checks. For example:
   ```
   ........................................
   Before continuing, login to all cluster nodes and check LLT
   links.Each LLT link must:
       * Not share a subnet with other LLT links on that system.
       * Have speed and autonegotiate settings matching the switch
       port
       * Have same jumbo frame settings
       * Must have unique MAC addresses

   Example:
   ```

```
    /usr/sbin/lsattr -l 'ent3' -E # to query speed and
autonegotiate
    /usr/sbin/lsattr -l 'ent3' -E # to query jumbo frame
.............................................................
```

Auto-negotiation, media speed, and jumbo frame settings on private NICs

For optimal LLT (Low Latency Transport) communication among the cluster nodes, the interface cards on each node must use the same media speed settings. Also, the settings for switches or hubs used for the interconnects must match that of the interface cards. Incorrect settings can cause poor network performance or even network failure.

Guidelines for Auto-negotiation, media speed, and jumbo frame settings for LLT interconnects:

- If you have hubs or switches for LLT interconnects, we recommend using the Auto_Negotiation media speed setting on each Ethernet card on each node.
- If you have hubs or switches for LLT interconnects and you do not use the Auto_Negotiation media speed setting, set the hub or switch port to the same setting as that used for the cards on each node.
- If you use directly connected Ethernet links (using crossover cables), set the media speed to the highest value common to both cards, typically 100_Full_Duplex.
- Settings on Ethernet cards for jumbo frames must match that of the switches or hubs. Symantec does not recommend use of jumbo frames in an SF Oracle RAC environment.
- Symantec does not recommend using dissimilar network cards for private links.

Displaying and setting the Ethernet media speed

The following paragraphs describe displaying information about the current settings for interface cards and how to modify them if necessary.

To display and set the Ethernet auto-negotiation setting and the media speed

1. Use the command lsattr E1 <device_name> to display current capabilities of the interface card. For example:

   ```
   # lsattr -R -l <device_name> -a media_speed
   ```

   If you receive an error such as:

   ```
   lsattr: 0514-528 The "media_speed" attribute does not exist
   in the predefined device configuration database.
   ```

   The card is a 10 Mbps card, capable of only 10 Mbps / half duplex.

Example output of a 100 Mbps card resembles:

```
10_Half_Duplex
10_Full_Duplex
100_Half_Duplex
100_Full_Duplex
Autonegotiation
```

2 To display the current settings of the card, enter:

```
# lsattr -EH -l <device_name> -a media_speed
```

3 To change the setting of the card, use the chdev command. For example, where the card is ent0 and the desired speed is 100 half duplex, enter:

```
# chdev -P -l ent0 -a media_speed=100_Half_Duplex
```

The change in value does not take effect until the system is restarted or the device is rediscovered.

Checking shared disks for SCSI-3 support

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

- Data disks that store shared data
- Coordinator disks that act as a global lock during membership changes. Coordinator disks are small LUNs (typically three per cluster)

When to check the shared disks for SCSI-3 support

You can choose to verify that the disks you plan to use for shared data storage or for coordinator disks at this time, before you configure the SF Oracle RAC components, or later, after configuring the components. In either case, review the guideline included in the installerand refer to the procedures in the section "Setting up I/O fencing" on page 105.

If you test the disks now and discover that the disks are not SCSI3 compliant, you can delay the configuration tasks until you obtain and verify compliant disks.

If, however, you have high confidence that the disks you plan to use are compliant, you can skip testing now and proceed with the configuration SF Oracle RAC components. You can test the storage later.

Checking that disks support SCSI-3 involves:

- Reviewing the guidelines for checking the disks
- Verifying that nodes have access to the same disk
- Using the `vxfentsthdw` utility to perform the check

Viewing guidelines for checking SCSI-3 support

Use the SF Oracle RAC configuration program to review this system check for SCSI support.

To view guidelines for checking SCSI-3 support

1. Log in as `root` user.
2. Navigate to the directory containing the installation program:
   ```
   # cd /cdrom/storage_foundation_for_oracle_rac
   ```
3. Start the `installsfrac` script:
   ```
   # ./installsfrac -configure
   ```
   By default, the installsfrac program uses SSH for remote communication. However, to use RSH, specify the -rsh option with the installsfrac program.

```
# ./installsfrac -rsh -configure
```

4 From the main menu, select **Check systems for SFRAC** .

5 Select **Check I/O fencing disks** .

6 Review the brief overview on testing disks for SCSI-3 compliance. If you desire to test the disks at this time, proceed to "Setting up I/O fencing" on page 105 and use the procedures:

- "Verifying the nodes see the same disk" on page 105
- "Testing the disks using the vxfentsthdw script" on page 107

Configuring SF Oracle RAC Components

The `installsfrac -configure` script prompts you for information necessary to set up and configure the cluster. You can also set up optional features including Symantec Authentication Services, Cluster Management Console, SMTP and SNMP notification, Storage Foundation Management Server, and various options for Veritas Volume Manager. Details on all of these products and features are in their respective product guides and may require initial setup using the Symantec product installer menu.

Tasks for configuring the cluster may include:

- "Configuring the cluster"
- "Configuring the cluster in secure mode"
- "Adding SF Oracle RAC users"
- "Configuring cluster connector"
- "Configuring the Cluster Management Console"
- "Configuring SMTP email notification"
- "Configuring SNMP trap notification"
- "Configuring the global cluster option"
- "Setting permissions for database administration"
- "Configuring the cluster volume manager"
- "Configuring VVR"

Configuring the cluster

Enter a cluster name and ID to perform the basic cluster configuration.

To configure the cluster

1. If you ran the `installsfrac -installonly` utility earlier, or if you ran the `installsfrac` utility but declined to configure SF Oracle RAC at that point, start the `installsfrac -configure` script:
   a. Log in as `root` user.
   b. Navigate to the directory containing the installation program:
      # **cd /cdrom/storage_foundation_for_oracle_rac**
   c. Start the configuration:
      # **./installsfrac -configure**

      By default, the installsfrac program uses SSH for remote communication. However, to use RSH, specify the -rsh option with the installsfrac program.
      # **./installsfrac -rsh -configure**
   d. Enter the system names, separated by spaces, on which to configure SF Oracle RAC.
2. From the main menu, select **Configure SFRAC.**
3. Select **Configure VCS, CVM and CFS.**
4. Enter the cluster details.
   a. Enter the unique cluster name. For example, type:

      rac_cluster101
   b. Enter the unique cluster ID between 0-65535. For example, type:
      101
   c. Enter the NICs for private heartbeat links.en0 is typically the network interface card for only the public network. In this example, `en1` and `en2` are the private heartbeat NICs on all nodes.

      Note: Oracle RAC requires the use of the same heartbeat interfaces on all hosts in the cluster.

```
Discovering NICs on galaxy ... discovered en0 en1 en2
    en3 en4
Enter the NIC for the first private heartbeat NIC on
    galaxy: [b,?] en1
Would you like to configure a second private heartbeat
    link? [y,n,q,b,?] (y)
Enter the NIC for the second private heartbeat link on
    galaxy: [b,?] en2
Would you like to configure a third private heartbeat
    link? [y,n,q,b,?] (n)
```

```
Are you using the same NICs for private heartbeat links
    on all systems? [y,n,q,b,?] (y)
```

You may use NICs with different device names on some nodes. If necessary, indicate the NICs are different when prompted.

d Verify the summary information.

5 Choose to configure SF Oracle RAC with Symantec Product Authentication Service, an option that encrypts all inter-node communication and verifies users with security credentials. See "Configuring the cluster in secure mode" on page 92.

If you decline to set up these services, specify whether you want to set the user name and password for the Administrator, and whether you want to add another user to the cluster.

Go to "Adding SF Oracle RAC users" on page 93.

Configuring the cluster in secure mode

Before you configure a cluster in a secure mode, make sure you have installed a root broker on another stable system. Also, make sure you meet the requirements for automatic or semiautomatic mode of configuration.

See "Symantec Product Authentication Service" on page 47.

To configure the cluster in secure mode

1 Choose whether to configure SF Oracle RAC to use Symanted Product Authentication Service.

- If you want to configure Authentication Service, make sure that you have installed the root broker, and answer **y**.
- If you decline to configure Authentication Service, answer **n** and proceed to adding SF Oracle RAC users.

  See "Adding SF Oracle RAC users" on page 93.

```
Would you like to configure VCS to use Symantec Security
Services? [y,n,q] (n) y
```

2 Select one of the options to configure security.

```
Select the Security option you would like to perform [1-3,q,?]
```

Based on the mode of configuration you want to use, enter one of the following:

Option	Tasks

1. Automatic configuration	Enter the name of the root broker system when prompted. Requires remote access to the root broker. Review the output as the installer verifies communication with the root broker system, checks vxatd process and version, and checks security domain.
2 . Semi-automatic using encrypted files	Enter the path of the file for each node when prompted.

3. Semi-automatic entering authentication information at installer prompts	Enter the following root broker information as the installer prompts you:
	`Enter root Broker name:` **`venus.symantecexample.com`**
	`Enter root broker FQDN: [b] (symantecexample.com)` **`symantecexample.com`**
	`Enter root broker domain: [b] (root@venus.symantecexample.com)` **`root@venus.symantecexample.com`**
	`Enter root broker port: [b] (2821)` **`2821`**
	`Enter path to the locally accessible root hash [b] (/var/tmp/installvcs-1Lcljr/root_hash)` **`/root/root_hash`**
	Enter the following authentication broker information as the installer prompts you for each node:
	`Enter authentication broker principal name on north [b] (north.symantecexample.com)` **`north.symantecexample.com`**
	`Enter authentication broker password on north:`
	`Enter authentication broker principal name on south [b] (south.symantecexample.com)` **`south.symantecexample.com`**
	`Enter authentication broker password on south:`

3 After configuring the cluster in secure mode, proceed to configure the Cluster Management Console cluster connector.
See "Configuring cluster connector" on page 94.

Adding SF Oracle RAC users

If you have enabled Symantec Product Authentication Service, you need not add SF Oracle RAC users. Proceed to configure the Cluster Management Console. Otherwise, on systems operating under an English locale, you can add SF Oracle RAC users at this time.

"Configuring the cluster in secure mode" on page 92

"Configuring cluster connector" on page 94

To add SF Oracle RAC users

1. Review the required information to add SF Oracle RAC users.
2. Reset the password for the Admin user, if necessary.
3. To add a user, enter **y** at the prompt.
4. Enter the user's name, password, and level of privileges.

```
Enter the user name: [?] smith
Enter New Password:*******

Enter Again:*******
Enter the privilege for user smith (A=Administrator, O=Operator,
G=Guest): [?] a
```

5. Enter **n** at the prompt if you have finished adding users.

```
Would you like to add another user? [y,n,q] (n)
```

6. Review the summary of the newly added users and confirm the information.

Configuring cluster connector

If you configured the Cluster Management Console management server to centrally manage this cluster, you can now configure cluster connector for the buffering feature. If a firewall exists between the management server and this cluster, then you must configure cluster connector to enable centralized management. Make sure you meet the prerequisites to configure cluster connector.

See "Veritas Cluster Management Console" on page 51.

To configure cluster connector

1. Review the information to configure Cluster Management Console.
2. Choose whether to configure cluster connector or not. Do one of the following:
   - To configure cluster connector on the systems, press Enter.

     ```
     Do you want this cluster to be managed by a management
     server? Enter 'y' if you have set up a management server.
     [y,n,q] (y) y
     ```

   - To skip configuring cluster connector and advance to configuring Cluster Management Console for local cluster management, enter **n**.

     See "Configuring the Cluster Management Console" on page 95.
3. Review the required information to configure cluster connector.
4. Enter the Management Server network address for Cluster Management Console.

```
Enter the network address used by the management server [?]
(north) mgmtserver1.symantecexample.com
```

5 Verify and confirm the management server information.

6 Enter the following information that is required to securely communicate with the management server.

- Password for the service account that is created during the management service installation
- Hash of Cluster Management Console management server's root broker

7 Verify and confirm the information.

Configuring the Cluster Management Console

If you want to locally manage this cluster, then you must configure the Cluster Management Console. Note that this cluster can also be a part of the centrally managed clusters.

See "Veritas Cluster Management Console" on page 51.

To configure the Cluster Management Console

1 Review the required information to configure the Cluster Management Console.

2 Choose whether to configure the Cluster Management Console or not. Do one of the following:

- To configure the Cluster Management Console on the systems, press **Enter**.

  ```
  Do you want to configure the Cluster Management Console
  [y,n,q](y)
  ```

- To skip configuring the Cluster Management Console and advance to configuring SMTP, enter **n**.

  See "Configuring SMTP email notification" on page 96.

3 Confirm whether you want to use the discovered public NIC on the first system. Do one of the following:

- If the discovered NIC is the one to use, press **Enter**.
- If you want to use a different NIC, type the name of a NIC to use and press **Enter**.

4 Confirm whether you want to use the same public NIC on all nodes. Do one of the following:

- If all nodes use the same public NIC, enter **y**.
- If unique NICs are used, enter **n** and enter a NIC for each node.

5 Enter the virtual IP address for the Cluster Management Console.

```
Enter the Virtual IP address for Cluster Management Console:
[b,?] 10.10.12.1
```

6 Confirm the default netmask or enter another one:

```
Enter the netmask for IP 10.10.12.1: [b,?] (255.255.240.0)
```

7 Verify and confirm the Cluster Management Console information.

Configuring SMTP email notification

You can choose to configure SF Oracle RAC to send event notifications to SMTP e-mail services. You need to provide the SMTP server name and e-mail addresses of people to be notified. Note that it is also possible to configure notification after installation.

To configure SMTP email notification

1 Review the required information to configure the SMTP email notification.

2 Specify whether you want to configure the SMTP notification. Do one of the following:

- To configure SMTP notification, press Enter.

  ```
  Do you want to configure SMTP notification? [y,n,q] (y) y
  ```

- To skip configuring SMTP notification and advance to configuring SNMP notification, enter **n**.

  See "Configuring SNMP trap notification" on page 97.

3 Provide information to configure SMTP notification.

- Enter the SMTP server's host name.

  ```
  Enter the domain-based hostname of the SMTP server
  (example: smtp.yourcompany.com): [b,?]
  smtp.symantecexample.com
  ```

- Enter the email address of each recipient.

  ```
  Enter the full email address of the SMTP recipient
  (example: user@yourcompany.com): [b,?]
  admin@symantecexample.com
  ```

- Enter the minimum security level of messages to be sent to each recipient.

  ```
  Enter the minimum severity of events for which mail should be
  sent to ozzie@example.com  [I=Information, W=Warning,
  E=Error, S=SevereError]: [b,?] i
  ```

4 Add more SMTP recipients, if necessary.

- If you want to add another SMTP recipient, enter **y** and provide the required information at the prompt.

  ```
  Would you like to add another SMTP recipient? [y,n,q,b] (n) y

  Enter the full email address of the SMTP recipient
  (example: user@yourcompany.com): [b,?]
  harriet@symantecexample.com
  ```

```
Enter the minimum severity of events for which mail should be
sent to harriet@example.com  [I=Information, W=Warning,
E=Error, S=SevereError]: [b,?] E
```

- If you do not want to add, answer **n**.

```
Would you like to add another SMTP recipient? [y,n,q,b] (n)
```

5 Verify and confirm the SMTP notification information.

Configuring SNMP trap notification

You can choose to configure SF Oracle RAC to send event notifications to SNMP management consoles. You need to provide the SNMP management console name to be notified and message severity levels. Note that it is also possible to configure notification after installation.

To configure the SNMP trap notification

1 Review the required information to configure the SNMP notification feature of SF Oracle RAC.

2 Specify whether you want to configure the SNMP notification. Do one of the following:

- To configure SNMP notification, press Enter.

```
Do you want to configure SNMP notification? [y,n,q] (y)
```

- To skip configuring SNMP notification and advance to configuring global clustering option, enter **n**.

See "Configuring the global cluster option" on page 98.

3 Provide information to configure SNMP trap notification.

- Enter the SNMP trap daemon port.

```
Enter the SNMP trap daemon port: [b,?] (162)
```

- Enter the SNMP console system name.

```
Enter the SNMP console system name: [b,?] system2
```

- Enter the minimum security level of messages to be sent to each console.

```
Enter the minimum severity of events for which SNMP traps
should be sent to saturn [I=Information, W=Warning, E=Error,
S=SevereError]: [b,?] E
```

4 Add more SNMP consoles, if necessary.

- If you want to add another SNMP console, enter **y** and provide the required information at the prompt.

```
Would you like to add another SNMP console? [y,n,q,b] (n) y
Enter the SNMP console system name: [b,?] jupiter
Enter the minimum severity of events for which SNMP traps
should be sent to jupiter [I=Information, W=Warning,
```

```
E=Error, S=SevereError]: [b,?] i
```

- If you do not want to add, answer **n**.

```
Would you like to add another SNMP console? [y,n,q,b] (n)
```

5 Verify and confirm the SNMP notification information.

Configuring the global cluster option

You can configure the option for global clustering to link clusters at separate locations and enable wide-area failover and disaster recovery.

Note: You must have added a license for Verital Volume Replicator to configure the option for global clustering.

See Chapter 16, "Preparing for global clustering" on page 273

To configure the global cluster option

1 Review the required information to configure the global cluster option.

2 Specify whether you want to configure the global cluster option. Do one of the following:

- To configure global cluster option, press Enter.

```
Do you want to configure the Global Cluster Option? [y,n,q]
(y)
```

- To skip configuring global cluster option and advance to configuring the cluster volume manager, enter **n**.

  See "Setting permissions for database administration" on page 99

3 Provide information to configure the Global Cluster option.

The installer discovers and displays the same virtual IP address and netmask used by the Cluster Manager.

See "Configuring the Cluster Management Console" on page 95

Do one of the following:

- If you want to use the default values, press Enter.
- If you do not want to use the default value, enter another IP address.

  The installer prompts you for a NIC, value for the netmask, and value for the network hosts.

```
Enter the Virtual IP address for Global Cluster Manager:
[b,?] (11.136.88.199)
```

4 Verify and confirm the configuration of the global cluster.

Setting permissions for database administration

After SF Oracle RAC is installed, the default settings allow only the superuser to access the /opt/VRTSdbed folder. If you want database administrators (DBAs) to access SF Oracle RAC components, you must set the required permissions. You can skip setting the database administration permission and advance to configuring the cluster volume manager.

See "Configuring the cluster volume manager" on page 99.

To set permissions for database administration

1. Review the required information to set up the permissions for database administration.
2. Specify whether you want to add single user access, group access, or both on each of the nodes as the installer prompts.
   - Provide information if you want to add single user access.

     ```
     Do you want to add single user access on galaxy [y,n,q,?]
        (y)
     Enter login account name for DBA user: dba
     ```
   - Provide information if you want to add group access.

     ```
     Do you want to add group access on galaxy [y,n,q,?] (y)
     Enter group name for DBA users: oper
     ```

Configuring the cluster volume manager

Cluster volume manager configuration tasks include:

- Setting up naming scheme
- Setting up default disk group

Setting up naming scheme

Disks on AIX systems typically use device names such as /dev/rdsk/rhdisk*n* to identify disks on the system. It is possible to use the VxVM enclosure-based naming scheme, which allows disk arrays to be more readily recognizable. Dynamic Multipathing (DMP) is a prerequisite for enclosure-based naming schemes. Refer to the Veritas Volume Manager documentation for details on this scheme.

To set up the naming scheme

1. If you want to set up the enclosure-based naming scheme, enter **y**.

   ```
   Do you want to set up the enclosure-based naming scheme?
   [y,n,q,?] (n)
   ```
2. Specify if you want to use the new naming scheme for all eligible systems.

```
Do you want to use the enclosure-based naming scheme for all of
the eligible systems? [y,n,q,?] (y)
```

Setting up default disk group

If applicable, set up the default disk group. Because some VxVM commands require that a disk group be specified, the installer enables you to register the name of a default VxVM disk group on each eligible node. Note that you can create the default disk group later.

1 If you want to set up a default disk group, enter **y**.

```
Do you want to set up a system wide default disk group?
[y,n,q,?] (y) y

Which disk group? [<group>,list,q,?] xyz_dg
```

2 If you specified setting up a default disk group, review the setup output.

```
Volume Manager default disk group setup and daemon startup

Setting default diskgroup to xyz_dg on north ......... Done
Starting vxrelocd on nebula ........................ Started
Starting vxcached on nebula ........................ Started
Starting vxconfigbackupd on nebula ................. Started
.
.
```

Configuring VVR

If you added license for Veritas Volume Replicator during installation, you can now accept the default settings or modify the settings for VVR. The installer prompts you for the information on each node.

Note: You must have added a license for the Veritas Volume Replicator option.

Setting up VVR ports

The installer identifies the default ports that would be used VVR. You can also assign different ports at this point. Note that the port settings must be identical for systems that will be part of the same Replicated Data Set. They must also be identical for all the systems in a cluster.

To set up VVR ports

1 Review and accept the default port values that the configuration program displays.

```
Following are the default ports that will be used by VVR on
nebula:
```

```
.
.
Do you want to change any of the VVR ports on nebula? [y,n,q]
(n)
```

2 If you want to change any of the VVR ports on the system, enter **y**.

```
Do you want to change any of the VVR ports on nebula? [y,n,q]
(n) y
```

3 Follow the instructions to change the port values. Note the following points:

- The port settings must be identical for systems that will be part of the same Replicated Data Set.
- The port settings must also be identical for all the systems in a cluster.

Configuring VVR statistics collector

The VVR administrative daemon vradmind collects and maintains various statistics, which are helpful in solving VVR performance issues. You can tune the collection using a few tunables:

frequency	for gathering the statistics default = 10 seconds
number of days	for which the collected statistics should be preserved, after which the earlier statistics are automatically deleted default = 3 days

The installation program provides an option to change the default settings.

To configure VVR statistics collector

1 Enter **y** at the prompt to change the default setting.

2 Enter the values when the installer prompts.

Configuring VVR tunables

As an advanced user, you can modify the VVR tunable parameters. Refer to Veritas Volume Replicator documentation for more details.

To configure VVR tunables

1 Enter **y** to view or modify the VVR tunables.

2 Review the output to find whether the configuration is successful.

Starting the VAILAgent

You must start the VAILAgent to access array discovery service for deep mapping. After starting the agent, this service for deep mapping becomes accessible across the domain. Refer to the Veritas Volume Manager documentation for more information.

To start the VAILAgent

1. When the configuration prompts you, confirm the fully qualified host names of the cluster nodes.

```
Is the fully qualified hostname of system "galaxy" =
"galaxy.example.com"? [y,n,q] (y)
Is the fully qualified hostname of system "nebula" =
"nebula.example.com"? [y,n,q] (y)
```

2. Review the output as the program verifies communication with the remote nodes.

About Veritas Storage Foundation Management Server

Veritas Storage Foundation Management Server by Symantec (SF Management Server) ties together Storage Foundation product offerings to ensure that hosts in your data center use storage as efficiently as possible. You can use it to centrally monitor, visualize, and manage Storage Foundation hosts and generate reports about the hosts and the storage resources they consume.

Note: You are prompted to set up an optional SF Management Server managed host during SF Oracle RAC installation. After reviewing the description of SF Management Server, answer **n** to the prompt:
Enable Storage Foundation Management Server Management? [y,n,q] (y) **n**

SF Management Server is not available on the Storage Foundation and High Availability Solutions release. For information on ordering SF Management Server, visit:

www.symantec.com/enterprise/sfms

Refer to the Storage Foundation Management Server documentation for details on enabling centrally managed Storage Foundation hosts in an SF Oracle RAC environment.

Starting SF Oracle RAC processes

After configuring the cluster and optional features, start SF Oracle RAC to complete the installation.

To start SF Oracle RAC processes

1. Confirm that you desire to start the SF Oracle RAC processes when you see:

   ```
   Do you want to start Veritas Storage Foundation for Oracle RAC
   processes now? [y,n,q] (y) y
   ```

2. The installer configures CFS agents for SF Oracle RAC.

3. At the end of the product installation, the utility creates informational files and indicates where they are stored:

   ```
   Configuration log files, summary file, and response file are
   saved at:
           /opt/VRTS/install/logs/installsfrac-DzQaFO
   ```

   - A log file containing executed system commands and output.
   - A response file used with the `-responsefile` option of the installer.
   - A summary file containing the output of the installation scripts.

Performing post-installation tasks

Perform these tasks after installing SF Oracle RAC:

- ✔ Verifying GAB port membership.
- ✔ Setting up I/O fencing.
- ✔ Verifying the fencing GAB port membership.
- ✔ Reimporting new disk groups.
- ✔ Verifying the CVM group is online.

Verifying GAB port membership

Use GAB port membership as a method of determining if a specific component of the SF Oracle RACSF Oracle RAC stack is operating properly and communicating with its peers. The output below shows the common ports in use in a functional SF Oracle RAC environment before fencing is configured. Each line lists a GAB port, such as port `a`, a generation number determining a startup time, such as `gen 4a1c0001`, and a membership showing which LLT node IDs are participating, such as `membership 01`. In the first line of the output below, each node (0 and 1) has membership with the GAB utility that uses port `a`.

- To view GAB port membership, type:

  ```
  # /sbin/gabconfig -a
  ```

  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
```

The software configures the ports in the list for these functions:

Port	Function
a	GAB
b	I/O fencing
d	ODM (Oracle Disk Manager)
f	CFS (Cluster File System)
h	VCS (Veritas Cluster Server: High Availability Daemon)
o	VCSMM driver
v	CVM (Cluster Volume Manager)
w	`vxconfigd` (module for CVM)

Setting up I/O fencing

The shared storage for SF Oracle RAC must support SCSI-3 persistent reservations to enable I/O fencing. To review general guidelines on the process of checking disks in the SF Oracle RAC configuration menu, see "Viewing guidelines for checking SCSI-3 support" on page 89.

SF Oracle RAC involves two types of shared storage: data disks to store shared data, and coordinator disks, which are small LUNs (typically three per cluster), to control access to data disks by the nodes. Both data disks and the disks used as coordinator disks must be SCSI-3 compliant.

Setting up I/O fencing involves:

1. Adding data disks and coordinator disks, verifying the systems see the same disks
2. Testing data disks and coordinator disks for SCSI-3 compliance
3. Configuring coordinator disks
4. Enabling I/O fencing in the VCS configuration.

- If you are installing SF Oracle RAC and want to check the disks for SCSI-3 compliance before you configure the SF Oracle RAC components, use the procedures:
  - "Verifying the nodes see the same disk" on page 105
  - "Testing the disks using the vxfentsthdw script" on page 107
- If you have already tested that some or all the disks you have added are SCSI-3 compliant and have configured SF Oracle RAC components, go to the procedure "Configuring coordinator disks" on page 108.

Verifying the nodes see the same disk

A disk or LUN that supports SCSI-3 persistent reservations requires that two nodes have simultaneous access to the same disks.

To verify node access to the same disk

1. Use the following command to list the disks:

   ```
   lsdev -Cc disk
   ```

2. Use the `vxdisk scandisks` command to scan all disk drives and their attributes, update the VxVM device list, and reconfigure DMP with the new devices. For example, type:

   ```
   # vxdisk scandisks
   ```

   See the Veritas Volume Manager documentation for details on adding and configuring disks.

3 Use the `vxdmpadm getdmpnode` command to determine the VxVM name by which a disk drive (or LUN) is known.

   a In the following example, a disk with the AIX block device name `/dev/hdisk75` is identified by VxVM as `EMC0_17`:

```
# vxdmpadm getdmpnode nodename=hdisk75
NAME     STATE    ENCLR-TYPE  PATHS  ENBL  DSBL  ENCLR-NAME
==========================================================
EMC0_17  ENABLED  EMC         1      1     0     EMC0
```

   b You can, as an option, run the command `vxdisk list` *`vxvm_device_name`* to see additional information about the disk, including the AIX device name. For example:

```
# vxdisk list EMC0_17
```

4 Initialize the disks as VxVM disks using one of these methods:

   - Use the interactive `vxdiskadm` utility. When prompted, Symantec recommends specifying that the disk support Cross-platform Data Sharing (CDS) format.
   - Use the `vxdisksetup` command. This example specifies the CDS format:

```
vxdisksetup -i VxVM_device_name format=cdsdisk
```

     For example:

```
# vxdisksetup -i EMC0_17 format=cdsdisk
```

5 To confirm whether a disk or LUN supports SCSI-3 persistent reservations, two nodes must have simultaneous 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 the same serial number for the LUN is generated on all paths to the LUN.

   For example, an EMC array is accessible by the `/dev/rhdisk75` path on node A and the `/dev/rhdisk76` path on node B.

   From node A, type:

```
# vxfenadm -i /dev/rhdisk75
Vendor id       : EMC
Product id      : SYMMETRIX
Revision        : 5567
Serial Number   : 42031000a
```

   Expect the same serial number details to appear when you enter the equivalent command on node B using the `/dev/rhdisk76` path.

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

```
# vxfenadm -i /dev/rhdisk77
Vendor id       : HITACHI
Product id      : OPEN-3
Revision        : 0117
```

```
Serial Number   : 0401EB6F0002
```

Refer to the vxfenadm(1M) manual page for more information.

Testing the disks using the vxfentsthdw script

Before using the vxfentsthdw utility to test the shared storage arrays support SCSI-3 persistent reservations and I/O fencing, make sure to test disks serving as coordinator disks (see "Configuring coordinator disks" on page 108). Keep in mind that the tests overwrite and destroy data on the disks unless you use the -r option. Review these guidelines on testing support for SCSI-3:

- Verify the connection of the shared storage to two of the nodes on which you installed SF Oracle RAC.
- To ensure both nodes are connected to the same disk during the test, use the vxfenadm -i *diskpath* command to verify the disk serial number.
  See "Verifying the nodes see the same disk" on page 105.
- The two nodes must have ssh (default) or rsh communication. If you use rsh, launch the vxfentsthdw utility with the -n option.
  See "Setting up inter-system communication" on page 71.
- The vxfentsthdw utility has additional options suitable for testing many disks. You can test disks without destroying data using the -r option. The options for testing disk groups (-g) and disks listed in a file (-f) are described in detail:
  See "vxfentsthdw options and methods" on page 470.

To run the vxfentsthdw utility

1. Make sure system-to-system communication is functioning properly before performing this step.
   See "Setting up inter-system communication" on page 71.
1. From one node, start the utility.
   - If you use ssh for communication:
     ```
     # /opt/VRTSvcs/vxfen/bin/vxfentsthdw
     ```
   - If you use rsh for communication:
     ```
     # /opt/VRTSvcs/vxfen/bin/vxfentsthdw -n
     ```
2. After reviewing the overview and warning about overwriting 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
   ```

3 Enter the name of the disk 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/rhdiskx
/dev/rhdisk75
Enter the disk name to be checked for SCSI-3 PGR on node nebula
in the format: /dev/rhdiskx
Make sure it's the same disk as seen by nodes galaxy and nebula
/dev/rhdisk75
```

Regardless if the disk names are identical, the names must refer to the same physical disk to facilitate the testing.

4 After performing the check, make sure the `vxfentsthdw` utility reports the disk is ready for I/O fencing on each node.

5 Run the `vxfentsthdw` utility for each disk you intend to verify.

Note: If you have checked disks before configuring SF Oracle RAC components, return to "Configuring SF Oracle RAC Components" on page 90 to continue.

If disks cannot be successfully verified

If the `vxfentsthdw` utility cannot successfully verify that the storage devices can support SCSI-3 PR, you may need to remove keys that are written to the disk during the testing. For troubleshooting:

See "Removing existing keys from disks" on page 413.

Note: SF Oracle RAC I/O fencing and EMC together do not support the use of gate keeper devices as coordinator disks. Such administrative devices are intended for EMC use only.

Configuring coordinator disks

I/O fencing requires coordinator disks that are configured in a disk group and accessible to each node. These disks enables the `vxfen` driver to resolve potential split-brain conditions and prevent data corruption. For a description of I/O fencing and the role of coordinator disks:

See "I/O fencing" on page 35

Because coordinator disks are not used to store data, configure them as the smallest possible LUN on a disk array to avoid wasting space. Symantec recommends using hardware-based mirroring for coordinator disks.

Review these requirements and make sure you already added and initialized disks for use as coordinator disks:

- Use an odd number of coordinator disks with a minimum of three coordinator disks. This requirement ensures a majority of disks can be achieved.
- Each of the coordinator disks uses a physically separate disk or LUN.
- Use, if possible, coordinator disks that exist on different disk arrays.
- Initialize each disk as a VxVM disk. Symantec recommends the default CDS format.
  See "Initializing disks as VxVM disks" on page 469.
- Test to verify that the coordinator disks support SCSI-3 persistent reservations.
  See "Testing the coordinator disk group with vxfentsthdw -c" on page 110.
- Configure the coordinator disks in a disk group (for example, `vxfencoorddg`). Set the coordinator attribute when creating the disk group to prevent the disks in the group from being used for other purposes.
  See "Creating the coordinator disk group (vxfencoorddg)" on page 109.

Configuring coordinator disks involves three phases:

- Creating `vxfencoorddg`, the coordinator disk group
- Testing the coordinator disk group with the `vxfentsthdw -c` utility
- Creating the `vxfendg` file

Coordinator attribute

SF Oracle RAC uses a "coordinator" attribute for disk groups. The `vxfen` driver uses this attribute to prevent the reassignment of coordinator disks to other disk groups. The procedure that follows includes the setting of this attribute.

Refer to the Veritas Volume Manger documentation for more information on the coordinator attribute.

Creating the coordinator disk group (vxfencoorddg)

From one node, create a disk group named `vxfencoorddg`. This group must contain an odd number of disks or LUNs and a minimum of three disks.

For example, assume the disks have the device names `EMC0_12`, `EMC0_16, and EMCO_17`.

To create the coordinator disk group

1. On one node, create the disk group by specifying the device name of one of the disks; the option coordinator=on sets the coordinator attribute:

   ```
   # vxdg -o coordinator=on init vxfencoorddg EMC0_12
   ```

2. Add the other two disks to the disk group:

```
# vxdg -g vxfencoorddg adddisk EMC0_16
# vxdg -g vxfencoorddg adddisk EMC0_17
```

Refer to the Veritas Volume Manager documentation for details on creating disk groups.

Testing the coordinator disk group with vxfentsthdw -c

Review these requirements before testing the coordinator disk group (vxfencoorddg) with the vxfentsthdw utility:

- The vxfencoorddg disk group is accessible from two nodes.
- The two nodes must have rsh permission set such that each node has root user access to the other. Temporarily modify the /.rhosts file to enable cluster communications for the vxfentsthdw utility, placing a "+" character in the first line of the file. You can also limit the remote access to specific systems. Refer to the manual page for the /.rhosts file for more
- To ensure both nodes are connected to the same disks during the testing process, use the vxfenadm -i *diskpath* command to verify the serial number. See "Verifying the nodes see the same disk" on page 105.

In the procedure, the vxfentsthdw utility tests the three disks one disk at a time from each node. From the galaxy node, the disks are:

EMC0_12, EMC0_12, and EMC0_12

From the nebula node, the same disks are seen as:

EMC0_12, EMC0_12, and EMC0_12

To test the coordinator disk group

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

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

2. Enter the nodes you are using to test the coordinator disks.
3. Review the output to ensure the tests are successful. After testing all disks in the disk group, the vxfencoorddg disk group is ready for use.

   If a disk in the coordinator disk group fails verification, complete these operations:

   - Use the vxdiskadm utility to remove the failed disk or LUN from the vxfencoorddgdisk group. Refer to the Veritas Volume Manager documentation.
   - Add a new disk to the node, initialize it, and add it to the coordinator disk group. See "Creating the coordinator disk group (vxfencoorddg)" on page 109.
   - Test the disk group again. See "Testing the coordinator disk group with vxfentsthdw -c" on page 110.

If you need to replace a disk in an active coordinator disk group, refer to the topic in the troubleshooting section.

See "Adding or removing coordinator disks" on page 416.

Creating the vxfendg file

After setting up and testing the coordinator disk group, configure it for use.

To create the vxfendg file

1. Deport the disk group:

   ```
   # vxdg deport vxfencoorddg
   ```

2. Import the disk group with the `-t` option to avoid automatically importing it when the nodes restart:

   ```
   # vxdg -t import vxfencoorddg
   ```

3. Deport the disk group. This operation prevents the coordinator disks from serving other purposes:

   ```
   # vxdg deport vxfencoorddg
   ```

4. On all nodes, type:

   ```
   # echo "vxfencoorddg" > /etc/vxfendg
   ```

   Do no use spaces between the quotes in the "vxfencoorddg" text.

   This command creates the `/etc/vxfendg` file, which includes the name of the coordinator disk group. Based on the contents of the `/etc/vxfendg` file, the `rc` script creates the `/etc/vxfentab` file for use by the `vxfen` driver when the system starts. `/etc/vxfentab` invokes the `vxfenconfig` command, which configures the `vxfen` driver to start and use the coordinator disks listed in `/etc/vxfentab`. `/etc/vxfentab` is a generated file; do not modify this file.

Reviewing a sample /etc/vxfentab file

On each node, the list of coordinator disks is in the `/etc/vxfentab` file. The same disks may appear using different names on each node.

- On one node, for raw disks, an example `/etc/vxfentab` file resembles:

  ```
  /dev/rhdisk75
  /dev/rhdisk76
  /dev/rhdisk77
  ```

- For DMP disks, the file `/etc/vxfentab` file resembles:

  ```
  /dev/vx/rdmp/rhdisk75
  /dev/vx/rdmp/rhdisk76
  /dev/vx/rdmp/rhdisk77
  ```

If you must remove or add disks in an existing coordinator disk group, see the procedure in the troubleshooting chapter.

See "Adding or removing coordinator disks" on page 416.

Enabling fencing in the VCS configuration

Enabling fencing involves editing the UseFence attribute in the VCS configuration file (`main.cf`), verifying the configuration file syntax, copying the `main.cf` to other nodes, setting the contents of the vxfenmode file (DMP or raw), and restarting the fencing driver and VCS.

To enable I/O fencing

1. Save the existing VCS configuration file, `/etc/VRTSvcs/conf/config/main.cf`:

   ```
   # haconf -dump -makero
   ```

2. Stop VCS on all nodes with the command:

   ```
   # hastop -all
   ```

3. On *each* node, enter the following command:

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

4. Make a backup copy of the `main.cf` file:

   ```
   # cd /etc/VRTSvcs/conf/config
   # cp main.cf main.orig
   ```

5. On one node, use `vi` or another text editor to edit the `main.cf` file. Modify the list of cluster attributes by adding the `UseFence` attribute and assigning its value of `SCSI3`:

   ```
   cluster rac_cluster1 (
       UserNames = { admin = "cDRpdxPmHpzS." }
       Administrators = { admin }
       HacliUserLevel = COMMANDROOT
       CounterInterval = 5
       UseFence = SCSI3
       )
   ```

6. Save and close the file.

7. Verify the syntax of the `/etc/VRTSvcs/conf/config/main.cf` file:

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

8. Using `rcp` or another utility, copy the VCS configuration file from a node (for example, `galaxy`) to the remaining cluster nodes. On each remaining node, type:

   ```
   # rcp galaxy:/etc/VRTSvcs/conf/config/main.cf
     /etc/VRTSvcs/conf/config
   ```

9. Depending on whether you want to use the DMP configuration or the raw device configuration, use one of the following commands:

   - For DMP configuration (preferred):

     ```
     # cp /etc/vxfen.d/vxfenmode_scsi3_dmp /etc/vxfenmode
     ```

   - For raw device configuration:

     ```
     # cp /etc/vxfen.d/vxfenmode_scsi3_raw /etc/vxfenmode
     ```

Enter the command on all cluster systems.

10 On *each* node enter the sequence of commands that resembles the following example in which the DMP device is configured:

```
# echo vxfencoorddg > /etc/vxfendg
# /etc/init.d/vxfen start
# /opt/VRTS/bin/hastart
```

Verifying the fencing GAB port

After configuring fencing and starting VCS, CVM, and CFS on each node, use the gabconfig command to verify that all nodes appear in GAB port b membership.

◆ To verify GAB port membership for fencing, type:

```
# /sbin/gabconfig -a
```

Review the output for port b:

```
GAB Port Memberships
===============================================================
Port a gen 4a1c0001 membership 01
Port b gen g8ty0002 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
```

Verifying the CVM group is online

On all nodes, type:

```
# hagrp -state cvm
```

to verify that the cvm group is ONLINE.

Verifying I/O fencing configuration

Verify from the vxfenadm output that the SCSI-3 disk policy reflects the configuration in the /etc/vxfenmode file.

To verify I/O fencing configuration

◆ On one of the nodes, type:

```
# vxfenadm -d

I/O Fencing Cluster Information:
================================

Fencing Protocol Version: 201
Fencing Mode: SCSI3
Fencing SCSI3 Disk Policy: raw
```

```
Cluster Members:

    * 0 (north)
    1 (south)

RFSM State Information:
    node   0 in state  8 (running)
    node   1 in state  8 (running)
```

Chapter 4

Upgrading SF Oracle RAC

This chapter describes how to upgrade cluster systems to Storage Foundation 5.0 for Oracle RAC.

Upgrading to SF 5.0 Oracle RAC

The following upgrade scenarios are supported:

- Upgrading from SF Oracle RAC 4.0 MP3 to SF 5.0 Oracle RAC
- License-based upgrade to SF Oracle 5.0 RAC

Upgrading from 4.0 MP3 to SF 5.0 Oracle RAC

The procedure in the following sections assumes that you have a working SF Oracle RAC cluster at 4.0MP3.

Upgrading consists of:

- "Upgrading AIX to 5300-04-04."
- "Pre-Upgrade tasks: stopping Oracle and cluster resources."
- "Invoking installsfrac to upgrade to SF Oracle 5.0."

  Note: When invoking the installer, use the -rsh option if you have not configured SSH communications between systems. SSH is the default.

- "Post-upgrade tasks for SF 5.0 Oracle RAC."

Note: The upgrade procedure necessitates a reboot after installing the SF 5.0 Oracle RAC filesets.

Upgrading AIX to 5300-04-04

SF Oracle RAC for 5.0 requires that the AIX operating system be at level 5300-04-04. If you are using an earlier version of AIX, upgrade now.

Pre-Upgrade tasks: stopping Oracle and cluster resources

1. Log in as root user to one of the nodes in the cluster and make a backup of the VCS configuration file, main.cf.

   ```
   # cp /etc/VRTSvcs/conf/config/main.cf \
   /etc/VRTSvcs/conf/config/main.cf.40mp3
   ```

2. For Oracle9i:

   a. Stop gsd. On each node, log in as oracle user and enter:

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

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

   ```
   # hagrp -offline <group_name> -any
   ```

3. For Oracle 10g:

   a. Stop all resources configured under CRS control. As oracle user, enter:

   ```
   $ srvctl stop nodeapps -n <sys>
   ```

   Repeat the above command for each node in the cluster. Note that if the Oracle listener or database instances are managed by CRS, they are taken offline by the “srvctl” command shown above.

   b. If the Oracle listener or database instances are under VCS control, you must take the corresponding VCS service groups offline. As root user, enter:

   ```
   # hagrp -offline <group_name> -any
   ```

   c. Stop CRS.

   - If CRS is not controlled by VCS, log in as root user on each system in the cluster and enter:

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

   - If CRS is controlled by VCS, log in as root user on any system in the cluster and enter:

     ```
     # hares -offline <cssd_res> -sys <sys>
     ```

     Repeat this command for each system in the cluster.

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

5. Stop all applications that use VxFS or VxVM but that are not under VCS control. Unmount the corresponding file systems and deport the corresponding disk groups.

6 Make sure VCS is running on all systems in the cluster.

7 Set the VCS resource attribute, AutoStart, to 0 for all VCS resources that manage CRS or Oracle database instances. From any system in the cluster, enter:

```
# haconf -makerw
# hares -modify <cssd_res> AutoStart 0
# hares -modify <oracle_res> AutoStart 0
# haconf -dump -makero
```

8 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:

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

9 On each system, invoke "slibclean" to unload all unused modules:

```
# /usr/sbin/slibclean
```

Invoking installsfrac to upgrade to SF Oracle 5.0

1 Mount the software disc:

```
# mkdir /cdrom
# mount -V cdrfs -o ro /dev/cd0 /cdrom
```

2 Start the SF Oracle RAC installer utility. Include the -rsh option if you have not set up SSH:

```
# cd /cdrom/dvd1/sfrac
# ./installsfrac
```

3 The installer displays information about the cluster, the systems, and the service groups, and prompts you about upgrading:

```
...
Do you want to upgrade to version 5.0 on these systems using the
current configuration? [y,n,q,?] (y)
```

Press Enter.

4 The installer verifies rsh accessibility and the existing licenses.

```
...
Do you want to enter another license key for galaxy? [y,n,q](n)
```

5 If necessary, answer "y" and enter the license keys at the prompt.

6 The installer checks existing AIX operating system filesets and reports any missing recommended filesets, prompting you to add them before continuing.

7 Install any required or recommended filesets for SF 5.0 Oracle RAC and answer "y" at the prompt to continue.

8 The installer displays the list of all filesets included with SF 5.0 Oracle RAC and proceeds to stop SF Oracle RAC processes on each system in the cluster. The installer prompts you to confirm whether to proceed with the upgrade.

```
All SFRAC processes that are currently running must be stopped.

Are you sure you want to upgade SFRAC [y,n,q](y)
```

Answer "y" to continue with the upgrade.

9 As the installer proceeds, it:

a Takes the cvm group and other dependent groups offline.

b Updates GAB and VCS configurations as needed for SF 5.0 Oracle RAC.

10 When the installer is ready to stop the cluster to make updates to the configuration, it prompts for your OK. At the prompt, press Enter.

```
...
Are you ready to begin the Veritas Storage Foundation for Oracle
RAC upgrade at this time? [y,n,q](y)
```

11 The installer backs up the configuration files, freezes VCS service groups, and uninstalls the 4.0 SF Oracle RAC filesets.

```
    Backing up types.cf and main.cf .................Done
    Freezing group cvm ..............................Done
    ...
    Stopping SFRAC: 100%
Shutdown completed successfully on all systems
    Uninstalling SFRAC: 100%
Uninstall completed successfully on all systems
    Installing SFRAC: 100%
Installation completed successfully on all systems
```

12 The installer displays where logs for the upgrade can be found. It then indicates you must reboot each system.

13 Reboot each system in the cluster.

```
# /usr/sbin/shutdown -r
```

Post-upgrade tasks for SF 5.0 Oracle RAC

When the systems come back up after the rebooting, GAB ports, a, b, d, h, and o are configured. The VCS service groups are frozen.

1 Invoke installsfrac once again, this time to enter configuration information. Include the -rsh option if you have not set up SSH:

```
# cd /opt/VRTS/install
# ./installsfrac -configure
```

2 Enter the system names when prompted:

```
Enter the system names separated by spaces on which to configure
SFRAC: galaxy nebula
```

3 The installer now verifies rsh accessibility between the systems and prompts you whether to use the same VCS/GAB/LLT configuration. Answer "y."

```
Are you sure you want to reuse configuration of SFRAC 4.0.3.0?
[y,n,q] (y)
```

4 The installer now stops all SF Oracle RAC processes.

```
Stopping SFRAC processes. Please wait...
SFRAC processes are stopped

Press [Return] to continue:
```

5 The installer next prompts you to change permissions to allow the current oracle user and group to access Veritas Storage Foundation for Oracle RAC tools. The questions are asked for each system in the cluster.

```
...
The Veritas Storage Foundation for Oracle RAC utility for
database administration requires permission changes to allow
database administrators (DBAs) access to the tools. The default
settings at installation time for the /opt/VRTSdbed directory
allows only the root user access to the
directory. You need to change permissions to allow a DBA or a
group of DBAs access to the Veritas Storage Foundation for
Oracle RAC tools

Do you want to add single user access on galaxy [y,n,q,?] (Y)
```

Answer "y"

```
Enter login account name for DBA user: oracle
Enter group name for DBA users: dba
```

6 For each system, the installer prompts you to verify the fully qualified hostnames. Answer "y" to confirm. For example:

```
...
Is the fully qualified hostname of system "galaxy" =
"galaxy.example.com"? [y,n,q,b,?] (y) y

Querying fully qualified domain name of host "galaxy" .. ok

Is the fully qualified hostname of system "nebula" =
"nebula.example.com"? [y,n,q,b] (y) y
```

7 When you are prompted about enabling Storage Foundation Management Sever, answer "n".

```
...
Enable Storage Foundation Management Server Management?
[y,n,q,b] (y) n
```

8 The installer now configures VEA, gridnode, actionagent, and Storageagent for all nodes in the cluster.

```
Configuring VEA in STANDALONE mode on "galaxy" ..... ok
Configuring gridnode on "galaxy" ................... ok
```

```
    Registering gridnode on "galaxy" ................... ok
    Configuring actionagent on "galaxy" ................ ok
    Registering actionagent on "galaxy" ................ ok
    Stopping StorageAgent on "galaxy" .................. ok
    Registering StorageAgent on "galaxy" ............... ok
....
```

9 When you are prompted to start SF Oracle RAC processes, accept the default "y" by pressing Enter.

```
Do you want to start Veritas Storage Foundation for Oracle RAC
processes now? [y,n,q,b] (y)
```

10 After starting the SF Oracle RAC processes, the installer exits.

```
    Setting default disk group to  on galaxy ........... Done
    Unfreezing cvm ..................................... Done
    Unfreezing oradb1_grp .............................. Done
    Onlining ClusterService Group on galaxy ............ Done
    Onlining cvm Group on galaxy ....................... Done
    Onlining cvm Group on nebula ....................... Done
    Setting default disk group to  on nebula ........... Done
```

11 Check the output of the gabconfig -a command to ensure that ports a, b, d, f, h, o, v, and w are configured.

12 All VCS resources, except the ones for which AutoStart is set to 0, should now be online.

See "Pre-Upgrade tasks: stopping Oracle and cluster resources" on page 116 for setting the AutoStart attribute for resources to 0.

13 Import any VxVM disk groups not under VCS control. Also mount any VxFS file systems not under VCS control.

14 If your systems run Oracle 10g, ensure that the file system containing CRS binaries is online on all systems in the cluster. Also, the OCR and Vote disks must be accessible from all nodes in the cluster.

15 Verify the file system containing Oracle binaries is mounted and accessible on all systems in the cluster. Also ensure that the file system containing the Oracle databases is mounted.

16 Ensure that CRS (for Oracle 10g) and the Oracle database instances are not running.

17 If your systems run Oracle9i, skip to step 20. If your systems run Oracle 10g, perform the following setup steps on each system in the cluster before bringing up CRS:

a Unload any unused modules:

```
# /usr/sbin/slibclean
```

b Create the 32-bit SKGXN archive library under /opt/VRTSvcs/rac/lib:

```
# cd /opt/VRTSvcs/rac/lib
```

```
# rm libskgxn2.a
# cp libvcsmm.so shr_skgxn2.o
# ar -X32 -qv libskgxn2.a  shr_skgxn2.o
# rm shr_skgnx2.o
```

c Create the 64-bit SKGXN archive library under /opt/VRTSvcs/rac/lib64

```
# cd /opt/VRTSvcs/rac/lib64
# rm libskgxn2.a
# cp libvcsmm.so shr_skgxn2.o
# ar -X64 -qv libskgxn2.a shr_skgxn2.o
# rm shr_skgxn2.o
```

18 If you are running 10gR1, copy these 5.0 SKGXN libraries to $CRS_HOME

```
# cp /opt/VRTSvcs/rac/lib/libskgxn2.a \
$CRS_HOME/lib32/libskgxn2.a
# cp /opt/VRTSvcs/rac/lib/libskgxn2.a \
$CRS_HOME/lib32/libskgxnr.a
# cp /opt/VRTSvcs/rac/lib64/libskgxn2.a \
$CRS_HOME/lib/libskgxn2.a
# cp /opt/VRTSvcs/rac/lib64/libskgxn2.a \
$CRS_HOME/lib/libskgxnr.a
```

Run the commands on *each* system in the cluster.

19 If you're running 10gR2, perform the following commands on each system in the cluster.

a Remove the following files from under /opt/ORCLcluster:

```
# rm /opt/ORCLcluster/lib/libskgxn2.a
# rm /opt/ORCLcluster/lib/libskgxn2.so
# rm /opt/ORCLcluster/lib32/libskgxn2.a
# rm /opt/ORCLcluster/lib32/libskgxn2.so
```

b Copy these SF 5.0 Oracle RAC SKGXN libraries to /opt/ORCLcluster.

```
# cp /opt/VRTSvcs/rac/lib/libskgxn2.a \
/opt/ORCLcluster/lib32/libskgxn2.a
# cp /opt/VRTSvcs/rac/lib/libvcsmm.so \
/opt/ORCLcluster/lib32/libskgxn2.so
# cp /opt/VRTSvcs/rac/lib64/libskgxn2.a \
opt/ORCLcluster/lib/libskgxn2.a
# cp /opt/VRTSvcs/rac/lib64/libvcsmm.so \
/opt/ORCLcluster/lib/libskgxn2.so
```

Run the commands on *each* system in the cluster.

c Copy the SF Oracle RAC SKGXP library to /opt/ORCLcluster:

```
# cd /opt/ORCLcluster/lib32
# cp /opt/VRTSvcs/rac/lib/libskgxp10_ver25_32.so .
# cp /opt/VRTSvcs/rac/lib/libskgxp10_ver25_32.a .

# cd /opt/ORCLcluster/lib
# cp /opt/VRTSvcs/rac/lib/libskgxp10_ver25_64.so .
# cp /opt/VRTSvcs/rac/lib/libskgxp10_ver25_64.a .
```

d Copy the clsinfo script to /opt/ORCLcluster.

```
# cd /opt/ORCLcluster
# mkdir bin
# cd bin
# cp /opt/VRTSvcs/rac/bin/scripts/clsinfo .
```

20 Next, relink Oracle. Invoke installsfrac, enter the system names, navigate to the “Install and Relink Oracle” menu, and select the appropriate Oracle version:

```
    1) Oracle 9iR2
    2) Oracle 10gR1
    3) Oracle 10gR2\
Select an option [1-3,2] 3
```

21 On the subsequent screens, enter the DISPLAY, Oracle User, Oracle Group, location of ORACLE_BASE, location of CRS_HOME, location of ORACLE_HOME, and the Oracle Patch level currently installed on the systems:

```
Enter Oracle UNIX user name: [b] (oracle) oracle
Enter Oracle UNIX group name: [b] (oinstall) oinstall
Enter Oracle base directory: [b] /app/oracle
Enter absolute path of CRS home directory: [b] /app/crshome
Enter absolute path of Database Home directory: [b]
 /app/loracle/orahome
Enter Oracle Patch level for 10.2: [b] (0.1)
```

22 Confirm your responses in the verification screen. The Installer now copies the 5.0 SF Oracle RAC libraries to ORACLE_HOME and relinks Oracle.

```
Oracle environment information verification

    Oracle Unix User: oracle
    Oracle Unix Group: oinstall
    Oracle Clusterware (CRS) Home: /app/oracle/orahome
    Oracle Release: 10.2
    Oracle Patch Level: 0.1
    Oracle Base: /app/oracle
    Oracle Home: /app/oracle/orahome

Is this informatin correct: [y,n,q] (y)

Verifying binaries in /app/oracle/orahome on galaxy ...ok
Verifying binaries in /app/oracle/orahome on nebula ...ok
Copying SFRAC libskgxn on galaxy .......................ok
Copying SFRAC libskgxn on nebula ........................ok
Copying SFRAC ODM library on galaxy ....................ok
Copying SFRAC ODM library on nebula ....................ok
Copying SFRAC libskgxp on galaxy .......................ok
Copying SFRAC libskgxp on nebula ........................ok
Relinking Oracle on galaxy .............................ok
Relinking Oracle on nebula .............................ok

Oracle Relinking is now complete.
```

```
After relinking is complete, enter "q" to quit the installer.
```

23 As root user, removing the prefix ":" to uncomment the CRS-specific lines in the /etc/inittab file.

24 Start the CRS daemons.

   a If CRS is not configured under VCS control, log in as root user on each system in the cluster and enter:

   ```
   # /etc/init.crs start
   ```

   b If Oracle instances are configured under VCS control, log in as root user and enter the following command for each system in the cluster:

   ```
   # hares -online <ora_res> -sys <sys>
   ```

25 Verify that the Oracle instances are up and running.

26 Reset the VCS resource attribute AutoStart to 1 for all resources that manage Oracle CRS and database instances. As root user on any system in the cluster, enter:

```
# haconf -makerw
# hares -modify <cssd_res. AutoStart 1
# hares -modify <oracle_res> AutoStart 1
# haconf -dump -makero
```

Upgrading licenses to SF Oracle RAC level

You can upgrade from other Storage Foundation 5.0 products to SF 5.0 Oracle RAC. The following SF products can be upgraded to SF Oracle RAC:

- Storage Foundation for Oracle
- Storage Foundation for Oracle High Availability
- Storage Foundation Cluster File System

The following procedures outline how to perform these upgrades.

Upgrading to SF Oracle RAC from Storage Foundation for Oracle

If you have Storage Foundation for Oracle installed, use the following procedure to add the license for SF Oracle RAC and implement it.

1. Install the license for SF Oracle RAC.

   ```
   # installsfrac -license
   ```

2. Shutdown and restart all nodes.

   ```
   # shutdown -r
   ```

3. Use the installer to configure SF Oracle RAC. Refer to "Configuring SF Oracle RAC Components" on page 90.

Upgrading to SF Oracle RAC from Storage Foundation HA for Oracle

If you have Storage Foundation for Oracle HA installed, use the following procedure to add the license for SF Oracle RAC and implement it.

1. Install the license for SF Oracle RAC.

   ```
   # installsfrac -license
   ```

2. Configure the vcsmm driver.

   ```
   # echo  '/sbin/vcsmmconfig -c &' > /etc/vcsmmtab
   ```

   Enter this command on all systems in the cluster.

3. Shutdown and restart all nodes.

   ```
   # shutdown -r
   ```

4. Configure I/O fencing. Refer to "Setting up I/O fencing" on page 105.

5. Enter the command to configure CFS.

   ```
   # /opt/VRTSvxfs/cfs/bin/cfscluster config
   ```

Upgrading to SF Oracle RAC from Storage Foundation Cluster File System

1. Install the license for SF Oracle RAC.

   ```
   # installsfrac -license
   ```

2. Shutdown and restart all nodes.

   ```
   # shutdown -r
   ```

3. Configure SF Oracle.

   ```
   # chmod 705 /opt/VRTSdbed
   # chown oracle:dba /optVRTSdbed
   ```

4. Create the database repository. Refer to Chapter 18, "Configuring the repository database for Oracle" on page 315.

Upgrading CVM protocol and disk group version

To take advantage of the new features in this release, you must upgrade the Veritas Cluster Volume Manager (CVM) protocol version (70), and upgrade to the latest disk group version (140).

To verify the cluster protocol version

Verify the cluster protocol version, enter the following command:

```
# /opt/VRTS/bin/vxdctl protocolversion
```

If the cluster protocol version is less than 70, then it needs to be upgraded to 70 for SF 5.0 Oracle RAC.

To upgrade the cluster protocol version

From the CVM master node run:

```
# /opt/VRTS/bin/vxdctl upgrade
```

To upgrade the disk group version

Upgrade the disk group version to 140 by entering the following command on the master node:

```
# vxdg -T 140 upgrade <disk_group_name>
```

Section III

Setting up SF Oracle RAC with Oracle9*i*

After installing and configuring SF Oracle RAC, install Oracle9*i* and configure SF Oracle RAC.

Chapter 5

Preparing to Install Oracle9i RAC

After setting up SF Oracle RAC, you can prepare to install Oracle9i. Install the software locally on each node.Make sure to review the Oracle installation manuals before installing Oracle9i.

This chapter contains the topics:

About the location of ORACLE_HOME

Before installing Oracle binaries (ORACLE_HOME), consider these points:

- Local installations provide a comfort level using traditional installation methods and the possibility of improved protection against a single point of failure.
- CFS installations provide a single Oracle installation to manage, regardless of number of nodes. This scenario offers a necessary reduction in storage requirements and easy addition of nodes.

Select the location based on your high availability requirements. Symantec generally recommends using local installations.

Performing preinstallation operations

Performing preinstallation operations involves manual and automated tasks from the SF Oracle RAC configuration program.

The Oracle installation requires some preparation such as creating the Oracle user and group ID, and creating disk groups, volumes, and mount points.

- "Adding users and groups for Oracle."
- "Setting up oracle user equivalence for RSH and RCP."
- "Verifying RSH access for "oracle" user."
- "Setting shell limits for "oracle" user."
- "Configuring maximum user Processes."
- "Verifying the user "nobody" exists."
- "Creating file system or volume for SRVM."
- "Creating a disk group and file system for the Oracle software."
- "Preparing $ORACLE_BASE on each node."
- "Preparing $ORACLE_HOME on each node."

Adding users and groups for Oracle

Note: When creating user IDs and group IDs make sure they are the same on each system.

To perform preinstallation tasks in the configuration menu

1. Launch the SF Oracle RAC configuration program. As root user on any one system, enter:

   ```
   # cd /opt/VRTS/install
   # installsfrac -configure
   ```

   By default, the installsfrac program uses SSH for remote communication. However, to use RSH, specify the -rsh option with the installsfrac program.

   ```
   # ./installsfrac -rsh -configure
   ```

2. Enter the system names, separated by spaces, and navigate to the menu, **Prepare to Install Oracle**.

3. Select **Create userid and groupid for Oracle** from the configuration menu. Enter Oracle UNIX user and primary group information:

   ```
   Enter Oracle UNIX user name: [b] oracle
   ```

```
Enter Oracle UNIX user id (numberical): [b] 1001
Enter Oracle UNIX user home dir: [b] /lhome/oracle
Enter Oracle UNIX group name: [b] dba
Enter Oracle UNIX group id (numerical): [b] 101
```

4 The installer verifies that the specified userid does not exist on any of the systems in the cluster and then creates it.

```
User oracle does not exist on any node. Do you want to create
it with the information provided [y,n,q] (y) y
    Creating user oracle on galaxy ................. Done
    Creating user oracle on nebula ................. Done
```

5 Enter the information to create secondary group “oper”:

```
Do you want to create secondary groups for Oracle user?
[y,n,q] (y) y
Enter Oracle UNIX secondary group name: [b] oper
Enter Oracle UNIX secondary group id (numerical): [b] 102

Group oper does not exist on any node. Do you want to create
it with the information provided [y,n,q] (y) y

Creating group oper on galaxy ......................... Done
Adding Oracle user (oracle) to group (oper) on galaxy ..Done
Creating group oper on nebula ......................... Done
Adding Oracle user (oracle) to group (oper) on nebula.. Done
```

6 After creating the secondary groups, the installer proceeds to verify RSH access for “oracle” user. You must first enable RSH access for the newly created “oracle” user. Leave the installer at this prompt and proceed to setup RSH access. You can return to the installer session after setting up oracle user equivalence.

Setting up oracle user equivalence for RSH and RCP

1 As `root` user on each system, edit /etc/hosts.equiv file and add entries similar to the following:

```
galaxy oracle
nebula oracle
```

2 On each system, set the password for the “oracle” user:

```
[root@galaxy /]# passwd oracle
Changing password for "oracle"
oracle's New password:
Re-enter oracle's new password:
```

3 On each system, login as user “oracle” and change the passwd.

```
[root@galaxy /]# su - oracle
$ passwd
Changing password for "oracle"
oracle's New password:
```

```
Re-enter oracle's new password:
```

4 On each system, as user “oracle”, verify “rsh” access:

```
$ rsh galaxy date
Mon Apr 24 10:02:45 PDT 2006
$ rsh nebula date
Mon Apr 24 10:02:45 PDT 2006
```

Verifying RSH access for “oracle” user

1 Return to the installsfrac session from step 9 in section “Adding users and groups for Oracle” on page 130.

2 At the installer prompt, answer “y” to verify “RSH” accessibility.

```
Do you want to perform rsh verification? [y,n,q] (y)
    Checking rsh communication for oracle user on galaxy .
Checked
    Checking rsh communication for oracle user on nebula ....
Checked
```

3 Quit the installation program

Setting shell limits for “oracle” user

Use smitty to verify that the shell limits for user “oracle” are set per the following table:

Soft FILE size	-1
Soft CPU time	-1
Soft Data Segment	-1
Soft STACK size	-1

To set the shell limits

1 Log in to a system as root user and enter:

```
# smitty chuser
```

2 For the “USER Name” field, enter: oracle.

3 Scroll down the list of attributes and edit their values. See the table above.

4 Press F10 to save the changes.

5 Perform the steps above on each system.

Configuring maximum user Processes

The maximum number of processes per user must be set 2048 or greater. Use the chdev command to configure maxuproc on each system. For example:

```
# chdev -l sys0 -a maxuproc='2048'
sys0 changed
```

Verifying the user "nobody" exists

Verify the user "nobody" exists on each system in the cluster:

```
# id nobody
uid=4294967294(nobody) gid=4294967294(nobody)
```

Creating file system or volume for SRVM

The Oracle Universal Installer (OUI) requires you to specify a location for the SRVM configuration file during the Oracle installation. The Server Control tool (srvctl) and Oracle Enterprise Manager use the SRVM configuration file. Symantec recommends creating a location for the configuration file to enable you to use the tools supplied by Oracle. The SRVM configuration file is a standalone volume or file system.

To create a raw volume for SRVM (option 1)

1. On the master node, create a shared disk group on a shared disk using the VxVM name of the disk, such as `EMC0_1` or `IBM_SHARK_5`. This procedure assumes you are creating the `srvm_vol` volume of 300 MB in the `orasrvdg` shared disk group on the `EMC0_1` disk:

   ```
   # vxdg -s init orasrvdg EMC0_1
   ```

   For information about listing disks on the system and finding their VxVM disk names, refer to "Verifying the nodes see the same disk" on page 105.

2. Create the volume in the shared disk group:

   ```
   # vxassist -g orasrvdg make orasrvvol 300M
   ```

3. Set the ownership and permissions for the volume:

   ```
   # vxedit -g orasrvdg set user=oracle group=dba
      mode=660 orasrvvol
   ```

To create a cluster file system for SRVM (option 2)

1. On any node, create a VxFS file system for SRVM on the shared volume created for this purpose (see previous section). For example, create the file system on `orasrvvol`:

   ```
   # mkfs -V vxfs /dev/vx/rdsk/orasrvdg/orasrvvol
   ```

2. On each node, create the mount point for the file system:

   ```
   # mkdir /orasrv
   ```

3 On each node, mount the file system using the device file for the block device:

```
# mount -V vxfs -o cluster \
    /dev/vx/dsk/orasrvdg/orasrvvol /orasrv
```

4 On only one of the nodes, change the ownership again:

```
# chown -R oracle:dba /orasrv
```

Creating a disk group and file system for the Oracle software

1 Log in as root user on one node.

2 On one node, create a disk group. For example, type:

```
# vxdg init orabindg_galaxy EMC0_1
```

3 Create the volume of about 7 GB in the group:

```
# vxassist -g orabindg_galaxy make \ orabinvol_galaxy 7168M
```

4 Create a VxFS file system on `orabinvol_galaxy` to install the binaries. For example, type:

```
# mkfs -V vxfs /dev/vx/dsk/orabindg_galaxy/orabinvol_galaxy
```

5 Create the mount point for the file system:

```
# mkdir /app
```

6 Mount the file system using the device file for the block device:

```
# mount -V vxfs /dev/vx/dsk/orabindg_galaxy/orabinvol_galaxy
/app
```

7 Edit the `/etc/filesystems` file. Find the listing for the `/app` file system and verify that it resembles this entry:

```
.
/app:
        dev             =
/dev/vx/dsk/orabindg_galaxy/orabinvol_galaxy
        vfs             = vxfs
        mount           = true
        check           = true
        account         = false
.
```

8 Repeat step 1 through step 7 on each other node in the cluster.

Preparing $ORACLE_BASE on each node

1 On each system, log in as root.

2 Create the direcotry for ORACLE_BASE:

```
# mkdir -p /app/oracle
```

3 Change ownership and permissions:

```
# chown -R oracle:dba /app/oracle
```

```
# chmod -R 775 /app/oracle
```

Preparing $ORACLE_HOME on each node

1. On each system, log in as root.
2. Create the directory for ORACLE_HOME:

   ```
   # mkdir -p /app/oracle/orahome
   ```

3. Change ownership and permissions:

   ```
   # chown -R oracle:dba /app/oracle/orahome
   # chmod -R 775 /app/oracle/orahome
   ```

Setting umask before installation

Use the following command to set the default file creation mode on each system where you plan to install Oracle.

```
# umask 0022
```

Chapter 6

Installing Oracle9i RAC

After installing SF Oracle RAC and preparing to install Oracle software, proceed to install Oracle9i R2. Installing this software in an SF Oracle RAC environment involves the following tasks:

✔ "Installing Oracle9i Database Software" on page 137

✔ "Completing post-installation operations" on page 141

Installing Oracle9i Database Software

You can install Oracle9i R2 on a local hard disk on each node. Install the software in a file system or a raw volume. See "Creating a disk group and file system for the Oracle software" on page 134.

Installing Oracle9i Release 2 binary on each node locally

In the procedure described below, Oracle 9i Database Software is installed on each node in the location created in the "About the location of ORACLE_HOME" on page 129.

Installing Oracle9i Using SF Oracle RAC installer

1 Launch the installsfrac program. As root user on any one node in the cluster, enter:

```
# cd /opt/VRTS/install
# installsfrac -configure
```

By default, the installsfrac program uses ssh for remote communication. However, rsh can be used in instead of SSH by using the "-rsh" option with the installsfrac utility.

```
# cd /opt/VRTS/install
# installsfrac -rsh -configure
```

2 Enter the system names on which you want to install Oracle9i software and navigate to the **Install or relink Oracle** menu.

3 Select the version of Oracle software (**Oracle 9iR2**).

4 Select **Install Oracle RDBMS server** to install Oracle software.

5 Enter the value for the DISPLAY environment variable. This value should refer to the address of the X11 server where you want OUI for Oracle installation displayed. For example, use “10.180.94.84:0.0.”

6 In the Oracle user name screen, enter the Oracle UNIX user name when prompted. The installer checks for the user on all systems. For example, use “oracle.”

7 In the Oracle Group name screen, enter the name of the Oracle inventory group when prompted. The installer checks for the existence of this group on all systems. For example, use “dba.”

8 In the Database Install Image screen, specify the location of the Oracle Database Software Install Image. For example, `/oraimage/9201/Disk1`.

9 In the Oracle base directory screen, specify the base of the Oracle directory structure for OFA compliant databases. For example, use `/app/oracle`.

10 In the Database Home Directory screen, specify the location of $ORACLE_HOME. For example, `/app/oracle/orahome`.

11 In the Patch Level screen, specify the patch level of Oracle Database Software being installed. Accept the default when installing base version of Oracle 9i (9.2.0.1).

12 The installer prints Oracle9i installation information for verification. Answer “y” if the information displayed is correct, answer “n” otherwise. For example:

```
Oracle environment information verification
Oracle Unix user: oracle
Oracle Unix Group: dba
Oracle Release: 9.2
Oracle Patch: 0.1
Oracle Base: /app/oracle
Oracle Home: /app/oracle/orahome
RDBMS Installation Path: /oraimage/9201/Disk1
Is this information correct: [y,n,q] (y)
```

13 The installer now validates the inputs and performs certain setup steps, after which it prompts you to run rootpre.sh on all cluster nodes.

```
Verifying  fileset prerequisites on galaxy ........... ok
Verifying fileset prerequisites on nebula ............ ok
Comparing Oracle UNIX user id on all systems ........ Done
Comparing Oracle group id on all systems ........... Done
Creating SFRAC odm entries on galaxy ................. ok
```

```
    Creating SFRAC odm entries on nebula ................. ok

For installation to proceed, Oracle requires you to run the
rootpre.sh script located at:
/oraimage/9201/Disk1

Run the following commands on all nodes:
# cd /oraimage/9201/Disk1
# ./rootpre.sh

Has rootpre.sh been run by root? [y,n,q]
```

Note: Do not respond to the prompt yet, but proceed to step 14.

14 On each system, log in as root in another terminal session.

a On each system, change the directory to the location of the rootpre.sh script, and invoke the script.

b After invoking rootpre.sh on all nodes, return to the installsfrac prompt (from step 13) and enter “y” to indicate that the script has been invoked on all systems in the cluster.

15 Installer now brings up the OUI for Oracle RDBMS Software Installation.

```
Invoking OUI to install Oracle Software. Please wait .. ok

Now install Oracle database software from the OUI. Refer to
Oracle documentation for installation instructions.

NOTE: Near the end of the installation, Oracle Database
Software installer will prompt you to run the root.sh
script.

Press <RETURN> here when prompted to run root.sh
```

Note: Do not press Return yet, but wait for the prompt to do so.

16 Enter the data in Oracle's OUI.

a Specify $ORACLE_BASE location in the “Inventory Location” dialog.

b Specify Oracle Inventory group name in the “UNIX Group Name” dialog.

c Invoke the script “orainstRoot.sh” when prompted.

d In the “Cluster Node Selection” dialog, select all nodes for installation. If this dialog box does not appear, refer to “Missing dialog box during installation of Oracle9i” on page 409.

e On the “File Locations” dialog, change the destination name from OUIHome to ORACLE_HOME and specify the path of $ORACLE_HOME.

f On the “Database Configuration” dialog, select “Software Only”.

g On the dialog titled “Shared Configuration File Name,” specify the location of the SRVM file.

h On the final screen, verify the inputs and press “Install” to start the Oracle database software installation.

17 When the Oracle database binaries have been installed, the OUI prompts you to invoke the root.sh script.

Note: Do not run root.sh yet, but return to the installsfrac session from step 13.

---

18 In the installsfrac session, press Return. The installsfrac utility now verifies the database software installation, copies the SF Oracle RAC libraries to $ORACLE_HOME, and relinks Oracle on each node in the cluster.

```
Verifying binaries in /app/oracle/orahome on galaxy ...ok
Verifying binaries in /app/oracle/orahome on nebula ...ok
Copying SFRAC libskgxn on galaxy .......................ok
Copying SFRAC libskgxn on nebula .......................ok
Copying SFRAC ODM library on galaxy ....................ok
Copying SFRAC ODM library on nebula ....................ok
Copying SFRAC libskgxp on galaxy .......................ok
Copying SFRAC libskgxp on nebula .......................ok
Relinking Oracle on galaxy .............................ok
Relinking Oracle on nebula .............................ok
Oracle RDBMS Software Installation is now complete. Enter "q" to
quit the SF Oracle RAC installer, installsfrac.
```

19 As root user on each system, invoke the root.sh script:

```
# cd $ORACLE_HOME
# ./root.sh
```

20 Return to the OUI and indicate you have run root.sh.

21 Exit the OUI.

22 After completing installation of Oracle 9i Release 2 software, you must install the Oracle9i Release 2 patches. See “Adding Oracle9i R2 patches - using the installer” on page 141.

Completing post-installation operations

After installing the Oracle software, complete these operations:

- Add Oracle patches.
- Relink the Oracle software.
- Configure network and Oracle listeners.
- Create the Oracle database.
- Test the configuration.

Adding Oracle9i R2 patches - using the installer

Use the following procedure to add Oracle9i Release 2 patches to your node if you have installed Oracle, but have not yet configured Oracle in your cluster. To install Oracle9i Release 2 Patch software, you must have installed Oracle9i Release 2 (9.2.0.1). Before installing the Oracle patch, review the *Patch Set Notes* that accompany the patch set for instructions on installing the patch set and performing the post-installation operations.

Note: For Oracle patchsets 9.2.0.6 and 9.2.0.7 you must copy the patch installation image to a writable file system and specify the path of the file system to the installsfrac utility.

---

1. Launch the installsfrac program. As root user on any one node in the cluster, enter:

   ```
   # cd /opt/VRTS/install
   # installsfrac -configure
   ```

   By default, the installsfrac program uses ssh for remote communication. However, rsh can be used in instead of SSH by using the “-rsh” option with the installsfrac utility.

   ```
   # cd /opt/VRTS/install
   # installsfrac -rsh -configure
   ```

2. Enter the system names on which you want to install Oracle9i software and navigate to the **Install or relink Oracle** menu.
3. Select the version of Oracle software (**Oracle 9iR2**).
4. Select **Install Oracle RDBMS server** to install Oracle software.
5. Enter the value for the DISPLAY environment variable. This value should refer to the address of the X11 server where you want OUI for Oracle installation displayed. For example, use “10.180.94.84:0.0.”

6 In the Oracle user name screen, enter the Oracle UNIX user name when prompted. The installer checks for the user on all systems. For example, use "oracle."

7 In the Oracle Group name screen, enter the name of the Oracle inventory group when prompted. The installer checks for the existence of this group on all systems. For example, use "dba."

8 In the Database Install Image screen, specify the location of the Oracle Database Software Install Image. For example, `/oraimage/9207/Disk1`.

9 In the Oracle base directory screen, specify the base of the Oracle directory structure for OFA compliant databases. For example, use `/app/oracle`.

10 In the Database Home Directory screen, specify the location of $ORACLE_HOME. For example, `/app/oracle/orahome`.

11 In the Patch Level screen, specify the patch level of Oracle Database Software being installed. For example, use "0.7."

12 The installer prints Oracle9i installation information for verification. Answer "y" if the information displayed is correct, answer "n" otherwise. For example:

```
Oracle environment information verification
Oracle Unix user: oracle
Oracle Unix Group: dba
Oracle Release: 9.2
Oracle Patch: 0.7
Oracle Base: /app/oracle
Oracle Home: /app/oracle/orahome
RDBMS Installation Path: /oraimage/9207/Disk1
Is this information correct: [y,n,q] (y)
```

13 The installer now validates the inputs and performs certain setup steps, after which it prompts you to run rootpre.sh on all cluster nodes.

```
Verifying  fileset prerequisites on galaxy ........... ok
Verifying fileset prerequisites on nebula ............ ok
Comparing Oracle UNIX user id on all systems ........ Done
Comparing Oracle group id on all systems ........... Done
Creating SFRAC odm entries on galaxy .................. ok
Creating SFRAC odm entries on nebula .................. ok
```

14 Installer now brings up the OUI for Oracle RDBMS Software Installation.

```
Invoking OUI to install Oracle Software. Please wait .. ok

Now install Oracle database software from the OUI. Refer to
Oracle documentation for installation instructions.

NOTE: Near the end of the installation, Oracle Database
Software installer will prompt you to run the root.sh
script.
```

```
Press <RETURN> here when prompted to run root.sh
```

Note: Do not press Return yet, but wait for the prompt to do so.

15 Enter the data in Oracle's OUI. Follow Oracle's instructions to enter the information.

16 When the Oracle database binaries have been installed, the OUI prompts you to invoke the root.sh script.

Note: Do not run root.sh yet, but return to the installsfrac session from step 13.

17 In the installsfrac session, press Return. The installsfrac utility now verifies the database software installation, copies the SF Oracle RAC libraries to $ORACLE_HOME, and relinks Oracle on each node in the cluster.

```
Verifying binaries in /app/oracle/orahome on galaxy ...ok
Verifying binaries in /app/oracle/orahome on nebula ...ok
Copying SFRAC libskgxn on galaxy .......................ok
Copying SFRAC libskgxn on nebula .......................ok
Copying SFRAC ODM library on galaxy ....................ok
Copying SFRAC ODM library on nebula ....................ok
Copying SFRAC libskgxp on galaxy .......................ok
Copying SFRAC libskgxp on nebula .......................ok
Relinking Oracle on galaxy .............................ok
Relinking Oracle on nebula .............................ok
```

18 Oracle RDBMS Software Installation is now complete. Enter "q" to quit the SF Oracle RAC installer, installsfrac.

19 As root user on each system, invoke the root.sh script:

```
# cd $ORACLE_HOME
# ./root.sh
```

20 Return to the OUI and indicate you have run root.sh.

21 Exit the OUI.

22 After completing installation of Oracle 9i patches. See "Relinking the SF Oracle RAC libraries to Oracle - using the installer." .

Relinking the SF Oracle RAC libraries to Oracle - using the installer

After installing the Oracle base software or Oracle patches, use the configuration program to relink SF Oracle RAC libraries to Oracle.

To relink the SF Oracle RAC libraries to Oracle

1 Launch the installsfrac program. As root user on any one node in the cluster, enter:

```
# cd /opt/VRTS/install
# installsfrac -configure
```

By default, the installsfrac program uses ssh for remote communication. However, rsh can be used in instead of SSH by using the "-rsh" option with the installsfrac utility.

```
# cd /opt/VRTS/install
# installsfrac -rsh -configure
```

2 Enter the system names on which you want to install Oracle9i software and navigate to the **Install or relink Oracle** menu.

3 Select the version of Oracle software (**Oracle 9iR2**).

4 Select **Relink Oracle** to relink Oracle software.

5 In the Oracle user name screen, enter the Oracle UNIX user name when prompted. The installer checks for the user on all systems. For example, use "oracle."

6 In the Oracle Group name screen, enter the name of the Oracle inventory group when prompted. The installer checks for the existence of this group on all systems. For example, use "dba."

7 In the Oracle base directory screen, specify the base of the Oracle directory structure for OFA compliant databases. For example, use `/app/oracle`.

8 In the Database Home Directory screen, specify the location of $ORACLE_HOME. For example, `/app/oracle/orahome.`

9 In the Patch Level screen, specify the patch level of Oracle Database Software being installed. For example, use "0.7."

10 The installer prints Oracle9i installation information for verification. Answer "y" if the information displayed is correct, answer "n" otherwise. For example:

```
Oracle environment information verification
Oracle Unix user: oracle
Oracle Unix Group: dba
Oracle Release: 9.2
Oracle Patch: 0.7
Oracle Base: /app/oracle
Oracle Home: /app/oracle/orahome
RDBMS Installation Path: /oraimage/9201/Disk1
Is this information correct: [y,n,q] (y)
```

11 Press Enter. The installer now copies 5.0 SF Oracle RAC libraries to ORACLE_HOME and relinks Oracle.
The installer reports its actions it performs them.

```
Verifying binaries in /app/oracle/orahome on galaxy ...ok
Verifying binaries in /app/oracle/orahome on nebula ...ok
Copying SFRAC libskgxn on galaxy .......................ok
Copying SFRAC libskgxn on nebula .......................ok
Copying SFRAC ODM library on galaxy ....................ok
Copying SFRAC ODM library on nebula ....................ok
Copying SFRAC libskgxp on galaxy .......................ok
Copying SFRAC libskgxp on nebula .......................ok
Relinking Oracle on galaxy .............................ok
Relinking Oracle on nebula .............................ok
```

12 After relinking is complete, enter “q” to quit the installer.

Configuring networking and Oracle listeners

Most configurations require an Oracle listener (Oracle Network Services Listener) for client access. The listener “listens” for incoming connections to the database.

Create a listener for each server. You can configure the listener to use the base (host) address of the server or a virtual address. A virtual address is typically used if you want to configure a NIC failover arrangement using VCS, or if the instances are running on different nodes.

You can create the `listener.ora` configuration manually or use the NETCA utility from Oracle. In most cases, the default `listener.ora` is sufficient, with the host address updated to reflect the proper host name.

After verifying the listeners start and stop on each node, update the database startup scripts (`pfile`) to use the remote listeners. Configure the database to register with every listener not created on the local node (the local node is automatically configured). Use the `Remote_Listeners` parameter in the startup `pfile`.

Creating the Oracle database

Create the Oracle database on shared storage. Use you own tools or refer to Appendix B, “Creating a starter database” on page 443 for information on using the Oracle `dbca` (Database Creation Assistant) tool to create a database on shared raw VxVM volumes or shared VxFS file systems.

- If you use dbca to create a database, first start the GSD daemon on all nodes:
  ```
  # gsdctl start
  ```
- If you use server manager to create a database, update the SRVM database to reflect the new database.
- Verify the database can start on all selected nodes.

Checking cluster_database flag in Oracle9i parameter file

On each node, confirm that Oracle instances are enabled to run in parallel. For example, set the `cluster_database` flag in the Oracle9i parameter file `$ORACLE_HOME/dbs/init$ORACLE_SID.ora`. The file should contain this line:

```
cluster_database = true
```

You may also enable the instances to run in parallel by editing the spfile. Refer to Oracle documentation.

Testing the configuration

Keep these points in mind to ensure reliable operations in the cluster:

- Start database instances at the same time on all desired nodes. This verification confirms multiple node access to the database.
- Start listeners on all nodes using the host names specified in the `listener.ora` configuration files. This action may require configuring IP aliases on servers.
- Verify all instances are registered with all the listeners. This critical step ensures clients can connect to any listener to access any instance.
- Verify the client systems can access the new database.

Chapter 7

Configuring Oracle9i service groups

Use this chapter to set up VCS to automate the Oracle RAC environment. Topics include:

About SF Oracle RAC service groups

The Veritas Cluster Server (VCS) package, provided as part of the installation ofStorage Foundation for Oracle RAC, provides the ability to automate the entire RAC environment. For example, VCS can be used to automatically start the Cluster Volume Manager and Cluster File System resources within the cluster, bring up IP addresses and the Oracle Listener, mount the file systems with the Oracle binaries, mount the storage for the database instances, and start the database instance. Placing the database under VCS control does not remove the DBA's capability for full control. It simply automates actions to enable the cluster to start up after any outage.

In a Storage Foundation for Oracle RAC cluster, the administrative staff is free to choose how much, or how little, automated control they desire. Less automation means more traditional hands-on interaction, but also requires the administrator to take corrective action in more circumstances. A better approach may be to allow VCS complete startup control to take care of system and power failures and restarts, while still allowing manual control if necessary.

VCS uses installed agents to manage the resources in the Storage Foundation for Oracle RAC environment. Each type of resource has an agent; for example, VCS uses a CFSMount agent for mounting shared file systems, the CVMVolDg agent for activating shared disk groups and monitoring shared volumes, an Oracle agent for starting the Oracle database, an IP agent for setting up and monitoring an IP address, and so on.

The VCS configuration file (`/etc/VRTSvcs/conf/config/main.cf`) contains the information the agents require to manage each resource in the environment. In addition, the configuration file specifies the dependencies between resources; the dependencies that one resource has upon another sets the order in which the resources are started or stopped by the agents.

About CVM and Oracle group configuration

During the installation of Storage Foundation for Oracle RAC, a rudimentary CVM service group is automatically defined in the VCS configuration file, `main.cf`. This CVM service group includes the CVMCluster resource, which enables Oracle9i RAC communications between the nodes of the cluster, and the `CFSfsckd` daemons that control the transmission of file system data within the cluster. If for any reason this group has not been created, you may create it by running the command:

```
/opt/VRTSvxfs/cfs/bin/cfscluster config
```

After installing Oracle9i Release 2 and creating a database, you can modify the CVM service group to include the Oracle listener process, the IP and NIC resources used by the listener process, and the CFSMount and CVMVolDg resources. The NIC and IP resources are used by the listener process to communicate with clients via the public network. The CFSMount and CVMVolDg resources are for the Oracle binaries installed on shared storage. The modification of the CVM service group is discussed in "Modifying the CVM service group in the main.cf file" on page 154.

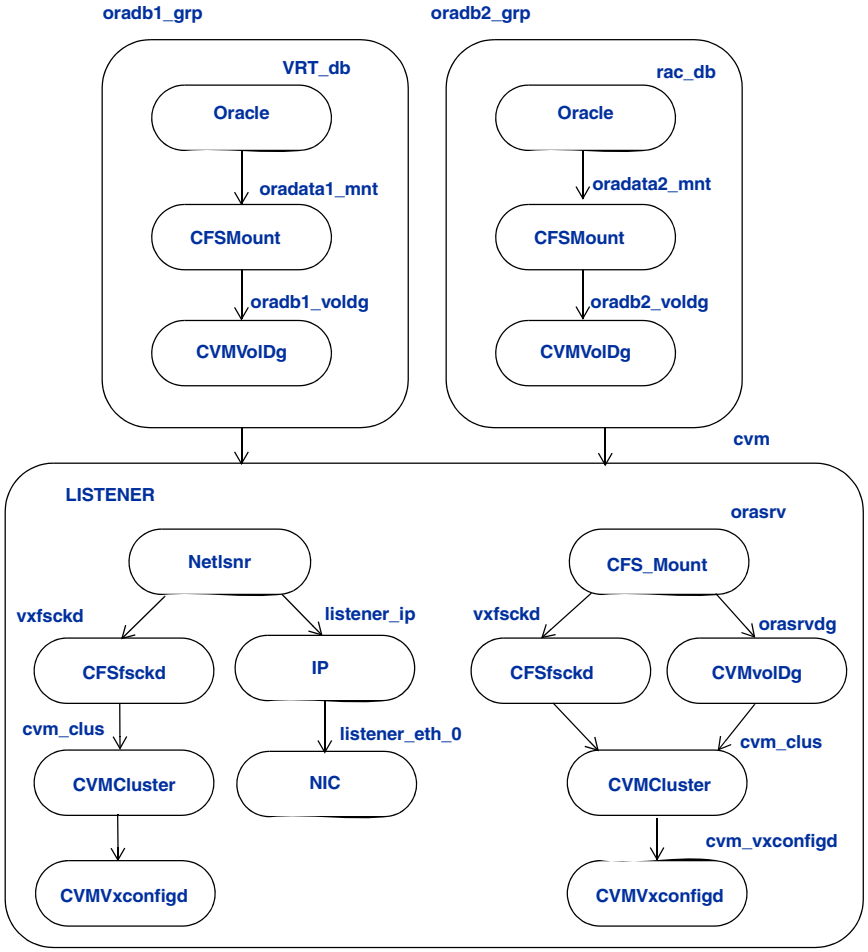
Then you can add the Oracle database service group using the Storage Foundation for Oracle RAC wizard. This service group typically consists of an Oracle database created on shared storage, and the CFSMount and CVMVolDg resources used by the database. There may be multiple Oracle database service groups, one for each database.

VCS service groups for RAC: dependencies

The following illustration shows the dependencies among the resources in a typical RAC environment. The illustration shows two Oracle service groups (`oradb1_grp` and `oradb2_grp`) and one CVM service group (`cvm`). The dependencies among the resources within a group are shown as well as the

dependency that each Oracle group has on the CVM group. This chapter describes how to specify the resources.

Figure 7-1 Dependencies for VCS Service Groups for RAC



Managing Oracle components of the SF Oracle RAC environment

If you choose to use VCS to manage components of the Oracle RAC configuration, you must understand the implications of configuration options for resources that manage the Oracle components to choose the best implementation for your environment.

Configuring the Oracle resource

An Oracle resource starts and stops an Oracle instance, and monitors the process table. In addition to the required attributes for the resource, you can modify these attributes to suit your environment. See the *Veritas High Availability Agent for Oracle Installation and Configuration Guide* for details on the Oracle agent and ways to manage database functions from within VCS.

StartUpOpt attribute

The Oracle agent supports the StartUpOpt attribute to enable you to control how VCS starts a database instance.

- RECOVERDB:Assume that in the event the database node fails for a non-database issue, you do not want the database to start up (come online) if any data file recovery needs to take place. If you set StartUpOpt to RECOVERDB and the system running the database faults, VCS starts the database in recovery mode when it fails to another node.
- CUSTOM: You may want to start the database up in standby mode and apply logs. In this case, you can create an SQL script to perform these actions, and this script is called when you set StartUpOpt to CUSTOM. You must start the script in `/opt/VRTSvcs/bin/oracle` with the name of `start_custom_SID.sql`, where SID is the same as the value of the SID attribute.

ShutDownOpt attribute

Configure the ShutDownOpt attribute for the Oracle resource to control how VCS stops a database instance when the Oracle resource is taken offline. If CUSTOM is specified, a script must exist in `/opt/VRTSvcs/bin/oracle` with the name of the `shut_custom_SID.sql`, where SID is the same as the value of the SID attribute for this Oracle resource.

Other optional Oracle attributes

Review these additional Oracle attributes:

- `AutoEndBkup` takes the data files in the database out of the backup mode when coming online. The default value is 1 (true).
- `EnvFile` speciifies the location of a file sourced by the Oracle agent entry point scripts containing environment variables.
- `Encoding` specifies encoding for other language sets.

Configuring Oracle detail monitoring

These optional Oracle attributes are used to configure detail monitoring:

- `DetailMonitor`: A flag to enable and disable detail monitoring. The default value of 0 prevents detail monitoring. You can set this value higher than zero to control how often detail monitoring occurs.
- `User`: A database user account with updated privileges.
- `Pword`: The password for the database user account.
- `Table`: The name of the database table VCS uses for additional monitoring.
- `MonScript`: The executable script file containing the SQL statements VCS uses when writing to the table.
- `EnvFile`: The file containing environmental variables sources by the agent.

Configuration prerequisites include:

- Create the database user and password for use by VCS.
- Create a test table with the monitored database.
- Create an executable script with SQL statements.

You can use the VCS encryption utility to encrypt database passwords before configuring the Pword attribute in the Oracle agent configuration.

Note: The value of Pword is automatically encrypted when you use the Oracle configuration wizard or one of the VCS GUIs to configure the resource.

◆ To encrypt a password, use the `vcsencrypt` command:

```
# root:/opt/VRTSvcs/bin> vcsencrypt -agent
```

You can configure the optional Oracle error handling file when using detail monitoring to specify VCS response based on the type of database errors. See `/opt/VRTSvcs/bin/Oracle/oraerror.dat`.

Oracle service group

You can configure VCS to support starting, stopping, and monitoring Oracle databases automatically. A database is configured in a VCS service group. Just as

one instance of a database runs per machine, one instance of a parallel VCS service group runs per machine.

Oracle service group dependencies

The Oracle service group has an online local firm dependency on the cvm service group:

- The Oracle instance cannot start until the Oracle binary disk group is imported and the file system is mounted.
- If the IP address and listener process are not available, clients cannot access the database.
- If the cvm service group faults, the Oracle service group comes offline.

Service groups in an SF Oracle RAC environment typically have an online local firm dependency configuration to ensure that the infrastructure for Oracle RAC is online before the database starts.

In an online local dependency, a child service group must be online on a node before a parent service group can come online on the same node. The parent group is taken offline when the child group faults. If the parent service group faults, the child service group continues to run.

Configuring the Oracle service group

Configuring the Oracle service group involves these tools:

- Command-line interface. For example, type:

```
haconf makerw
hagrp -add oradatadg
hagrp -modify oradatadg Parallel 1
hagrp -modify oradatadg SystemList S1 0 S2 1
hagrp -modify oradatadg AutoStartList S1 S2
haconf -dump -makero
```

- Graphical user interface
- Wizard: `/opt/VRTSvcs/bin/hawizard rac`.
- Manual modification of `main.cf`

Defining database fault behavior

Use the resource and service group attributes to control how VCS responds when resources or systems fault:

- If you want the listener to restart and never affect other resources if it faults:
  - Set the listener resource to non-critical.

  - Do not make the listener a child of a critical resource.
  - Set RestartLimit=x.
- If you do not want VCS to forcibly clean up if the database faults:
  - Set ManageFaults=None for the service group.
  - Set notification on fault, no clean, no failover.

Even if a database resource is non-critical, the clean entry point runs if the resource faults. This action shuts down the database and leaves it offline. See the Veritas Cluster Server documentation for details on configuring critical resources, VCS response to faults, and the impact of service group attributes on failover.

Configuring CVM and Oracle service groups manually

CVM and Oracle groups can be configured manually or by using the VCS Oracle RAC Configuration Wizard. This section describes how to manually edit the main.cf file to configure the CVM and Oracle service groups.

To configure CVM and Oracle service groups manually

1. Log in to one system as `root`.
2. Save your existing configuration to prevent any changes while you modify `main.cf`:

   ```
   # haconf -dump -makero
   ```

3. Ensure VCS is not running while you edit `main.cf` by using the `hastop` command to stop the VCS engine on all systems and leave the resources available:

   ```
   # hastop -all -force
   ```

4. Make a backup copy of the `main.cf` file:

   ```
   # cd /etc/VRTSvcs/conf/config
   # cp main.cf main.orig
   ```

5. Using `vi` or another text editor, edit the `main.cf` file, modifying the `cvm` service group and creating `Oracle` service groups using the guidelines in the following sections.

Modifying the CVM service group in the main.cf file

The `cvm` service group is created during the installation of Storage Foundation for Oracle RAC. After installation, the `main.cf` file resembles the example shown in "Oracle9i configuration" on page 426. Because Oracle9i had not been installed, the `cvm` service group includes only resources for the CFSfsckd daemon, the CVMCluster resource, and the CVMVxconfigd daemon.

Adding NetIsnr, NIC, IP CVMVolDg, and CFSMount resources

You must modify the `cvm` service group to add the NetIsnr, NIC, IP, CVMVolDg, and CFSMount resources to the configuration. Refer to "Oracle9i configuration" on page 426 to see a complete example of how a `cvm` group is configured.

To configure NetIsnr, NIC, IP CVMVolDg, and CFSMount resources

Key lines of the configuration file are shown in bold font.

1. Make sure the `cvm` group has the group `Parallel` attribute set to 1. Typically this is already done during installation.

```
.
group cvm (
     SystemList = { galaxy = 0, nebula = 1 }
     AutoFailOver = 0
     Parallel = 1
     AutoStartList = { galaxy, nebula }
     )
.
```

2 Define the NIC and IP resources. The VCS bundled NIC and IP agents are described in *Veritas Cluster Server Bundled Agents Reference Guide*. The device name and the IP addresses are required by the listener for public network communication.

Note: For the IP resource, the `Address` attribute is localized for each node (see "Local CVM and Oracle group attributes" on page 160).

```
NIC listener_nic (
    Device = en0
    NetworkType = ether
    NetworkHosts = { "10.10.11.101", "10.10.11.102" }
    )

IP listener_ip
    Device = en0
    Address @galaxy = "10.10.11.1"
    Address @nebula = "10.10.11.2"
    NetMask = "255.255.240.0"
    )
```

3 Define the Netlsnr resource. The Netlsnr listener agent is described in detail in the *Veritas™ High Availability Agent for Oracle Installation and Configuration Guide*.

Note: The `Listener` attribute is localized for each node.

```
Netlsnr listener (
    Owner = oracle
    Home = "/app/oracle/orahome"
    TnsAdmin = "/app/oracle/orahome/network/admin"
    Listener @galaxy = listener_galaxy
    Listener @nebula = listener_nebula
    ListEnvFile = "/opt/VRTSvcs/bin/Netlsnr/envfile"
    MonScript = "./bin/Netlsnr/LsnrTest.pl"
    )
    .
```

4 You must configure the CVMVolDg and CFSMount resources in the *cvm* group for the Oracle binaries installed on shared storage. Refer to the appendix Appendix C, "Agent reference" on page 449 for description of CVMVolDg and CFSMount agents.

```
CFSMount oradata_mnt (
    Critical = 0
    MountPoint = "/oradata"
    BlockDevice = "/dev/vx/dsk/oradatadg/oradatavol"
    )

CVMVolDg oradata_voldg (
    CVMDiskGroup = oradatadg
    CVMVolume = { oradatavol }
    CVMActivation = sw
    )
```

5 Define the dependencies of resources in the group. The dependencies are specified such that the Netlsnr resource requires the IP resource that, in turn, depends on the NIC resource. The Netlsnr resource also requires the CFSMount resource. The CFSMount resource requires the daemons and vxfsckd used by the cluster file system. The CFSMount resource also depends on the CVMVolDg resource, which, in turn, requires the CVMCluster resource for communications within the cluster.

The "VCS service groups for RAC: dependencies" on page 148 visually shows the dependencies specified in the following statements:

```
.
.
listener requires listener_ip
listener_ip requires listener_nic
orasrvm_voldg requires cvm_clus
orasrvm_mnt requires orasrvm_voldg
orasrvm_mnt requires vxfsckd
vxfsckd requires cvm_clus
cvm_clus requires cvm_vxconfigd
.
.
```

Adding Oracle resource groups in the main.cf file

For a complete description of the VCS Oracle enterprise agent, refer to the document, *Veritas High Availability Agent for Oracle Installation and Configuration Guide*. That document includes instructions for configuring the Oracle and Netlsnr agents.

The VCS Enterprise Agent for Oracle version 5.0 is installed when you run `installsfrac`. When you refer to the *Veritas High Availability Agent for Oracle Installation and Configuration Guide,* ignore the steps described in the section "Installing the Agent Software."

To add Oracle resource groups in the main.cf file

1 Using the "Oracle9i configuration" on page 426 as an example, add a service group to contain the resources for an Oracle database. For example, add the group `oradb1_grp`. Make sure you assign the `Parallel` attribute a value of 1.

```
.
group oradb1_grp (
SystemList = { galaxy = 0, nebula = 1 }
AutoFailOver = 0
Parallel = 1
AutoStartList = { galaxy, nebula }
)
```

2 Create the CVMVolDg and CFSMount resource definitions. See Appendix C, "Agent reference" on page 449 for a description of these agents and their attributes.

```
.
.
    CVMVolDg oradata_voldg (
    CVMDiskGroup = oradatadg
    CVMVolume = { oradatavol }
    CVMActivation = sw
    )

CFSMount oradata_mnt (
    Critical = 0
    MountPoint = "/oradata"
    BlockDevice = "/dev/vx/dsk/oradatadg/oradatavol"
    )
.
.
```

3 Define the Oracle database resource. Refer to the *Veritas High Availability Agent for Oracle Installation and Configuration Guide* for information on the VCS enterprise agent for Oracle.

Note: Oracle attributes `Sid` and `Pfile` must be set locally, that is, they must be defined for each cluster system.

```
.
Oracle ora1 (
    Critical = 0
    Sid @galaxy = vrts1
    Sid @nebula = vrts2
    Owner = oracle
    Home = "/app/oracle/orahome"
    Pfile @galaxy = "/app/oracle/orahome/dbs/initvrts1.ora"
    Pfile @nebula = "/app/oracle/orahome/dbs/initvrts2.ora"
    )
```

.

4 Define the dependencies for the Oracle service group. Note that the Oracle database group is specified to require the `cvm` group, and that the required dependency is defined as "`online local firm`," meaning that the `cvm` group must be online and remain online on a system before the Oracle group can come online on the same system. Refer to the *Veritas Cluster Server User's Guide* for a description of group dependencies.
Refer to "VCS service groups for RAC: dependencies" on page 148.

```
.
.
requires group cvm online local firm
ora1 requires oradata_mnt
oradata_mnt requires oradata_voldg
.
.
```

See the Appendix A, "Sample VCS configuration files for SF Oracle RAC" on page 425 for a complete example. You can also find the complete file in `/etc/VRTSvcs/conf/sample_rac/main.cf`.

Additional RAC Processes Monitored by the VCS Oracle Agent

For shallow monitoring, the VCS Oracle agent monitors the `ora_lmon` and `ora_lmd` processes in addition to the `ora_dbw`, `ora_smon`, `ora_pmon`, and `ora_lgwr` processes.

Saving and checking the configuration

When you finish configuring the CVM and Oracle service groups by editing the `main.cf` file, verify the new configuration.

To save and check the configuration

1 Save and close the `main.cf` file.

2 Verify the syntax of the file `/etc/VRTSvcs/conf/config/main.cf`:

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

3 Start the VCS engine on one system:

```
# hastart
```

4 Type the command `hastatus`:

```
# hastatus
```

5 When "`LOCAL_BUILD`" is listed in the message column, start VCS on the other system:

```
# hastart
```

6 Verify that the service group resources are brought online. On one system, enter:

```
# hagrp -display
```

Local CVM and Oracle group attributes

The following table lists attributes that must be defined as local for the CVM and Oracle service groups (note that each attribute has string-scalar as the type and dimension).

Resource	Attribute	Definition

IP	Address	The virtual IP address (not the base IP address) associated with the interface. For example:
		`Address @sysa = "192.2.40.21"`
		`Address @sysb = "192.2.40.22"`

Netlsnr	Listener	The name of the Listener. For example:
		`Listener @sysa = LISTENER_a`
		`Listener @sysb = LISTENER_b`

Oracle	Sid	The variable $ORACLE_SID represents the Oracle system ID. For example, if the SIDs for two systems, `sysa` and `sysb`, are VRT1 and VRT2 respectively, their definitions would be:
		`Sid @sysa = VRT1`
		`Sid @sysb = VRT2`

Oracle	Pfile	The parameter file: `$ORACLE_HOME/dbs/init$ORACLE_SID.ora`. For example:
		`Pfile @sysa = "/oracle/VRT/dbs/initVRT1.ora"`
		`Pfile @sysb = "/oracle/VRT/dbs/initVRT2.ora"`

Oracle	Table	The table used for in-depth monitoring by `User/PWord` on each cluster node. For example:
		`Table @sysa = vcstable_sysa`
		`Table @sysb = vcstable_sysb`
		Using the same table on all Oracle instances is not recommended. Using the same table generates IPC traffic and could cause conflicts between the Oracle recovery processes and the agent monitoring processes in accessing the table.
		Note: `Table` is only required if in-depth monitoring is used. If the `PWord` varies by RAC instance, it must also be defined as local.

If other attributes for the Oracle resource differ for various RAC instances, define them locally as well. These attributes may include the Oracle resource attributes `User`, `PWord`, the `CVMVolDg` resource attribute `CVMActivation`, and others.

Modifying the VCS configuration

For additional information and instructions on modifying the VCS configuration by editing the `main.cf` file, refer to the *Veritas Cluster Server User's Guide*.

Configuring the service groups using the wizard

You can use the configuration wizard to configure the VCS service groups for Storage Foundation for Oracle RAC environment. The wizard enables you to create the service group for Oracle and modify the CVM service group.

The Oracle9i RAC configuration wizard guides you through the creation of an Oracle service group and the definition of the Oracle, CFSMount, and CVMVolDg resources. It adds the `Netlsnr` resources to the existing CVM group. If the listeners use the virtual IP, the wizard also adds the IP and NIC resources to the CVM group.

The wizard configures the Oracle service group to depend on the CVM group with an online-local-firm dependency.

Verifying configuration requirements

Before starting the Wizard, you can verify that your Oracle installation can be configured. Review the requirements listed below. Also, you need to provide the wizard information as it proceeds. Make sure you have that information at hand.

Prerequisites:

- ✔ Oracle RAC instances and listeners must be running on all cluster nodes.
- ✔ The database files of all instances must be on shared storage, that is, in a cluster file system.
- ✔ Each Oracle instance must be associated with a listener. The listener may be configured to listen to either the base IP or a virtual IP.
- ✔ The IP addresses and host names specified in the files `listener.ora` and `tnsnames.ora` must be the same.
- ✔ If detail monitoring is to be used for a database instance, the table used for detail monitoring must be set up, with user and password assigned.

Required Information:

- ✔ The names of the database instances to be configured
- ✔ The information required for the detail monitoring configuration
- ✔ The location of the pfile for each instance

Establishing graphical access for the wizard

The configuration wizard requires graphical access to the VCS systems where you want to configure service groups. If your VCS systems do not have monitors, or if you want to run the wizards from a remote UNIX system, do the following:

To establish graphical access from a remote system

1. From the remote system, (`jupiter`, for example), run:

   ```
   # xhost +
   ```

2. Do one of the following, depending on your shell:

   - For Bourne shell (`bash`, `sh`, or `ksh`), run this step on one of the systems where the wizard is to run, for example `galaxy`:

     ```
     # export DISPLAY=jupiter:0.0
     ```

   - For C shell (`csh`), run this step

     ```
     # setenv DISPLAY jupiter:0.0
     ```

3. Verify that the DISPLAY environment variable has been updated:

   ```
   # echo $DISPLAY
   jupiter:0.0
   ```

To set up pre-Oracle configuration for the wizard

1. Become `oracle` user:

   ```
   # su - oracle
   ```

2. Start up the listeners on all cluster nodes:

   ```
   # $ORACLE_HOME/bin/lsnrctl start LISTENER_X
   ```

3. Start up the Oracle RAC instances on each node. For example:

   ```
   # export ORACLE_SID=rac1
   # sqlplus "/ as sysdba"
   sqlplus> startup
   ```

4. Set the `TNS_NAMES` parameter if the `listeners.ora` and `tnsnames.ora` files are not under the default directory.

Creating service groups using the configuration wizard

To create service groups using the configuration wizard

1. Log on to one of your VCS systems as root. The configuration wizard for Oracle9i RAC is started at the command line.

2. Start the configuration wizard.

   ```
   # /opt/VRTSvcs/bin/hawizard rac
   ```

3 In the Welcome screen that displays, the prerequisites for configuration and the information you will need to complete the configuration is displayed.

- If your configuration does not meet the configuration requirements, you can stop the wizard by pressing **Cancel**. Take the necessary steps to meet the requirements and start the wizard again. See step 1
- If you are ready to configure Oracle service group, press **Next** on the Welcome screen.

  The wizard begins discovering the current Oracle RAC information before proceeding with the next screen.

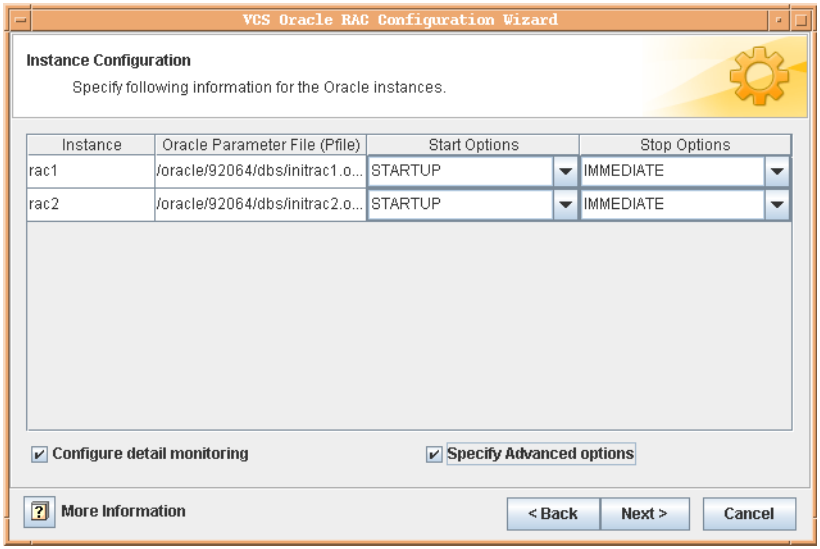
  If the wizard does not find all databases and listeners running on all systems in the cluster, it halts with an error, indicating the problem. Press **Cancel**, and start the wizard again after you correct the problem.

4 In the Wizard Options dialog box, **Create RAC service group** option.

5 Enter a name for the RAC service group in the **Service group name** box and click **Next**.

6 In the Database Selection dialog box, select a database and click Next.

7 In the Instance Configuration dialog box, specify the basic database instance information for the databases you selected.



a For each database instance discovered, basic configuration information is displayed. If necessary, double click in a field to select and edit its contents.

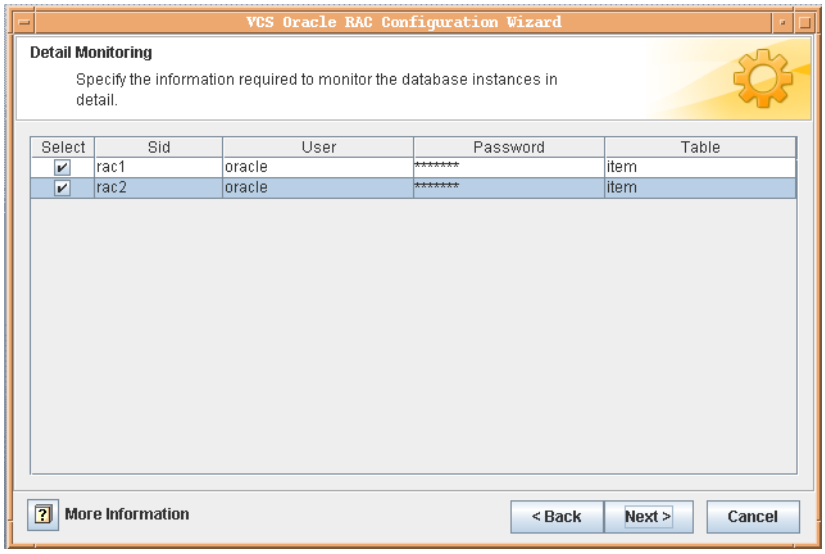
- **Instance** name: Each instance is listed in the left hand column.
- **Oracle Parameter File (Pfile)**: The file that is used to start Oracle. The default location for a given instance is listed. Edit the information if necessary.
- **Start Options**: the default is STARTUP_FORCE.
- **Stop Options**: Accept the displayed IMMEDIATE option, or select an option from the drop-down menu.

b Select **Configure detail monitoring** to enable monitoring in detail. If you check enable detail monitoring, be sure you have previously set up the database table, user, and password for the agent to use during monitoring. See step 8 on page 164. If you are not set up for detail monitoring, do not select it.

c Select **Specify Advanced options** to define monitoring options such as environment variables, `Encoding`, and an `AutoEndBkup` parameter. See step 8.

d Click **Next**.

8 If you choose to monitor the database in detail, The Detail Monitoring dialog box displays.

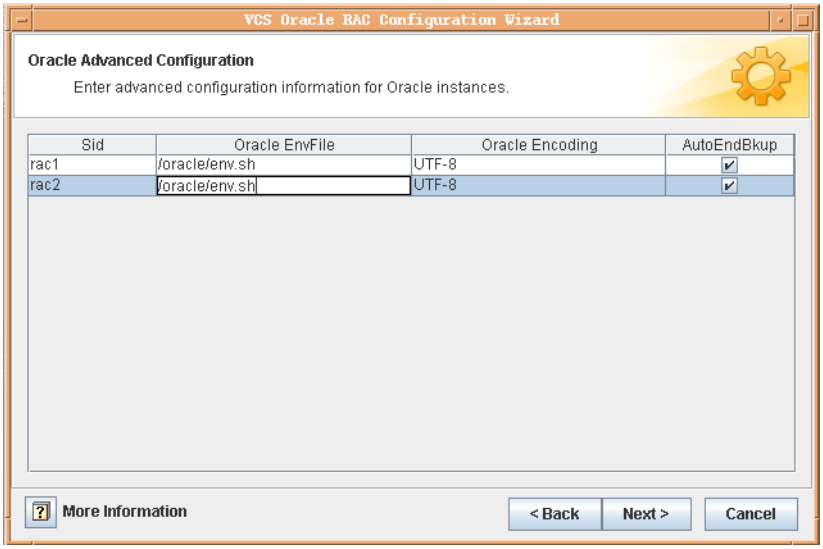


For each database instance identified by its **Sid**, this screen displays fields for defining the attributes that enable detail monitoring of the Oracle database resource. You do not have to enable detail monitoring on all

instances, but for each instance for which you check **Select**, all fields are required:

- **User**: Oracle user, which the Oracle agent uses to log on to monitor the health of the database.
- **Password**: Password for the Oracle user. The wizard takes care of encrypting passwords.
- **Table**: Name of the database table to be used by the Oracle agent monitor.

9 The Oracle Advanced Configuration dialog box is displayed if you have checked **Specify Advance options** at the bottom of the Instance Configuration Screen.



For each database instance identified by its **Sid**, this dialog box displays fields for configuring the advanced attributes of the Oracle service group. You may select which database instance you want to configure advance attributes for, and which attributes you want to define, and click **Next** when you are done.

Advanced attributes include:

- **Oracle EnvFile**: the source file used by the agent entry point scripts.
- **Oracle Encoding**: the operating system encoding that corresponds to Oracle encoding for the displayed Oracle output; the encoding value must match the encoding value used by the Netlsnr configuration.

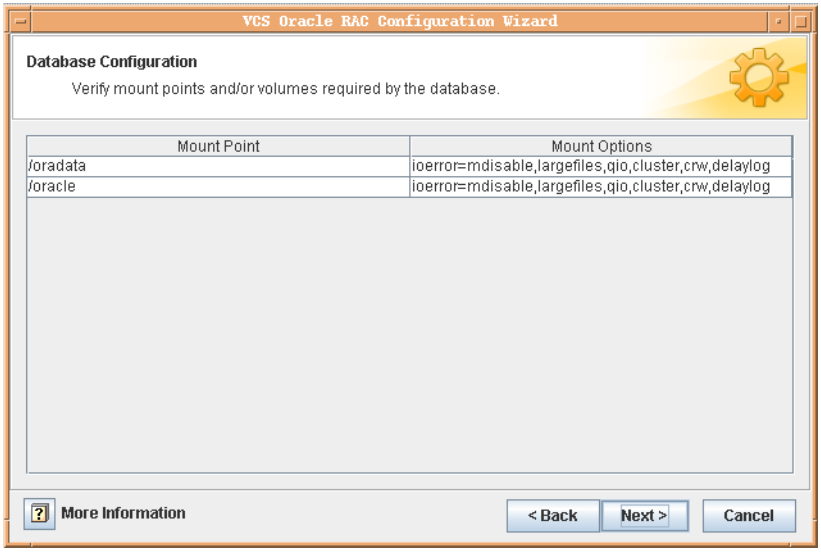
- **AutoEndBkup**: specifies that datafiles in the database are taken out of the backup mode when instance is brought online.

See the *Veritas™ High Availability Agent for Oracle Installation and Configuration Guide* for a complete description of the `EnvFile`, `Encoding`, and `AutoEndBkup` attributes.

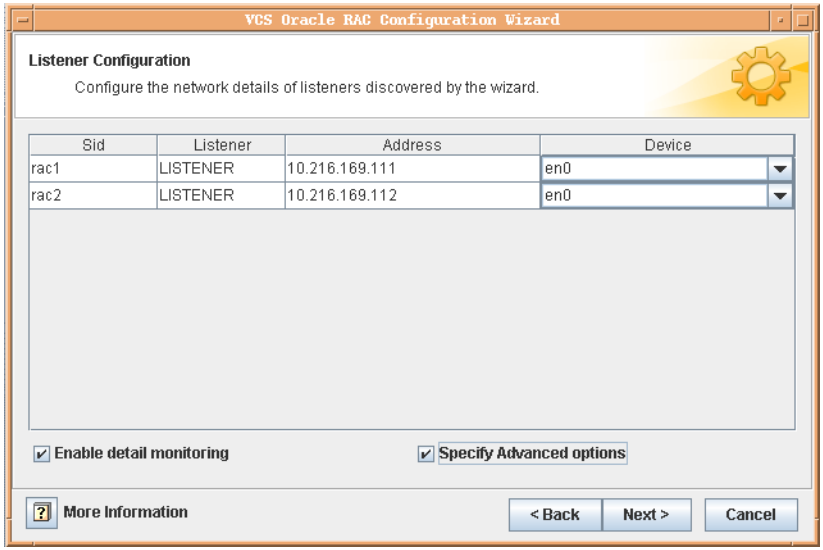
10 In the Monitor Option Configuration dialog box, for each database instance you can specify whether to use the **Process check** (the default) or **Health** check option for monitoring method. Click **Next** to continue.

11 In the Database Configuration dialog box, the wizard discovers the **Mount Point** of the database if it installed on a cluster file system and displays it on the Database Configuration screen. If the database exists on raw volumes, the wizard discovers the volumes.

Confirm or modify the **Mount Options** displayed and click **Next**.



12 The Listener Configuration dialog box displays the **Sid** and name of the **Listener** corresponding to each database instance, as well as the IP **Address** and **Device** name used by each listener. Typically, you cannot change this information, only verify it.

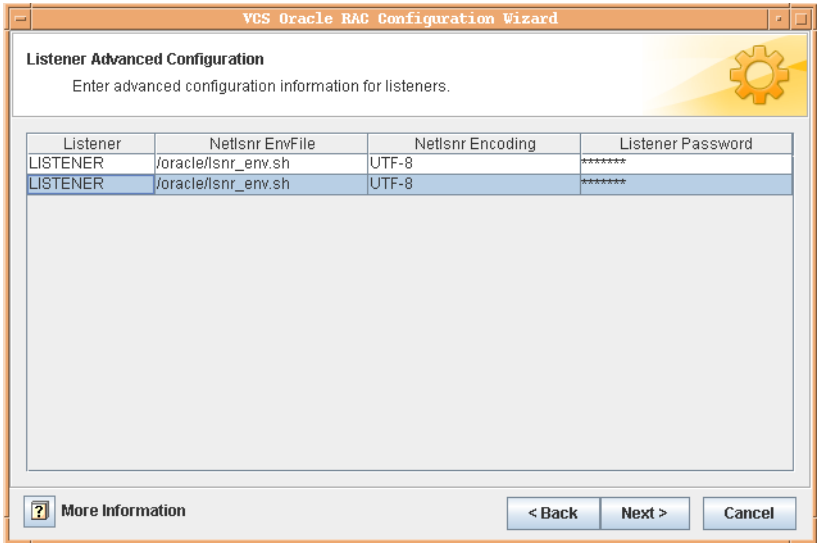


a You can choose to configure detail monitoring for the Netlsnr agent by clicking the **Enable detail monitoring** checkbox. The wizard uses the monitor script `/opt/VRTSvcs/bin/Netlsnr/LsnrTest.pl` to monitor the listeners in detail.

b You can choose **Specify Advanced options** to set up an `EnvFile` (to define environment variables), specify `Encoding` and the `LsnrPwd` parameters. See step 13 on page 168.

c Click **Next**.

13 The Listener Advanced Configuration dialog box displays if you have checked **Specify Advanced options** at the bottom of the Listener Configuration dialog box. If you have set up the listener to use the base, or host, IP address, the wizard displays a warning message when you press **Next** on the Listener Configuration dialog box. You must click **Yes** to proceed.

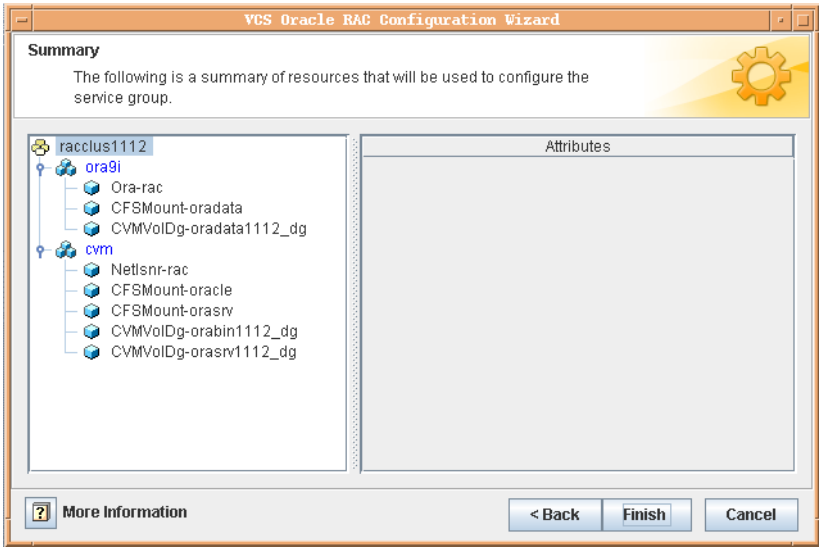


For each **Listener** identified by name, this screen displays fields for defining the advanced attributes of the Netlsnr resource.

- **NetIsnr EnvFile**: the name of the source file used by the agent entry point scripts; this file must exist.
- **NetIsnr Encoding**: the operating system encoding that corresponds to Oracle encoding for the displayed Oracle output; the encoding value must match the encoding value used by the Oracle configuration.
- **Listener Password**: the password used for `Netlsnr`; must specify the password as it appears in the `listener.ora` file. The wizard takes care of encrypting passwords.

Click **Next**.

14 In the Service Group Summary dialog box, the wizard displays the configuration after you have configured the database and listener resources.



Click a resource within the service group to display its attributes and their values. For example, if you click the name of the Oracle resource, `Ora-racj1`, the wizard displays details of the Oracle resource.

Attributes for the CFSMount resource show dependencies. The NetLsnr resource is configured as part of the CVM service group. The CVM service group also contains other resources, which may not be displayed by the wizard because the wizard does not control them.

Change names of resources, if desired; the wizard assigns unique names to resources based on their respective name rules. To edit a resource name, select the resource name and click on it, and press Enter after editing each attribute.

15 Review the configuration and Click **Finish**.

The wizard implements the configuration changes. The wizard creates the Oracle service group, and adds the `Netlsnr` resource to the CVM configuration.

Modifying service groups using the configuration wizard

Once an Oracle RAC service group is created on a system, the configuration wizard can be used to modify the service group's Oracle, Netlsnr, and CVM components.

Note: If modification of underlying mount point or volume information is necessary, the mount points or volumes must be deleted and added in the Oracle database before the wizard is started. Then, the wizard discovers the new information.

Prerequisites

✔ To modify some resources, you must make the changes in the Oracle database before you start the RAC configuration wizard. When you start the wizard, it discovers the new information. This applies to:

- Adding or removing the database mount points
- Adding or removing shared volumes

✔ To modify network resources, make sure that the service group is offline.

✔ To add or remove database files from your configuration, make sure that the service group is online.

To modify the service group configuration

1. Start the RAC configuration wizard as root on the VCS system:
   # **/opt/VRTSvcs/bin/hawizard rac**
2. On the Welcome window, click **Next**.
3. In the Wizard Options dialog box, select the **Modify RAC service group** option, select the service group to be modified, and click **Next**.
4. Follow the wizard instructions and make modifications according to your configuration.
   See "Modifying service groups using the configuration wizard" on page 169.

Chapter 8

Adding and removing nodes in Oracle9i clusters

You can use this chapter to add to or remove a node from cluster running Oracle9i. It contains the following topics:

- Adding a Node to an Oracle9i AIX SF Oracle RAC cluster.
- Adding a Node to an Oracle9i AIX SF Oracle RAC cluster.

Adding a node to an Oracle9i SF Oracle RAC cluster

The examples in these procedures describes adding one node to a two-system cluster.

- "Installing and Configuring Oracle on the new node" on page 179
- "Starting VCS on the new system" on page 180
- "Starting GSD on the new system" on page 181
- "Configuring a new Oracle9i instance" on page 181
- "Adding the new system to the SFDB repository" on page 183

Checking system requirements for a new node

Make sure that the new systems you add to the cluster meet all of the requirements for installing and using Storage Foundateion for Oracle RAC.

✔ The new system must have the identical operating system and patch level as the existing systems. Refer to "SF Oracle RAC prerequisites" on page 67.

✔ Make sure you use a text window of 80 columns minimum by 24 lines minimum; 80 columns by 24 lines is the recommended size for the optimum display of the `installsfrac` script.

Modifying /etc/pse.conf to enable the dlpi driver

The file /etc/pse.conf must be configured to enable the Streams DLPI driver. The following procedure describes checking the file and modifying it, if necessary, and loading the dlpi driver.

1. Check to see if the Ethernet driver is configured in the /etc/pse.conf file:
   ```
   # egrep 'ethernet driver' /etc/pse.conf
   ```
2. In the output, examine the line containing the "ethernetdriver" expression:
   ```
   #d+ dlpi en /dev/dlpi/en # streams dlpi ethernet driver
   ```
3. If the comment symbol ("#") precedes the line, the DLPI driver is not configured. Using vi or another text editor, edit the file:
   ```
   # vi /etc/pse.conf
   ```
4. Find the section in the file labeled "#PSE drivers" and look for the line shown in step 2. Uncomment the line by removing the initial "#" symbol.
5. Save and close the file.
6. To load the dlpi driver, enter:
   ```
   # strload -f /etc/dlpi.conf
   ```

Physically adding a new system to the cluster

When you physically add the new system to the cluster, it must have private network connections to two independent switches used by the cluster and be

connected to the same shared storage devices as the existing nodes. Refer to the *Veritas Cluster Server Installation Guide*.

After installing Storage Foundation for Oracle RAC on the new system and starting VxVM, the new system can access the same shared storage devices. It is important that the shared storage devices, including coordinator disks, are exactly the same among all nodes. If the new node does not see the same disks as the existing nodes, it will be unable to join the cluster.

Installing Storage Foundation for Oracle RAC on the new system

Use the following procedure to install SF Oracle RAC software on the new system.

To install Storage Foundation for Oracle RAC without configuration

1. Log in to the new system as root. For example, the new system is `saturn`.
2. Insert the software disc containing the product into the new system's media drive.
3. Create a mount point directory for the software disc and mount it.

```
# mkdir /cdrom
# mount -V cdrfs -o ro /dev/cd0 /cdrom
```

4. Change to the directory containing the `installsfrac` script:

```
# cd /cdrom/dvd1/sfrac
```

5. Start the `installsfrac` script:

```
# ./installsfrac -installonly
```

6. Enter the names of the systems separated by spaces where you want to install the software:

```
Enter the system names separated by spaces on which to
install SFRAC: saturn
```

7. The installer verifies that the local node can communicate with remote nodes, discovers any pre-existing SF Oracle RAC installation, installs licensing packages, and prompts you to enter the license key for SF Oracle RAC.

```
Enter a SFRAC license key for saturn:
XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXX
XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXX successully registered on
on saturn
SFRAC license registered on saturn

Do you want to enter another license key for saturn?
[y,n,q,?] (n)
```

8 Respond to the script as it verifies system requirements and installs the software. If requirements for installation are not met, the utility stops and indicates the actions required to proceed with the process.

- The script checks whether the required operating system patches are installed. The installer notes missing patches and recommends to stop the installation and install them.
  See "Software requirements" on page 69.
- The script checks for the required file system space.
- The script checks for the presence of processes that could conflict with the installation.

9 The installer now lists all the filesets in the SF Oracle RAC product and then proceeds to install them.

To start / enable VxVM

1 To start Veritas Volume Manager on the new node, use the `vxinstall` utility. As you run the utility, answer "N" to prompts about licensing because you installed the appropriate license when you ran the `installsfrac` utility

```
# vxinstall
VxVM vxinstall INFO V-5-2-2541 Licensing information:
    System host ID: 0xab69380
    Host type: xxxxxx

Some licenses are already installed.  Do you wish to review them
[y,n,q,?] (default: y) n

Do you wish to enter another license key [y,n,q,?] (default: n)

VxVM vxinstall INFO V-5-2-1807   Populating VxVM DMP device
directories ....
```

2 Answer "N" when prompted to setup a default disk group.

```
Do you want to setup a system wide default disk group? [y,n,q,?]
(default: y) n

  Starting the relocation daemon, vxrelocd.
  Starting the cache daemon, vxcached.
  Starting the diskgroup config backup daemon, vxconfigbackupd.
  Starting the dg monitoring deamon, vxvvrsecdgd.

  VxVM vxinstall INFO V-5-2-2291 The installation is
  successfully completed.
```

3 Verify that the daemons are up and running. Enter the command:

```
# vxdisk list
```

The output should display the shared disks without errors.

Configuring LLT, GAB, VCSMM, and VXFEN drivers

To configure LLT, GAB, VCSMM, and VxFEN drivers

1. Edit the file `/etc/llthosts` on the two existing systems. Using `vi` or another text editor, add the line for the new node to the file. The file should resemble:

   ```
   0 galaxy
   1 nebula
   2 saturn
   ```

2. Copy the `/etc/llthosts` file from one of the existing systems over to the new system. The `/etc/llthosts` file must be identical on all systems in the cluster. For example, on saturn, enter:

   ```
   # scp galaxy:/etc/llthosts /etc/llthosts
   ```

3. Create an `/etc/llttab` file on the new system. For example:

   ```
   set-node saturn
   set-cluster 7
   link en1 /dev/dlpi/en:1 - ether --
   link en2 /dev/dlpi/en:2 - ether --
   ```

   Except for the first line that refers to the system, the file resembles the `/etc/llttab` files on the other two systems. The cluster ID must be the same as that for the existing cluster.

4. Use `vi` or another text editor to create the file `/etc/gabtab` on the new system. It should resemble the following example:

   ```
   /sbin/gabconfig -c -nN
   ```

   Where *N* represents the number of systems in the cluster. For a three-system cluster, *N* would equal 3.

5. Edit the `/etc/gabtab` file on each of the existing systems, changing the content to match the file on the new system.

6. Set up the `/etc/vcsmmtab` and `/etc/vxfendg` files on the new system by copying them from one of the other existing nodes:

   ```
   # rcp galaxy:/etc/vcsmmtab /etc
   # rcp galaxy:/etc/vxfendg /etc
   ```

7. Run the following commands to start LLT and GAB on the new node:

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

8. On the new node, start the VXFEN, VCSMM, and LMX drivers. Use the following commands in the order shown:

   ```
   # /etc/init.d/vxfen.rc start
   # /etc/init.d/vcsmm.rc start
   # /etc/init.d/lmx.rc start
   ```

9 On the new node, start the GMS and ODM drivers. Use the following commands in the order shown:

```
# /etc/rc.d/rc2.d/S99odm start
```

10 On the new node, verify that the GAB port memberships are a, b, d, and o. All systems should be in the membership. Run the command:

```
# /sbin/gabconfig -a
GAB Port Memberships
===============================================================
Port a gen 4a1c0001 membership 01
Port b gen g8ty0002 membership 01
Port d gen 40100001 membership 01
Port o gen f1100002 membership 01
```

Configuring CVM on Existing Cluster

You can modify the VCS configuration using two methods. You can use the VCS GUI (Cluster Manager) or you can use the command line, as illustrated in the following example. Please refer to the *Veritas Cluster Server User's Guide* for details about how to configure VCS.

To configure CVM

1 Log in as root on one of the existing cluster nodes.

2 Check the groups dependent on the CVM service group:

```
# hagrp -dep cvm
Parent          Child           Relationship
oradb1_grp      cvm             online local firm
oradb2_grp      cvm             online local firm
```

3 Make the VCS configuration writable:

```
# haconf -makerw
```

4 Add the new system to the cluster:

```
# hasys -add saturn
```

5 If the `ClusterService` service group is configured, add the new system to its system list and specify a failover priority of 2:

```
# hagrp -modify ClusterService SystemList -add saturn 2
```

6 If the `ClusterService` service group is configured, add the new system to the service group's `AutoStartList`:

```
# hagrp -modify ClusterService AutoStartList galaxy nebula
saturn
```

7 Add the new system to the `cvm` service group's system list and specify a failover priority of 2:

```
# hagrp -modify cvm SystemList -add saturn 2
```

8 Add the new system to the `cvm` service group's `AutoStartList`:

```
# hagrp -modify cvm AutoStartList galaxy nebula saturn
```

9 Add the new system and its node ID (refer to the `/etc/llthosts` changes in "Configuring LLT, GAB, VCSMM, and VXFEN drivers" on page 175) to the `cvm_clus` resource:

```
# hares -modify cvm_clus CVMNodeId -add saturn 2
```

10 If the IP resource is *not* part of the `cvm` service group, skip to the next step. If the IP is part of the `cvm` service group, add the new system's virtual IP address to the IP resource:

```
# hares -modify listener_ip Address 10.182.2.130 -sys saturn
```

11 If the listener name is the default, skip this step. Otherwise, add the local listener name to the Netlsnr resource:

```
# hares -modify LISTENER Listener listener_saturn -sys saturn
```

12 Save the new configuration to disk:

```
# haconf -dump -makero
```

13 On each of the existing nodes, run the following command to enable them to recognize the new node:

```
# /etc/vx/bin/vxclustadm -m vcs -t gab reinit
```

14 On one of the existing nodes, run the following command to ensure the new node is recognized:

```
# /etc/vx/bin/vxclustadm nidmap
Name CVM Nid CM Nid State
galaxy 0 0 Joined: Slave
nebula 1 1 Joined: Master
saturn 2 2 Out of Cluster
```

Creating Oracle user and groups on the new node

1 1. On the new node, create user and groups for Oracle. Be sure to assign the same user ID, group ID, and home directory as exists on the existing cluster systems. For example, enter:

```
# mkgroup id='101' dba
# mkgroup id='102' oper
# mkuser id='1001' pgrp='dba' groups='oper' \ home='/opt/oracle'
oracle
```

2 Create a password for the user oracle:

```
# passwd oracle
```

3 Log in as user "oracle" and change the password.

```
$ passwd
```

Setting shell limits for “oracle” user

Use smitty to verify that the shell limits for user “oracle” are set per the following table:

Soft FILE size	-1
Soft CPU time	-1
Soft Data Segment	-1
Soft STACK size	-1

To set the shell limits

1. Log in to a system as root user and enter:
   ```
   # smitty chuser
   ```
2. For the “USER Name” field, enter: oracle.
3. Scroll down the list of attributes and edit their values. See the table above.
4. Press F10 to save the changes.
5. Perform the steps above on each system.

Configuring maximum user Processes

The maximum number of processes per user must be set 2048 or greater. Use the chdev command to configure maxuproc on each system. For example:

```
# chdev -l sys0 -a maxuproc='2048'
sys0 changed
```

Verifying the user “nobody” exists

Verify the user “nobody” exists on each system in the cluster:

```
# id nobody
uid=4294967294(nobody) gid=4294967294(nobody)
```

Creating file system for Oracle binaries

1. If Oracle binaries are installed on shared storage on the existing cluster, skip to the next step. If Oracle binaries are installed locally on the existing cluster systems, you must create a local filesystem on the new node as well. Refer to “Creating a disk group and file system for the Oracle software” on page 134.
2. If the Oracle binaries are installed on shared storage on the existing cluster, do the following to access the shared filesystem on the new node.

As root user on the new system, create the mountpoint for the filesystem containing the Oracle binaries.

```
# mkdir /app
```

Accessing shared Filesystem/Volumes for Database/SRVM

1. If the Oracle database and SRVM are located on CVM volumes, no setup steps are required to access the CVM volumes from the new system. The new system will be able to access the CVM volumes when VCS is started. You may skip this section.
2. If the Oracle database and SRVM are located on CFS filesystems, you must create a mountpoint for the filesystems.
   a. To create a mountpoint for accessing the Oracle database:
      ```
      # mkdir /oradata
      ```
   b. To create a mountpoint for accessing the SRVM file:
      ```
      # mkdir /orasrv
      ```

Installing and Configuring Oracle on the new node

1. 1. If Oracle binaries are installed on shared storage on the existing cluster, do the following:
   a. On the new system, log in as “root” and copy the directory, /var/opt/oracle, from one of the existing system in place.
      ```
      # rcp -rp galaxy:/var/opt/oracle /var/opt
      ```
   b. On one of the existing systems, open a new window and log in as “oracle”. Edit the listener.ora file and add the IP address (or the virtual IP address) for the new system. For example, where the IP address of the new system is 192.2.40.23, create a section for LISTNER_saturn that resembles:
      ```
      LISTENER_saturn =
             (DESCRIPTION_LIST =
              (DESCRIPTION =
               (ADDRESS_LIST =
                (ADDRESS = (PROTOCOL = TCP)(HOST =
                     192.2.40.23)(PORT = 1521)
               )
              )
             )
            )
      ```
2. If Oracle binaries are installed locally on the existing cluster systems, you must install Oracle on the new system’s local disk in the same location as it is installed on the existing systems.

a On the new system, as root user, create the directory, /var/opt/oracle:

```
# mkdir -p /var/opt/oracle
```

b Follow the instructions in the chapter on installing Oracle9i. Key points to note:

- To install Oracle binaries, invoke the SF Oracle RAC installsfrac program from the new node.
- When prompted to enter the system names, specify just the name of the new system.

c Follow the instructions in chapter on installing Oracle9i that describes adding pathes and to apply Oracle 9i patches to the new system. Key points to note:

- To install Oracle binaries, invoke the SF Oracle RAC installsfrac program from the new node.
- When prompted to enter the system names, specify just the name of the new system.

d Edit the listener.ora file on the new system to specify the IP address (or the virtual IP address) for the new system. For example, where the new system's virtual IP address is 192.2.40.23, edit the section for LISTENER so that it resembles:

```
LISTENER =
 (DESCRIPTION_LIST =
  (DESCRIPTION =
   (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 192.2.40.23)(PORT
       =1521)
    )
   )
  )
 )
```

Starting VCS on the new system

1 To start VCS on the new system:

```
# hastart
```

2 Run the following command to verify that the group, "cvm" is online on each node, including the new node:

```
# hastatus -sum
-- SYSTEM STATE
-- System State Frozen
A galaxy RUNNING 0
A nebula RUNNING 0
A saturn RUNNING 0
-- GROUP STATE
```

```
-- Group System Probed AutoDisabled State
B cvm galaxy Y N ONLINE
B cvm nebula Y N ONLINE
B cvm saturn Y N ONLINE
```

3 On one of the existing nodes, run the following command to ensure the new node is recognized:

```
# /etc/vx/bin/vxclustadm nidmap
Name CVM Nid CM Nid State
galaxy 0 0 Joined: Slave
nebula 1 1 Joined: Master
saturn 2 2 Joined: Slave
```

Starting GSD on the new system

Whether you installed Oracle9i locally or on shared storage, you must start the global services daemon (gsd) on the new system. To start gsd, log in as “oracle” user on the new system and enter:

```
$ $ORACLE_HOME/bin/gsdctl start
```

Configuring a new Oracle9i instance

1 On an existing system, add a new instance. Refer to the *Oracle9i Installation Guide*. Highlights of the steps to add a new instance include:

- Log in as the user `oracle` and connect to the instance.
- Create a new “undotbs” tablespace for the new instance. For example, if the tablespace is for the third instance, name it “undotbs3.” If the database uses raw volumes, create the volume first. Use the same size used by the existing “undotbs” volumes.
- Create two new “redo” log groups for the new instance. For example, if the tablespace is for the third instance, create the tablespaces “redo3_1” and “redo3_2.” If the database uses raw volumes, create the volume for the redo logs first. Use the size used by the existing redo volumes.
- Enable “thread 3” where 3 can be the number of the new instance.
- Prepare the `init{SID}.ora` file for the new instance on the new node.
- If the Oracle is installed locally on the new system, prepare the directories `bdump`, `cdump`, `udump`, and `pfile`.

2 If you use in-depth monitoring for the database, create the table for the database instance. Create the table on the new system. Refer to the *Veritas High Availability Agent for Oracle Installation and Configuration Guide* for instructions on creating the table.

3 On the new node, create a mount point directory for the database file system. For example:

```
# mkdir /oradata
```

4 Mount the file system with the database on the new system:

```
# mount -V vxfs -o cluster /dev/vx/dsk/oradatadg/oradatavol \
/oradata
```

5 Log in as Oracle user and attempt to manually start the new instance; the following example is for a third system:

```
$ export ORACLE_SID=vrts3
$ sqlplus '/as sysdba'
sqlplus> startup pfile=/app/oracle/orahome/dbs/initvrts3.ora
```

6 After you bring up the new Oracle instance on the new system, place the instance under VCS control.

a Add the new system to the `SystemList`. For example, where the existing nodes, `galaxy` and `nebula` are nodes 0 and 1, `saturn`, the new node, would be node 2:

```
# haconf -makerw
# hagrp -modify oradb1_grp SystemList -add saturn 2
```

b Add the new system to the `AutoStartList` for `oradb1_grp`:

```
# hagrp -modify oradb1_grp AutoStartList galaxy nebula \
    saturn
```

c Modify the `Sid` (system ID) and `Pfile` (parameter file location) attributes of the the Oracle resource. For example:

```
# hares -modify ora1 Sid vrts3 -sys saturn
# hares -modify ora1 Pfile \
   Pfile=/app/oracle/orahome/dbs/initvrts3.ora -sys saturn
```

d If you have created a table for in-depth monitoring, modify the `Table` attribute of the Oracle resource. For example:

```
# hares -modify ora1 Table vcstable_saturn -sys saturn
```

e Close and save the configuration:

```
# haconf -dump -makero
```

7 As an optional step, at this point verify the configuration by entering the following command on the new system:

```
# hastop -local
```

All resources should come offline on the new system.

8 If you performed step 7, verify that all resources come online after starting VCS on the new system:

```
# hastart
```

Adding the new system to the SFDB repository

Add the new system to the Storage Foundation database repository using the following procedure.

Add the new system to the SFDB repository

1. Add the system using the following sequence of commands:

```
# haconf -makerw
# hagrp -modify Sfua_Base SystemList -add saturn
# hares -modify sfua_ip Device en3 -sys saturn
# haconf -dump -makero
```

2. Copy the /etc/vx/vxdbed/ .odbc.ini file from and existing node to the new system using a remote file copy utility such as rcp, tcp, or scp. For example, to use rcp, enter:

```
# rcp /etc/vx/vxdbed/.odbc.ini saturn:/etc/vx/vxdbed
```

Removing a node from an Oracle9i SF Oracle RAC Cluster

The procedures in this section describe removing a node from a cluster running SF Oracle RAC and Oracle9i.

To remove a node from an Oracle9i cluster, use the following steps:

- Stop processes before removing the software.
- Uninstall Oracle (if you choose).
- Stop VCS in the cluster and modify the VCS configuration.
- Modify the GAB and LLT configuration files.
- Run the `uninstallsfrac` utility.
- Remove unnecessary files.

Stopping processes before removing software

Before removing Oracle9i from the SF Oracle RAC configuration, stop the `gsd` process and stop any applications using cluster file systems.

Stopping gsd

Stop the `gsd` processes on each node you plan to remove. As user `oracle`, enter the command:

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

Stopping applications using CFS that are not VCS controlled

Stop all applications using the CFS mounts *not* under VCS control.

1. To check if any processes use a CFS mount point, enter:

   ```
   # fuser -c mount_point
   ```

2. Stop any processes that use a CFS mount point or use CVM volumes.

Unmounting CFS file systems that are not VCS controlled

On the systems you are removing, unmount any CFS file systems *not* under VCS control.

1. Determine file systems to be unmounted. For example, enter:

   ```
   # mount | grep vxfs | grep cluster
   ```

2. Unmount each of the file systems listed in the output by specifying its mount point:

   ```
   # umount mount_point
   ```

Removing the Oracle Instance on the Node

Use the following procedures to remove the Oracle instance from the leaving node.

Disabling the Oracle instance

Use `dbca` to delete the Oracle instance, or use SQL statements to drop the log groups and disable the public thread.

Uninstalling Oracle (optional) if it is installed locally

Uninstalling Oracle from the node your are removing is optional.

- If you choose not to remove Oracle, and if Oracle is installed on a cluster file system, go to "Stopping VCS and modifying the configuration" on page 186.
- If you choose not to remove Oracle, and if Oracle is installed on your local system, go to "Unlinking Veritas libraries from Oracle binary (optional)."

If Oracle is installed locally and you choose to uninstall it, use the Oracle `runInstaller` utility. Run the utility on each system you are removing from a RAC cluster.

1. On one system, log in as `oracle`.
2. Set the DISPLAY variable.
   If you use the Bourne Shell (`sh` or `ksh`):
   ```
   $ DISPLAY=host:0.0;export DISPLAY
   ```
   If you use the C Shell (`csh` or `tcsh`):
   ```
   $ setenv DISPLAY host:0.0
   ```
3. Run the Oracle utility `runInstaller`:
   ```
   $ cd $ORACLE_BASE/oui/install
   $ ./runInstaller
   ```
   As the utility starts up, be sure to select the option to uninstall the Oracle software from the local node you are removing. Refer to the *Oracle Installation Guide* for additional information about running the utility.

Unlinking Veritas libraries from Oracle binary (optional)

Use this procedure only if you are *not* uninstalling Oracle from the node you are removing.

Note: If Oracle is on cluster file system, or if you have uninstalled Oracle, skip to "Stopping VCS and modifying the configuration" on page 186.

Remove the links to the Veritas libraries and convert to Oracle single-instance binary. Use the following procedure on each system you are removing from the RAC cluster:

1. Log in as the user `oracle`.
2. Using `vi` or another text editor, edit the `init$ORACLE_SID.ora` file:

   ```
   $ cd $ORACLE_HOME/dbs/init$ORACLE_SID.ora
   $ vi init$ORACLE_SID.ora
   ```

   Set the `cluster_database` parameter to FALSE:

   ```
   cluster_database=FALSE
   ```

3. Change directories:

   ```
   $ cd $ORACLE_HOME/lib
   ```

4. Remove the file linked to the Veritas ODM library:

   ```
   $ rm libodm9.so
   ```

5. Copy file `$ORACLE_HOME/lib/libodmd9.so` to remove the ODM function:

   ```
   $ cp $ORACLE_HOME/lib/libodmd9.so $ORACLE_HOME/lib/libodm9.so
   ```

6. To convert from the Oracle RAC binary to the Oracle single-instance binary, enter:

   ```
   $ export LDR_CNTRL=MAXDATA=0x9000000
   $ cd $ORACLE_HOME/rdbms/lib
   $ make -f ins_rdbms.mk rac_off
   $ make -f ins_rdbms.mk ioracle
   ```

7. Verify that Oracle single-instance is configured. Use the command:

   ```
   $ dump -x64 -Hv $ORACLE_HOME/bin/oracle
   ```

   If Oracle RAC is configured, the list of libraries in the output contains:

   ```
    ..........
   libskgxp9.a
   libskgxn9.a
   .......
   ```

   These libraries are not listed for an Oracle single instance configuration. If there is no output, Oracle single-instance binary is configured.

Stopping VCS and modifying the configuration

Stopping VCS takes all service groups on the system offline and allows reconfiguration.

1. Log in as root user on the cluster system you are removing.
2. Stop VCS on the node:

   ```
   # hastop -local
   ```

Modifying the VCS configuration to remove system

You can modify the VCS configuration using one of three possible methods. You can edit `/etc/VRTSvcs/conf/config/main.cf` (the VCS configuration file) directly, you can use the VCS GUI (Cluster Manager), or you can use the command line, as illustrated in the following example. Please refer to the *Veritas Cluster Server User's Guide* for details about how to configure VCS.

On one of the existing nodes, as root user, run the following commands:

1. Make the configuration writable:

   # **haconf -makerw**

2. Remove the system from the `AutoStartList` attribute of the ClusterService group by specifying the remaining nodes in the desired order:

   # **hagrp -modify ClusterService AutoStartList galaxy nebula**

3. If you have the ClusterService service group configured, remove the system (for example, `saturn`) from the system list:

   # **hagrp -modify ClusterService SystemList -delete saturn**

4. Remove the system from the `AutoStartList` of the Oracle service group `oradb1_grp` by specifying the remaining nodes in the desired order:

   # **hagrp -modify oradb1_grp AutoStartList galaxy nebula**

5. Remove the system from the `SystemList` attribute of the `oradb1_grp` service group:

   # **hagrp -modify oradb1_grp SystemList -delete saturn**

6. If you have other Oracle service groups with Oracle resources that have the node to be removed in their configuration, perform step 2 and step 3 for each of them.

7. Delete the system from the CVMCluster resource in the CVM group by removing it from the `CVMNodeId` attribute keylist:

   # **hares -modify cvm_clus CVMNodeId -delete saturn**

8. Delete the system from the CVM group `AutoStartList` attribute by specifying only the remaining nodes in the desired order:

   # **hagrp -modify cvm AutoStartList galaxy nebula**

9. Delete the system from the CVM service group `SystemList` attribute:

   # **hagrp -modify cvm SystemList -delete saturn**

10. After deleting the node to be removed from all service groups in the configuration, delete the removed node from the cluster system list:

    # **hasys -delete saturn**

11. Save the new configuration to disk:

    # **haconf -dump -makero**

Running vxclustadm to reconfigure CVM membership

On each node that remains in the cluster, run the following command to enable GAB to recognize the new membership:

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

Editing /etc/llthosts and /etc/gabtab files on existing nodes

Before running the `uninstallsfrac`, modify the configuration files on the remaining nodes to remove references to the removed node.

Editing /etc/llthost

On the each of the existing nodes, using `vi` or another editor, edit the file `/etc/llthosts`. Remove lines corresponding to the removed nodes. For example, if `saturn` is the node being removed from the cluster, remove the line "2 saturn" from the file:

```
0 galaxy
1 nebula
2 saturn
```

It should now resemble:

```
0 galaxy
1 nebula
```

Editing /etc/gabtab

In the file `/etc/gabtab`, change the command contained in the file to reflect the number of systems after the node is removed:

```
/sbin/gabconfig -c -nN
```

where *N* is the number of nodes remaining. For example, with two nodes remaining, the file resembles:

```
/sbin/gabconfig -c -n2
```

Running the uninstallsfrac utility

Run the `uninstallsfrac` script on the node where you are uninstalling SF Oracle RAC. For this example, `saturn` is the node.

1. As root user, enter:

   ```
   # cd /opt/VRTS/install
   # ./uninstallsfrac
   ```

2. The uninstaller checks for Storage Foundation for Oracle RAC `filesets` currently installed on your system. It also checks for dependencies between `filesets` to determine which `filesets` it can safely uninstall.

3. Proceed with uninstallation when the uninstaller is ready:

   ```
   uninstallsfrac is now ready to uninstall SFRAC filesets.
   ```

```
 All SFRAC processes that are currently running will be stopped.

Are you sure you want to uninstall SFRAC filesets? [y,n,q] (y)
```

4 When you press enter to proceed, the uninstaller stops processes and drivers running on `saturn`, and reports its activities:

```
  Stopping SFRAC processes on saturn:
    Checking vxsvc process ..............................running
    Stopping vxsvc ....................................... Done
    Checking vxfsckd process ....................... not running
    Checking odm driver .......................odm module loaded
    Stopping odm driver ................................... Done
    .
    .
  Stopping SFRAC processes on saturn:
```

5 When the installer begins removing `filesets` from the system, it indicates its progress by listing each step of the total number of steps required.

6 When the uninstaller is done, it displays the location of the summary file and the log for uninstallation activities. Refer to the logs to make sure that all filesets are uninstalled.

Chapter 9

Uninstalling SF Oracle RAC from Oracle9i systems

At the completion of the uninstallation activity, you can continue to run Oracle using the single-instance binary generated when you unlink the Veritas binaries from Oracle.

Uninstalling SF Oracle RAC on Oracle9i:

Stopping gsd and applications using Oracle9i on CFS

If you are using Oracle9i, stop any `gsd` processes running on all nodes in the SF Oracle RAC cluster.

To stop gsd

1. Enter the following command as user `oracle` to determine if `gsd` processes are running:

   $ **$ORACLE_HOME/bin/gsdctl stat**

2. Use the following command to stop the processes:

   $ **$ORACLE_HOME/bin/gsdctl stop**

Stopping applications using CFS (outside of VCS control)

All Oracle users must stop all applications using the CFS mounts *not* under VCS control.

To verify that no processes are using the CFS mount point

1. Enter:

   # **fuser -c *mount_point***

2. Stop any processes using a CFS mount point.

Unmounting CFS file systems (outside of VCS control)

All Oracle users must unmount any CFS file systems *not* under VCS control on all nodes.

To unmount CFS file systems not under VCS control

1. Determine the file systems to unmounted. For example, enter:

   # **mount | grep vxfs | grep cluster**

   The output lists all vxfs file systems mounted in the cluster mode.

2. By specifying its mount point, unmount each file system listed in the output:

   # **umount *mount_point***

Offlining the Oracle and Netlsnr resources

Offline the Oracle and Netlsnr resources on each node. For example:

```
# hares -offline ora1 -sys galaxy
# hares -offline ora1 -sys nebula
# hares -offline listener -sys galaxy
# hares -offline listener -sys nebula
```

These commands stop the Oracle instances and listeners running on the specified systems.

Removing the Oracle database (optional)

You can remove the Oracle database after safely relocating the data as necessary.

Uninstalling Oracle9i (optional)

To uninstall Oracle9i, use the `runInstaller` utility. You must run the utility on each system if Oracle9i is installed locally. If Oracle9i is installed on a cluster file system, you need only run the utility once.

1. On one system, log in as `oracle`.
2. Set the DISPLAY variable.
   Example: for the Bourne Shell (`bash`, `sh` or `ksh`):
   ```
   $ export DISPLAY=host:0.0
   ```
   Example: for the C Shell (`csh`):
   ```
   $ setenv DISPLAY host:0.0
   ```
3. Run the Oracle9i utility `runInstaller`.
   ```
   $ cd $ORACLE_BASE/oui/install
   $ runInstaller
   ```
   As the utility starts up, be sure to select the option to uninstall the Oracle9i software. Refer to the *Oracle9i Installation Guide* for addition information about running the utility.
4. If necessary, remove Oracle9i from the other systems.

Unlinking Veritas libraries from Oracle9i binaries

If you have uninstalled Oracle, skip this procedure. If you have not uninstalled Oracle, unlink the Veritas libraries, using the following procedure, which generates a single-instance Oracle binary.

To unlink Oracle9i binaries from Veritas libraries

1 Log in as the `oracle` user.

2 Using `vi` or another text editor, edit the `init$ORACLE_SID.ora` file.

```
$ cd $ORACLE_HOME/dbs/init$ORACLE_SID.ora
$ vi init$ORACLE_SID.ora
```

Set the `cluster_database` parameter to FALSE.

```
cluster_database=FALSE
```

3 Change to the ORACLE_HOME directory.

```
$ cd $ORACLE_HOME/lib
```

Note: Replace the **XX_XX_XX-XX_XX_XX** in the following examples with the most recent timestamp.

---

4 Restore the original Oracle libraries.

```
$ rm libodm9.so
$ cp libodmd9.so libodm9.so
$ rm libskgxpu.a
$ cp libskgxpu.a.XX_XX_XX-XX_XX_XX libskgxpu.a
$ rm libskgxn9.a
$ cp libskgxn9.a.XX_XX_XX-XX_XX_XX libskgxn9.a
```

5 Relink Oracle.

```
$ cd $ORACLE_HOME/rdbms/lib
$ cp env_rdbms.mk.XX_XX_XX-XX_XX_XX env_rdbms.mk
$ make -f ins_rdbms.mk ioracle
```

Stopping VCS

All Oracle users must stop VCS to take the service groups on all nodes offline.

To stop VCS

1 Log in as root user on one cluster node.

2 Stop VCS on all nodes.

```
# hastop -all
```

Removing SF Oracle RAC packages

The `uninstallsfrac` script removes packages installed by `installsfrac` on all systems in the cluster. The installer removes all the SF Oracle RAC filesets, regardless of the version of Oracle used.

To run uninstallsfrac

1. As root user, navigate to the directory containing the `uninstallsfrac` program.

   ```
   # cd /opt/VRTS/install
   ```

2. Start `uninstallsfrac`:

   ```
   # ./uninstallsfrac
   ```

3. The utility reports the cluster and systems for uninstalling. Enter **y** if the cluster information is correct.

   After entering the systems where the uninstallation will take place, the script checks the operating system on each system, verifies system-to-system communication, and sets up the log files.

   ```
   Using /usr/bin/ssh -x and /usr/bin/scp to communicate with
remote systems.

   Initial system check completed successfully.
   ```

   The script checks for SF Oracle RAC packages currently installed on the nodes. This process involves identifying system uninstall requirements and dependencies between packages to determine the safety and order of uninstalling packages.

4. Confirm to uninstall SF Oracle RAC. packages.

5. Review the output as the script stops processes and drivers running on each node, and reports its activities.

6. Review the output as the script indicates the progress of removing packages from the nodes by listing the steps that are completed. The total number of steps depends on the nature of the installation. For example:

   ```
    Uninstalling Storage Foundation for Oracle RAC 5.0 on all
systems
    simultaneously:

     Uninstalling VRTSormap 5.0 on galaxy ...... Done 1 of 90 steps
     Uninstalling VRTSormap 5.0 on nebula ...... Done 2 of 90 steps
     Uninstalling VRTSdbckp 5.0 on galaxy ...... Done 3 of 90 steps
     Uninstalling VRTSdbckp 5.0 on nebula ...... Done 4 of 90 steps
     Uninstalling VRTScsocw 5.0 on galaxy ...... Done 5 of 90 steps
     .
     .
     Uninstalling VRTSvmman 5.0 on galaxy ..... Done 87 of 90 steps
     Uninstalling VRTSvmman 5.0 on nebula ..... Done 88 of 90 steps
   ```

```
 Uninstalling VRTSvxvm 5.0 on galaxy ...... Done 89 of 90 steps
 Uninstalling VRTSvxvm 5.0 on nebula ...... Done 90 of 90 steps

Storage Foundation for Oracle RAC package uninstall completed
successfully.
```

7 If necessary, review the summary and log files of uninstallation activities. The location of the log files and summary files are displayed.

Removing repository database

To remove the Storage Foundation for Oracle repository database, use the following procedure.

To remove repository database

1 Run the following commands to remove the repository configuration from the VCS configuration and deport the repository diskgroup.

```
# /opt/VRTS/bin/sfua_db_config -o unconfig_cluster
```

2 Import the repository disk group using the command:

```
# vxdg import <name_of_disk_group>
```

3 Mount the repository volume using the command:

```
# /opt/VRTSdbcom/configu/sfua_rep_mount
```

4 Drop the repository database using the command:

```
# /opt/VRTS/bin/sfua_db_config -o dropdb
```

Removing license files

To remove license files

1 To see what license key files you have installed on a system, enter:

```
# /sbin/vxlicrep
```

The output lists the license keys and information about their respective products.

2 Go to the directory containing the license key files and list them. Enter:

```
# cd /etc/vx/licenses/lic
# ls -a
```

3 Using the output from step 1, identify and delete unwanted key files listed in step 2. Unwanted keys may be deleted by removing the license key file.

Removing other configuration files

You can remove the following configuration files:

```
/etc/vcsmmtab
/etc/vxfentab
/etc/vxfendg
/etc/llttab
/etc/gabtab
/etc/llthosts
```

Section IV

Setting up SF Oracle RAC with Oracle 10g

After installing and configuring SF Oracle RAC, use the following procedures to install and configure Oracle 10*g*:

Chapter 10

Preparing to Install Oracle 10g RAC

After setting up SF Oracle RAC, prepare to install Oracle10g. You can install the software on shared storage or locally on each node. Make sure to review the Oracle installation manuals before installing Oracle 10g.

This chapter contains the topics:

About Oracle 10g RAC in an SF Oracle RAC environment

Review the information on infrastructure requirements and Oracle RAC in an SF Oracle RAC environment.

Oracle RAC infrastructure requirements

Oracle RAC requires a cluster infrastructure that deals with these aspects:

- Shared concurrent access to storage
  - ODM support
  - ODM-compliant cluster file system
  - Cluster-volume management
- Cluster membership management
  - Tracking current members
  - Joining systems

  - Leaving systems
- Communications channels between systems
  - Inter-instance messaging
  - Cluster state
  - Cache fusion

Oracle RAC in a Veritas SF Oracle RAC environment

Veritas SF Oracle RAC provides all software infrastructure components for Oracle RAC. Multiple systems running database instances provide access to the same physical database on behalf of multiple clients. Multiple instances accessing the same data provide increased scalability by spreading the load across systems, and they provide increased availability. Multiple instances also increase the need for coordination. Instances must coordinate access to data to ensure one instance does not overwrite or corrupt data. For a view of the overall environment:

See "How SF Oracle RAC works (high-level perspective)" on page 20.

About the location of ORACLE_HOME

Before installing Oracle binaries (ORACLE_HOME) locally on each system or on a cluster file system on shared storage as described in "Installing Oracle 10g RAC" on page 213, consider these points:

- Local installations provide a comfort level using traditional installation methods and the possibility of improved protection against a single point of failure.
- CFS installations provide a single Oracle installation to manage, regardless of number of nodes. This scenario offers a necessary reduction in storage requirements and easy addition of nodes.

Select the location based on your high availability requirements. Symantec and Oracle generally recommend using local installations.

Performing preinstallation operations

Performing preinstallation operations involves manual and automated tasks from the SF Oracle RAC configuration program. Before installing Oracle 10*g* you must perform the following tasks if you have not already performed them:

- Setting Oracle user
- Setting up Oracle user equivalence for RSH and RCP

- Configuring shell limits for Oracle user
- Configuring shell limits for Oracle user
- Configuring kernel parameter for processes per user
- Verifying the user “nobody” exists
- Configuring private IP addresses for CRS
- Creating public virtual IP addresses for use by Oracle
- Creating disk groups, volumes, and mount points
- Identify the directories required for installing Oracle and CRS software:

Using the SF Oracle RAC configuration program

The Oracle installation requires some preparation such as creating the Oracle user and group ID, creating disk groups, volumes, and mount points, and configuring private IP addresses for CRS. You can perform tasks in the configuration program sequentially at one time, or you can access the program for individual tasks later.

To perform preinstallation tasks in the configuration menu

1. Launch the SF Oracle RAC configuration program. As root user on any one system, enter:

   ```
   # cd /opt/VRTS/install
   # installsfrac -configure
   ```

   By default, the installsfrac program uses SSH for remote communication. However, to use RSH, specify the -rsh option with the installsfrac program.

   ```
   # ./installsfrac -rsh -configure
   ```

2. Enter the system names, separated by spaces.
3. From the configuration program menu, select **Prepare to install Oracle**. From this menu, you can choose to perform all installation tasks, or to perform specific tasks.

Setting Oracle user

Configure the Oracle user and group settings in the SF Oracle RAC configuration program.

To set Oracle user

1. Access the SF Oracle RAC configuration program if you are not currently using it.

   See “Using the SF Oracle RAC configuration program” on page 203.

2 Select **Create userid and group id for Oracle** from the configuration menu and provide the required information.

- Refer to Oracle documentation for information on creating the `oinstall` (Oracle Inventory),`dba` and `oper` groups, and the `oracle` user.

3 Enter Oracle UNIX user and primary group information:

```
Enter Oracle UNIX user name: [b] oracle
Enter Oracle UNIX user id (numerical): [b] 1001
Enter Oracle UNIX user home dir: [b] /opt/oracle
Enter Oracle UNIX group name: [b] (oinstall) oinstall
Enter Oracle UNIX group id (numerical): [b] 101
```

Note: The set of Oracle user IDs and group IDs in each cluster configuration must be the same.

4 The installer verifies that the specified `userid` does not exist on any of the systems in the cluster and then creates it. Enter **y** to create the oracle user with the information provided.

5 Enter the information to create secondary groups, "dba" and "oper":

```
Do you want to create secondary groups for Oracle user? [y,n,q]
(y)
Enter Oracle UNIX secondary group name: [b] dba
Enter Oracle UNIX secondary group id (numerical): [b] 102
Group dba does not exist on any node. Do you want to create it
with the information provided [y,n,q] (y)
    Creating group dba on galaxy ... Done
    Adding Oracle user (oracle) to group (dba) on galaxy ... Done
    Creating group dba on nebula ... Done
    Adding Oracle user (oracle) to group (dba) on nebula ... Done
```

6 You must first enable RSH access for the newly created “oracle” user.

7 After creating the secondary groups, the installer proceeds to verify RSH access for "oracle" user. Leave the installer at this prompt and proceed to setup RSH access. You can return to this installer session after setting up oracle user equivalance.
See “Setting up Oracle user equivalence for RSH and RCP” on page 204.

Setting up Oracle user equivalence for RSH and RCP

To set up Oracle user equivalence for RSH and RCP

1 Access the SF Oracle RAC configuration program even if you are currently using it and open another terminal session as `root` user.
See “Using the SF Oracle RAC configuration program” on page 203.

2 As `root` user on each system, edit /etc/hosts.equiv file and add entries similar to the following:

```
galaxy oracle
nebula oracle
```

3 On each system, set the password for the “oracle” user:

```
[root@galaxy /]# passwd oracle
Changing password for "oracle"
oracle's New password:
Re-enter oracle's new password:
```

4 On each system, login as user "oracle" and change the passwd.

```
[root@galaxy /]# su - oracle
$ passwd
Changing password for "oracle"
oracle's New password:
Re-enter oracle's new password:
```

5 On each system, as user “oracle”, verify “rsh” access:

```
$ rsh galaxy date
Mon Apr 24 10:02:45 PDT 2006
$ rsh nebula date
Mon Apr 24 10:02:45 PDT 2006
```

You can now create the secondary groups for Oracle.

Verifying RSH access for Oracle user

To verify RSH access for "oracle" user

1 Return to the installer session the end of “To set Oracle user” on page 203.

2 At the installer prompt, answer "y" to verify "RSH" accessiblity.

3 Quit the installation program.

Configuring shell limits for Oracle user

The values for the shell soft limits must match those shown in the following table:

Shell Limit	Recommended Value
Soft FILE size	-1
Soft CPU time	-1
Soft DATA segment	-1
Soft STACK size	-1

If the values do not match those of the table, change them.

To configure shell limits

1. Use SMIT to verify the current value. For example:

   ```
   # smitty chuser
   ```

2. In the field, enter the user name: `oracle`
3. Scroll down the list. Verify the current values against the table above, editing them if necessary.
4. Press F10 to exit when you are done.
5. Repeat the steps above and verify the values are the same on all cluster nodes.

Configuring kernel parameter for processes per user

Oracle 10g requires that the AIX configurable kernel parameter that specifies the maximum number of processes per user is greater than 2048. The value must be the same on each of the cluster nodes.

To configure maximum processes per user

1. Check the current value by entering the command:

   ```
   # smitty chgsys
   ```

2. Examine the value of the kernel parameter:

   ```
   Maximum number of PROCESSES allowed per user
   ```

   If necessary, change the value to be 2048 or greater.
3. Press F10 to exit when you are done.
4. Use the previous steps to check that the value is the same on each cluster node.

Verifying the user “nobody” exists

Verify the user “nobody” exists on each system in the cluster:

```
# id nobody
uid=4294967294(nobody) gid=4294967294(nobody)
```

Configuring private IP addresses for CRS

The CRS daemon requires a private IP address on each node to enable communications and heartbeating. After confirming the values, the installer adds a new section in the VCS configuration file (`main.cf`) for the PrivNIC resource in the CVM group.

To add private IP addresses to /etc/hosts

1 Log in to each system as root

2 Add the following entries to the /etc/hosts file:

```
192.168.12.1    galaxy_priv
192.168.12.2    nebula_priv
```

To configure private IP addresses for CRS

1 Access the SF Oracle RAC configuration program if you are not currently using it.

See "Using the SF Oracle RAC configuration program" on page 203.

2 Select **Configure private IP addresses for CRS** from the configuration menu.

3 Enter the private IP address information for each host.

```
Enter the private IP for galaxy: [b] 192.168.12.1
    Checking 192.168.12.1 in /etc/hosts on galaxy..... exists
    Discovering NICs on galaxy ........ discovered en0 en1 en2

Enter the NIC 1 for private network for galaxy (x if done): [b]
en1
Enter the NIC 2 for private network for galaxy (x if done): [b]
en2
Enter the NIC 3 for private network for galaxy (x if done): [b] x
Enter the private IP for nebula: [b] 192.168.12.2
    Checking 192.168.12.2 in /etc/hosts on nebul ..... exists
    Discovering NICs on nebula ........ discovered en0 en1 en2
Enter the NIC 1 for private network for nebula (x if done): [b]
en1
Enter the NIC 2 for private network for nebula (x if done): [b]
en2
Enter the NIC 3 for private network for nebula (x if done): [b] x
Enter the netmask for private network: [b] 255.255.240.0
```

Note: The private IP addresses of all nodes should be on the same physical network in the same IP subnet.

4 Confirm the private IP address information.

5 After the private IP address is configured, you can exit the installer by entering **q** or continue with your configuration.

To verify the private NIC address configuration

1 Verify that the PrivNIC resource, ora_priv, shows up in VCS main.cf:

```
PrivNIC ora_priv (
    Critical = 0
     Device @galaxy = { en1 = 0, en2 = 1 }
     Device @nebula = { en1 = 0, en2 = 1 }
     Address @galaxy = "192.168.12.1"
```

```
Address @nebula = 2192.168.12.1"
NetMask = "255.255.240.0"
)
```

2 Verify that the “ora_priv” resourceis online on all systems in the cluster:

```
# hares -state ora_priv
#Resource    Attribute        System     Value
ora_priv       State           galaxy   ONLINE
ora_priv       State            nebula ONLINE
```

3 On each system, check the output of “ifconfig”

```
# ifconfig -a
```

4 From each system, ping the private IP addresses:

```
# ping 192.168.12.2
PING 192.168.12.2 (192.168.12.2): 56 data bytes
64 bytes from 192.168.12.2: icmp_seq=0 ttl=255 time=0 ms
64 bytes from 192.168.12.2: icmp_seq=1 ttl=255 time=0 ms
# ping 192.168.12.1
PING 192.168.12.1 (192.168.12.1): 56 data bytes
64 bytes from 192.168.12.1: icmp_seq=0 ttl=255 time=0 ms
64 bytes from 192.168.12.1: icmp_seq=1 ttl=255 time=0 ms
```

Creating public virtual IP addresses for use by Oracle

Oracle requires one virtual public network address for the Oracle listener process on each node.

1 On each node, select a public NIC device and configure a virtual IP address for it. For example:

```
# ifconfig en0:1 plumb
# ifconfig en0:1 inet 110.10.11.1 netmask 255.255.255.0
# ifconfig en0:1 up
```

Configure one virtual public NIC on each node.

Note: Add the virtual IP addresses to DNS.

Creating disk groups, volumes, and mount points

To create disk groups, volumes, and mount points for Oracle, review these guidelines. Before you install the Oracle Cluster Ready Services (CRS) and Oracle 10g binaries, you must create storage space for these installations. You need to provide storage for:

- The home directories, CRS_HOME and ORACLE_HOME, for CRS and Oracle binaries. See “Identify the directories required for installing Oracle and CRS software:” on page 209.

- The CRS files for Oracle Cluster Registry (OCR) and the VOTE-disk. The files can reside in volumes on raw device, or in directories in a cluster file system. See "Creating OCR and VOTE-disk volumes" on page 210.

Note: The displayed task to create CVM volumes or a directory on CFS for database file storage is covered later.

Identifying required directories for CRS and Oracle

Identify the directories required for installing Oracle and CRS software:

Directory	Component
/app	Mountpoint for Oracle/CRS binaries
/app/crshome	CRS_HOME
/app/oracle	ORACLE_BASE
/app/oracle/home	ORACLE_HOME
/ocrvote	Mountpoint for OCR/Vote
/oradata	Mountpoint for Oracle database

Preparing $CRS_HOME and $ORACLE_HOME on each node

To create a file system on local storage for Oracle/CRS binaries (/app)

1. As root user, first create a VxVM local diskgroup, orabindg_*hostname*:
```
# vxdg init orabindg_galaxy Disk_1
```
2. Create a volume, orabinvol_*hostname*:
```
# vxassist -g orabindg_galaxy make orabinvol_galaxy 12G
```
3. Create directory, /app
```
# mkdir /app
```
4. Create a filesystem with this volume, orabinvol_*hostname*
```
# mkfs -V vxfs  /dev/vx/dsk/orabindg/orabinvol
```
5. Mount /app
```
# mount -V vxfs /dev/vx/dsk/orabindg/orabinvol /app
```
6. Add an entry for this filesystem in /etc/filesystems.
```
/app:
    dev = dev/vx/dsk/orabindg/orabinvol
    vfs = vxfs
```

```
mount = true
options = rw
account = false
```

To prepare $CRS_HOME on each node

1. On each system, log in as "root".
2. Create the directory for CRS_HOME:
   ```
   #mkdir -p /app/crshome
   ```
3. Change ownership and permissions
   ```
   #chown -R oracle:oinstall /app/crshome
   #chmod -R 775 /app/crshome
   ```

To prepare $ORACLE_BASE on each node

1. On each system, log in as "root".
2. Create the directory for ORACLE_BASE:
   ```
   #mkdir -p /app/oracle
   ```
3. Change ownership and permissions
   ```
   #chown -R oracle:oinstall /app/oracle
   #chmod -R 775 /app/oracle
   ```

To prepare $ORACLE_HOME on each node

1. On each system, log in as "root".
2. Create the directory for ORACLE_HOME:
   ```
   #mkdir -p /app/oracle/orahome
   ```
3. Change ownership and permissions
   ```
   #chown -R oracle:oinstall /app/oracle/orahome
   #chmod -R 775 /app/oracle/orahome
   ```

Creating OCR and VOTE-disk volumes

The installation of CRS requires predefined locations for the OCR and VOTE-disk components. Whether you create volumes or file system directories, you can add them to the VCS configuration to make them highly available.

To create a filesystem for OCR and VOTE disks (/ocrvote)

1. As root user, from the CVM master, create a shared VxVM diskgroup
   ```
   # vxdg -s init ocrvotedg Disk_2
   ```
2. As root user, from the CVM master, create a volume, ocrvotevol:
   ```
   # vxassist -g ocrvotedg make ocrvotevol 1G
   ```
3. As root user, from CVM master, create a filesystem with the volume, ocrvotevol.

```
# mkfs -V vxfs /dev/vx/dsk/ocrvotedg/ocrvotevol
```

4 On each system, create a directory, /ocrvote:

```
# mkdir /ocrvote
```

5 On each system, mount /ocrvote

```
# mount -V vxfs -o cluster /dev/vx/dsk/ocrvotedg/ocrvotevol
/ocrvote
```

6 As root user, from any system, change permissions on /ocrvote

```
# chown -R oracle:oinstall /ocrvote
```

7 Add the CFSMount and CVMVolDg resources corresponding to this "/ocrvote" filesystem to the "cvm" group in VCS configuration:

```
#haconf -makerw

#hares -add ocrvote_mnt CFSMount cvm
#hares -modify ocrvote_mnt Critical 0
#hares -modify ocrvote_mnt MountPoint "/ocrvote"
#hares -modify ocrvote_mnt BlockDevice  \
"/dev/vx/dsk/ocrvotedg/ocrvotevol"

#hares -add ocrvote_voldg CVMVolDg cvm
#hares -modify ocrvote_voldg CVMDiskGroup ocrvotedg
#hares -modify ocrvote_voldg CVMVolume -add ocrvotevol
#hares -modify ocrvote_voldg CVMActivation sw

#hares -link ocrvote_mnt ocrvote_voldg
#hares -link ocrvote_mnt vxfsckd
#hares -link ocrvote_voldg cvm_clus

#hares -modify ocrvote_voldg Enabled 1
#hares -modify ocrvote_mnt Enabled 1

#haconf -dump -makero
```

8 Verify the stanzas corresponding to the ocrvote_mnt and ocrvote_voldg resources in main.cf:

```
      CFSMount ocrvote_mnt (
        Critical = 0
                MountPoint = "/ocrvote"
                BlockDevice = "/dev/vx/dsk/ocrvotedg/ocrvotevol"
                )

CVMVolDg ocrvote_voldg (
    Critical = 0
    CVMDiskGroup = ocrvotedg
    CVMVolume = { ocrvotevol }
    CVMActivation = sw
)
```

```
ocrvote_mnt requires ocrvote_voldg
   ocrvote_mnt requires vxfsckd
ocrvote_voldg requires cvm_clus
```

9 Verify that the VCS resources, ocrvote_mnt and ocrvote_voldg, are ONLINE on all systems in the cluster.

```
#hares -state ocrvote_mnt
#hares -state ocrvote_voldg
```

Setting umask before installation

Use the following command to set the default file creation mode on each system where you plan to install Oracle.

```
# umask 0022
```

Chapter 11

Installing Oracle 10g RAC

After installing SF Oracle RAC and preparing to install Oracle 10g, proceed to install theOracle 10g software. Installing Oracle10g R1 or R2 in an SF Oracle RAC environment involves these tasks:

✔ "Installing CRS" on page 213

✔ "Installing Oracle 10g database software" on page 217

✔ "Completing post-installation operations" on page 221

Installing CRS

The CRS software is installed on each node.

Note: When installing CRS for Oracle 10.1.0.2 (10gR1 base installation), you must copy the CRS installation image to a directory that is writable by the root user.

To install CRS

1. Launch the installsfrac program. As root user on any one node in the cluster, enter:

   ```
   #cd /opt/VRTS/install
   #./installsfrac -configure
   ```

   By default, the installsfrac utility uses ssh for remote communication. However, rsh can be used in place of ssh by using the "rsh" option with the installsfrac utility.

   ```
   #cd /opt/VRTS/install
   #./installsfrac -rsh -configure
   ```

2. On the same node where you have set the environment variables, execute the following command as root:

   ```
   # cd /opt/VRTS/install
   ```

```
# ./installsfrac -configure
```

The installer will display the copyright message.

3 Enter the system names on which you want to install Oracle 10g software and navigate to the "Install and Relink Oracle" menu.

4 In the Choose Oracle version menu, select the appropriate version of Oracle (10gR1 or 10gR2).

5 In the Choose task menu, select the task Install Oracle Clusterware (CRS).

6 In the Set DISPLAY dialog, enter the value for the DISPLAY environment variable. This value should refer to the address of the X11 server where you want OUI for CRS installation displayed..

7 Enter Oracle user and group information.

- In the Oracle Username dialog, enter Oracle Unix User Account when prompted. The installer checks for the user on all systems.
- In the Oracle Groupname dialog, enter Oracle Inventory group when prompted. The installer checks for group existence on all systems.
- Press Enter to continue.

8 In the CRS Install Image dialog, enter the absolute path of CRS install image when prompted. The installer validates the CRS installer. Press Enter to continue.

9 In the Oracle base directory dialog, enter the location of base of the Oracle directory structure for OFA compliant databases. The installer validates the Oracle Base Directory. If the directory doesn't exist , installer prompts for the creation of oracle base directory on all nodes. Choose option 'y' to create oracle base directory on all nodes.

10 Press Enter to continue.

11 In the CRS Home directory dialog, enter absolute path of CRS home directory when prompted. Installer validates the CRS home directory. If the directory doesn't exist , installer prompts for the creation of the directory on all nodes. Choose option 'y' to create oracle base directory on all nodes.

12 Press Enter to continue.

13 In the Patch Level dialog, specify the patch level of Oracle CRS/Database Software being installed or relinked. Accept the default when installing base version of Oracle 10gR1 (10.1.0.2) or 10gR2 (10.2.0.1)

14 The Installer prints the CRS installation information for verification. If the information displayed by the installer is correct choose option "y" otherwise choose option "n".

Example:

```
Oracle environment information verification
Oracle Unix User: oracle
Oracle Unix Group: oinstall
Oracle Clusterware (CRS) Installation Path:
/orcl/10gR2/CRS/Disk1
Oracle Clusterware (CRS) Home: /app/oracle/crshome
Oracle Release: 10.2
Oracle Patch Level: 0.1
Oracle Base: /app/oracle
Is this information correct? [y,n,q] (y)
```

15 Press Enter to continue.

16 The installer now validates the inputs and performs certain setup steps, after which it prompts you to run rootpre.sh on all cluster nodes.

```
    Verifying  fileset prerequisites on galaxy ............ ok
    Verifying fileset prerequisites on nebula ............. ok
    Comparing Oracle UNIX user id on all systems ........ Done
    Comparing Oracle group id on all systems ............ Done
    Creating SFRAC odm entries on galaxy .................. ok
    Creating SFRAC odm entries on nebula .................. ok

For installation to proceed, Oracle requires you to run the
rootpre.sh script located at:
/orcl/10gR2/CRS/Disk1/rootpre

Run the following commands on all nodes:
# cd /orcl/10gR2/CRS/Disk1/rootpre
# ./rootpre.sh

Has rootpre.sh been run by root? [y,n,q]
```

Note: Do not respond to the prompt yet, but proceed to step 14.

17 On each system, log in as root in another terminal session.

- On each system, change the directory to the location of the rootpre.sh script, and invoke the script.
- After invoking rootpre.sh on all nodes, return to the installsfrac prompt (from step 16) and enter “y” to indicate that the script has been invoked on all systems in the cluster.

18 Installer now brings up the OUI for Oracle CRS Software Installation.

```
    Invoking OUI to install Oracle CRS. Please wait .. ok

    Now install Oracle CRS software from the OUI. Refer to
    Oracle documentation for installation instructions.

    NOTE: Near the end of the installation, Oracle CRS
    Software installer will prompt you to run the root.sh
```

```
script.

Press <RETURN> here when prompted to run root.sh
```

Note: Do not press Return yet, but wait for the prompt to do so.

To install Oracle 10.2.0.1 CRS using the Oracle interface

1. If the “Specify Inventory Directory and Credentials” window is presented, enter the location of Oracle inventory and the AIX group name.
2. When the Oracle Universal Installer (OUI) for Oracle CRS Installation displays, navigate to the “Specify Home Details” window. Change OUIHome1 to CRS_HOME and specify the path for $CRS_HOME.
3. Make sure that all systems in the cluster appear on the "Specify Cluster Configuration" screen. Modify the private name and virtual name as required.
4. When the “Specify Network Interface Usage” window presents a list of interface devices for which IP addresses are configured, verify that the public interface and private interface match those you have configured for Oracle 10g. If devices are shown on the list that are not part of the configuration for Oracle 10g, be sure they are labelled with "Do Not Use".
5. On the "Oracle Cluster Registry" window:
   - Choose External Redundancy
   - Set “OCR Location” to “/ocrvote/ocr”
6. In the "Voting Disk" window:
   - Choose External Redundancy
   - Set “Voting Disk Location” to “/ocrvote/vote”
7. Confirm the settings on the final screen. Make sure that /tmp has ample space. Select “Install” to start the CRS installation.
8. When Oracle CRS binaries have been installed, the OUI prompts you to invoke two scripts, namely, orainstRoot.sh and root.sh.

   Note: Do not run root.sh yet, but return to the installsfrac session from step 18 in “To install CRS.”

9. In the installsfrac session, press "Return". The installsfrac utility now verifies the CRS installation and copies the SF Oracle RAC libraries to $CRS_HOME.

10 Next, login as “root” on each system and invoke the two scripts, orainstRoot.sh and root.sh. Note that you must NOT run these scripts simultaneously on all systems.

11 Return to the OUI and indicate that you've run root.sh on all systems in the cluster.

12 Exit the OUI.

13 If the installation and setup is successful, then root.sh should be able to start the CRS daemons on each node. To verify if CRS is up and running, enter:

```
#/app/crshome/bin/crs_stat -t
#No resources configured.
```

14 If the crs_stat command fails with message “failed to contact crs daemons”, it means the CRS installation was unsuccessful. For troubleshooting:
See “Troubleshooting SF Oracle RAC” on page 405.

15 Oracle CRS installation is now complete. Add a resource to the VCS configuration to monitor the CSSD daemon:
See “Configuring the Application agent to monitor CSSD” on page 462.

Installing Oracle 10g database software

After installing the CRS component, install the Oracle database software.

Note: When installing CRS for Oracle 10.1.0.2 (10gR1 base installation), you must copy the CRS installation image to a directory that is writable by the root user.

To install Oracle Database Software

1 Launch the installsfrac program. As root user on any one node in the cluster, enter:

```
#cd /opt/VRTS/install
#./installsfrac -configure
```

By default, the installsfrac utility uses ssh for remote communication. However, rsh can be used in place of ssh by using the "rsh" option with the installsfrac utility.

```
#cd /opt/VRTS/install
#./installsfrac -rsh -configure
```

2 Navigate to the “Main Menu” and select “Install and Relink Oracle”.

3 In the Choose Oracle version menu, select the appropriate version of Oracle (10gR1 or 10gR2).

4 In the Choose task menu, select “Install Oracle RDBMS server.”

5 In the Set DISPLAY dialog, enter the value for the DISPLAY environment variable. This value should refer to the address of the X11 server where you want OUI for CRS installation displayed..

6 Enter Oracle user and group information.
   - In the Oracle Username dialog, enter Oracle Unix User Account when prompted. The installer checks for the user on all systems.
   - In the Oracle Groupname dialog, enter Oracle Inventory group when prompted. The installer checks for group existence on all systems.
   - Press Enter to continue.

7 In the Database Install Image dialog, specify the location of Oracle Database Software install image.

8 In the Oracle base directory dialog, specify the base of the Oracle directory structure for OFA compliant databases.

9 In the CRS Home directory dialog, specify the directory containing Oracle CRS Software.

10 In the Database home directory dialog, specify the directory containing Oracle Database Software.

11 In the Patch Level dialog, specify the patch level of Oracle CRS/Database Software being installed or relinked. Accept the default when installing base version of Oracle 10gR1 (10.1.0.2) or 10gR2 (10.2.0.1)

12 The installer prints the oracle environment information for verification. If the information displayed by the installer is correct choose option “y” otherwise choose option “n.”
   Example:
   ```
   Oracle environment information verification
   Oracle Unix User: oracle
   Oracle Unix Group: oinstall
   Oracle Clusterware (CRS) Home: /app/crshome
   Oracle Release: 10.2
   Oracle Patch Level: 0.1
   Oracle Base: /app/oracle
   Oracle Home: /app/oracle/orahome
   RDBMS Installation Path: /orcl/10gR2/DB/Disk1
   Is this information correct? [y,n,q] (y)
   ```
   Press Enter to continue.

13 The installer now validates the inputs and performs certain setup steps, after which it prompts you to run rootpre.sh on all cluster nodes.
   ```
   Comparing Oracle UNIX user id on all systems ........ Done
   Comparing Oracle group id on all systems ............ Done
   ```

```
For installation to proceed, Oracle requires you to run the
rootpre.sh script located at:
/orcl/10gR2/DB/Disk1/rootpre

Run the following commands on all nodes:
# cd /orcl/10gR2/DB/Disk1/rootpre
# ./rootpre.sh

Has rootpre.sh been run by root? [y,n,q]
```

Note: Do not respond to the prompt yet, but proceed to step 14.

14 On each system, log in as root in another terminal session.

- On each system, change the directory to the location of the rootpre.sh script, and invoke the script.
- After invoking rootpre.sh on all nodes, return to the installsfrac prompt (from step 16) and enter "y" to indicate that the script has been invoked on all systems in the cluster.

15 Installer now brings up the OUI for Oracle RDBMS Software Installation.

```
Invoking OUI to install Oracle Software. Please wait .. ok

Now install Oracle database software from the OUI. Refer to
Oracle documentation for installation instructions.

NOTE: Near the end of the installation, Oracle Database
Software installer will prompt you to run the root.sh
script.

Press <RETURN> here when prompted to run root.sh
```

Note: Do not press Return yet, but wait for the prompt to do so.

16 When the OUI for Oracle Database Software installation displays, follow the instructions and respond to the prompts to install Oracle 10g software. When you are prompted for the names of the systems for installation, select all nodes. In the "Select Database Configuration" window, choose the option, "Do not create a starter database."

17 Confirm the settings on the final OUI window. Make sure that /tmp has ample space. Select "Install" to start the Oracle database software installation.

18 When the Oracle database binaries have been installed, the OUI prompts you to invoke the root.sh script.

Note: Do not run root.sh yet, but return to the installsfrac session from step 15.

19 In the installsfrac session, press Return. The installsfrac utility now verifies the database software installation, copies the SF Oracle RAC libraries to $ORACLE_HOME, and relinks Oracle on each node in the cluster.

```
Verifying binaries in /app/oracle/orahome on galaxy ...ok
Verifying binaries in /app/oracle/orahome on nebula ...ok
Copying SFRAC libskgxn on galaxy .......................ok
Copying SFRAC libskgxn on nebula .......................ok
Copying SFRAC ODM library on galaxy ....................ok
Copying SFRAC ODM library on nebula ....................ok
Copying SFRAC libskgxp on galaxy .......................ok
Copying SFRAC libskgxp on nebula .......................ok
Relinking Oracle on galaxy .............................ok
Relinking Oracle on nebula .............................ok
Oracle RDBMS Software Installation is now complete.
```

20 Enter "q" to quit the SF Oracle RAC installer, installsfrac.

21 As root user on each system, invoke the root.sh script:

```
# cd $ORACLE_HOME
# ./root.sh
```

22 Return to the OUI and indicate you have run root.sh.

23 Exit the OUI.

24 After completing installation of Oracle 10g software, you must install the appropriate patches. See “Adding Oracle 10g R1 or R2 patches” on page 221.

Verifying the Oracle CRS and Oracle 10g Installation

To verify that the installations of the Oracle CRS and Oracle 10g have succeeded, issue the command described below from any node in the cluster. The output should show processes running on all nodes, as in the following example:

```
# $CRS_HOME/bin/crs_stat
        NAME=ora.galaxy.vip
        TYPE=application
        TARGET=ONLINE
        STATE=ONLINE on galaxy

        NAME=ora.galaxy.gsd
        TYPE=application
        TARGET=ONLINE
        STATE=ONLINE on galaxy

        NAME=ora.galaxy.ons
        TYPE=application
        TARGET=ONLINE
```

```
STATE=ONLINE on galaxy

NAME=ora.nebula.vip
TYPE=application
TARGET=ONLINE
STATE=ONLINE on nebula

NAME=ora.nebula.gsd
TYPE=application
TARGET=ONLINE
STATE=ONLINE on nebula

NAME=ora.nebula.ons
TYPE=application
TARGET=ONLINE
STATE=ONLINE on nebula
```

Completing post-installation operations

After installing the Oracle software, complete these operations:

- Adding Oracle 10g R1 or R2 patches
- Relinking the SF Oracle RAC libraries to Oracle
- Creating the Oracle database
- Configuring the Oracle Service Group in a VCS Configuration

Adding Oracle 10g R1 or R2 patches

Use the following procedures to add Oracle 10g R1 or R2 patches to your node if you have installed Oracle, but have not yet configured Oracle in your cluster. To install Oracle 10g patch software, you must have installed Oracle 10g software.

Applying Oracle 10.1.0.5 Patchset

Applying Oracle 10.1.0.5 patchset involves the following tasks:

- Install 10.1.0.5 CRS
- Install 10.1.0.5 Database Software
- Relink Oracle

To install 10.1.0.5 CRS

1. Stop Oracle instances on all systems in the cluster.
2. Stop CRS on all system in the cluster.
3. Log in as root and change permissions for $CRS_HOME.

```
#chown -R oracle:oinstall /app/crshome
```

4 Refer to Oracle 10.1.0.5 Patch note and apply the patch to CRS binaries.

5 Near the end of the CRS patch installation, OUI will prompt you to invoke root10105.sh on all systems in the cluster. Log in as root user on each system in the cluster and run the following commands:

- Change directory to $CRS_HOME/install/patch10105/lib

```
#cd  $CRS_HOME/install/patch10105/lib
```

- Check if the 64-bit libskgxn2.a and libskgxn2.so exist. If they do, copy SF Oracle RAC SKGXN libraries in their place.

```
#ls libskgxn2.a libskgxn2.so
#cp /opt/VRTSvcs/rac/lib64/libskgxn2.a libskgxn2.a
#cp /opt/VRTSvcs/rac/lib64/libvcsmm.so libskgxn2.so
```

- Check if the 64-bit libskgxp10.a exists. If it does, copy SF Oracle RAC SKGXP library in its place.

```
#ls libskgxp10.a
#cp /opt/VRTSvcs/rac/lib/libskgxp10_ver25_64.a libskgxp10.a
```

- Change directory to $CRS_HOME/install/patch10105/lib32

```
#cd $CRS_HOME/install/patch10105/lib32
```

- Check if 32-bit libskgxn2.a and libskgxn2.so exist. If they do, copy SF Oracle RAC SKGXN libraries in their place.

```
#ls libskgxn2.a libskgxn2.so
#cp /opt/VRTSvcs/rac/lib/libskgxn2.a libskgxn2.a
#cp /opt/VRTSvcs/rac/lib/libvcsmm.so libskgxn2.so
```

- Check if 32-bit libskgxp10.a exists. If it does, copy SF Oracle RAC SKGXP library in its place.

```
#ls libskgxp10.a
#cp /opt/VRTSvcs/rac/lib/libskgxp10_ver25_32.a libskgxp10.a
```

6 Now invoke the script, root10105.sh. This script copies everything from under $CRS_HOME/install/patch10105 directory to their corresponding location under $CRS_HOME. It then starts the CRS daemons. You must run this script on each system in the cluster (in the same order as specified by Oracle installer).

```
#cd $CRS_HOME/install
#./root10105.sh
```

7 Verify that CRS daemons are up on systems in the cluster. The command, crs_stat, should list all the CRS resources.

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

To install 10.1.0.5 Oracle database Software

1 Refer to Oracle 10.1.0.5 Patch note and apply the patch to Oracle binaries.

2 Run root.sh script at the end of the patch installation as instructed by Oracle.

3 Relink the Oracle libraries.
See "Relinking the SF Oracle RAC libraries to Oracle" on page 224.

Applying Oracle 10.2.0.2 Patchset

Applying Oracle 10.2.0.2 patchset involves the following tasks:

- Install 10.2.0.2 CRS
- Install 10.2.0.2 Database Software
- Relink Oracle

To install 10.2.0.2 CRS

1 Stop Oracle instances on all systems in the cluster.

2 Stop CRS on all system in the cluster.

3 Log in as root and change permissions for $CRS_HOME.
```
#chown -R oracle:oinstall /app/crshome
```

4 Refer to Oracle 10.2.0.2 Patch note and apply the patch to CRS binaries.

5 Near the end of the CRS patch installation, OUI will prompt you to invoke root102.sh on all systems in the cluster. Log in as root user on each system in the cluster and run the following commands:
   - Change directory to $CRS_HOME/install/patch102/lib
     ```
     #cd  $CRS_HOME/install/patch102/lib
     ```
   - Check if the 64-bit libskgxp10.a and libskgxg.a libraries exist. If they do, copy SF Oracle RAC SKGXP libraries in their place.
     ```
     #ls libskgxp10.a libskgxpg.a
     #cp /opt/VRTSvcs/rac/lib/libskgxp10_ver25_64.a libskgxp10.a
     #cp /opt/VRTSvcs/rac/lib/libskgxp10_ver25_64.a libskgxpg.a
     ```
   - Change directory to $CRS_HOME/install/patch102/lib32
     ```
     #cd $CRS_HOME/install/patch102/lib32
     ```
   - Check if 32-bit libskgxp10.a and libskgxpg.a libraries exists. If they do, copy SF Oracle RAC SKGXP library in their place.
     ```
     #ls libskgxp10.a
     #cp /opt/VRTSvcs/rac/lib/libskgxp10_ver25_32.a libskgxp10.a
     #cp /opt/VRTSvcs/rac/lib/libskgxp10_ver25_32.a libskgxpg.a
     ```

6 Now invoke the script, root102.sh. This script copies everything from under $CRS_HOME/install/patch102 directory to their corresponding location under $CRS_HOME. It then starts the CRS daemons. You must run this script on each system in the cluster (in the same order as specified by Oracle installer).

```
#cd $CRS_HOME/install
#./root102.sh
```

7 Verify that CRS daemons are up on systems in the cluster. The command, crs_stat, should list all the CRS resources.

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

To install 10.2.0.2 Oracle database Software

1 Make sure CRS is running on all the nodes.

2 Refer to Oracle 10.2.0.2 Patch note and apply the patch to Oracle binaries.

3 Run root.sh script at the end of the patch installation as instructed by Oracle.

4 Relink the Oracle libraries.
See "Relinking the SF Oracle RAC libraries to Oracle" on page 224.

Relinking the SF Oracle RAC libraries to Oracle

After installing the Oracle base software or Oracle patches, use the configuration program to relink SF Oracle RAC libraries to Oracle.

To relink the SF Oracle RAC libraries to Oracle

1 Launch the installsfrac program. As root user on any one node in the cluster, enter:

```
# cd /opt/VRTS/install
# installsfrac -configure
```

By default, the installsfrac program uses ssh for remote communication. However, rsh can be used in instead of SSH by using the "-rsh" option with the installsfrac utility.

```
# cd /opt/VRTS/install
# installsfrac -rsh -configure
```

2 Enter the system names on which you want to install Oracle9i software and navigate to the Install or relink Oracle menu.

3 Select the version of Oracle software (Oracle 10gR1 or Oracle 10gR2).

4 Select Install Oracle RDBMS server to install Oracle software.

5 In the Oracle user name screen, enter the Oracle UNIX user name when prompted. The installer checks for the user on all systems.

6 In the Oracle Group name screen, enter the name of the Oracle inventory group when prompted. The installer checks for the existence of this group on all systems.

7 In the Oracle base directory screen, specify the base of the Oracle directory structure for OFA compliant databases.

8 In the Database Home Directory screen, specify the location of $ORACLE_HOME.

9 In the Patch Level screen, specify the last two digits of the patch level of Oracle database software being installed or relinked. For example, enter 0.5 for 10.1.0.5; enter 0.2 for 10.2.0.2.

10 The installer utility prints the entered information for verification. Answer "y" if the information displayed is correct, answer "n" otherwise.

For example:

```
Oracle environment information verification
Oracle Unix user: oracle
Oracle Unix Group: dba
Oracle Release: 10.2
Patch Level: 0.2
Oracle Base: /app/oracle
Oracle Home: /app/oracle/orahome
Is this information correct: [y,n,q] (y)
```

11 Installer now copies 5.0 SF Oracle RAC libraries to ORACLE_HOME and then relinks Oracle.

```
Verifying binaries in /app/oracle/orahome on galaxy ...ok
Verifying binaries in /app/oracle/orahome on nebula ...ok
Copying SFRAC libskgxn on galaxy .......................ok
Copying SFRAC libskgxn on nebula .......................ok
Copying SFRAC ODM library on galaxy ....................ok
Copying SFRAC ODM library on nebula ....................ok
Copying SFRAC libskgxp on galaxy .......................ok
Copying SFRAC libskgxp on nebula .......................ok
Relinking Oracle on galaxy .............................ok
Relinking Oracle on nebula .............................ok

Oracle Relinking is now complete.
```

12 After relinking is complete, enter "q" to quit the SF Oracle RAC installer.

Creating the Oracle database

Create the Oracle database on shared storage. Use you own tools or refer to Appendix B, "Creating a starter database" on page 443 for guidelines on using the Oracle `dbca` (Database Creation Assistant) tool to create a database on shared raw VxVM volumes or shared VxFS file systems.

Configuring the Oracle Service Group in a VCS Configuration

After you install Oracle10g and create a database, make the proper modifications in the VCS configuration file. Refer to Chapter 13, "Configuring Oracle 10g service groups" on page 233 for details on configuring service groups in an Oracle 10g environment.

Chapter 12

Upgrading and migrating Oracle software

Use this chapter for upgrading Oracle software and migrating Oracle databases:

- Oracle 9iR2 to Oracle 10gR2
- Oracle 10gR1 to Oracle 10gR2

Migrating from Oracle9i R2 to Oracle 10g

The migration procedure assumes that the beginning configuration includes the following are up and running on the cluster nodes:

- AIX 5.3 TL3 SP4 or AIX 5.2 TL 7
- Storage Foundation for Oracle RAC 5.0
- Oracle 9iR2

Tasks for Migration:

- Completing pre-upgrade Tasks
- Installing Oracle 10.2.0.1
- Migrating an existing Oracle 9iR2 database to Oracle 10gR2
- Unfreezing the VCS service groups

Completing pre-upgrade Tasks

To complete pre-upgrade tasks

1. Upgrade OS and install any patches, if required. Refer to Oracle metalink note 169706.1, "Operating Systems Installation and Configuration Requirements Quick Reference" for details.

2 Take a hot or cold backup of the existing Oracle 9iR2 database.

3 Take a backup of the existing Oracle Home and Central Inventory.

4 Freeze the VCS service groups that contain resources to monitor Oracle database instances and/or listener processes. As root user, enter:

```
#haconf -makerw
#hagrp -freeze oradb1_grp -persistent
#haconf -dump -makero
```

5 Shutdown Oracle database instances and listener processes on each node in the cluster.

6 Stop the Global Services Daemon, gsd. As "oracle" user, enter:

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

7 7.Rename the SRVM configuration file. As "oracle" user, enter:

```
$cd /var/opt/oracle
$mv srvConfig.loc srvConfig.loc_backup
```

8 Configure Private IP addresses for CRS communication. Refer to "Chapter 10" for details.

9 Create home directory for CRS binaries and a new home directory for Oracle 10gR2 binaries. Refer to "Chapter 10" for details.

10 Create diskgroup, volume and filesystem for OCR and vote disk. Refer to "Chapter 10" for details. Add a CFSMount and CVMVolDg resource to VCS configuration for monitoring the shared diskgroup and filesystem.

Installing Oracle 10g

To install Oracle 10g

1 Install 10g Oracle Clusterware Software. Refer to the instructions in "Chapter 11" for detailed instructions.

2 Verify that the CRS daemons are running. As user "oracle", enter:

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

3 Configure the virtual IP addresses using VIPCA

4 Add the "cssd" resource (for monitoring the CRS daemons) to VCS configuration.

5 Install 10.2.0.1 Oracle RDBMS software. For detailed instructions:
See “Installing Oracle 10g RAC” on page 213.

6 Install the latest patchset. For detailed instructions:
See “Installing Oracle 10g RAC” on page 213.

Migrating an existing Oracle 9iR2 database to Oracle 10gR2

To migrate an existing database fromOracle 9iR2 database to Oracle 10gR2

Upgrade the database to Oracle 10gR2. Refer to Oracle metalink Note, 316889.1 and the patchset upgrade procedure for details.

Unfreezing the VCS service groups

To unfreeze the VCS service groups

Unfreeze the VCS service groups that were frozen earlier in step 4 of "Pre-upgrade Tasks". As root user, enter:

```
#haconf -makerw
#hagrp -unfreeze oradb1_grp -persistent
#haconf -dump -makero
```

Migrating from Oracle10gR1 to Oracle 10.2.0.1

The migration procedure assumes that the beginning configuration includes the following are up and running on the cluster nodes:

- AIX 5.3 TL3 SP4 or AIX 5.2 TL 7
- Storage Foundation for Oracle RAC 5.0
- Oracle 10gR1

Tasks for Migration:

- Completing pre-upgrade Tasks
- Installing Oracle 10.2.0.1
- Completing post-upgrade tasks

Completing pre-upgrade tasks

To complete pre-upgrade tasks

1. Upgrade the OS and install any patches, if required. Refer to Oracle metalink note 169706.1, "Operating Systems Installation and Configuration Requirements Quick Reference"..
2. Take a hot or cold backup of the existing database
3. Take a backup of the existing Oracle home and central inventory.
4. Shutdown the Oracle instance.
   - If Oracle is under VCS control, freeze the `oracle` group:

```
# haconf -makerw
# hagrp -freeze <oracle group> -persistent
# haconf -dump -makero
```

- Use oracle commands to shutdown oracle.

5 Shutdown CRS.

- If CRS is under VCS

```
# haconf -makerw
# hagrp -freeze <cssd group> -persistent
# haconf -dump -makero
```

- Stop CRS

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

Installing Oracle 10.2.0.1

After completing the pre-upgrade tasks, complete the upgrade procedure.

To upgradefrom Oracle10gR1 to Oracle 10.2.0.1

1 Run pre-update.

```
# /cdrom/CRS/upgrade/preupdate.sh
```

2 Install the 10gR2 CRS.
See "Installing CRS" on page 213.

3 As root user run rootupgrade as directed by CRS installer

```
# $ORA_CRS_HOME/install/rootupgrade
```

4 Make sure 10gR2 CRS is running.

- To list the version of CRS software installed

```
# $ORA_CRS_HOME/bin/crsctl query crs softwareversion
```

- To list the CRS software operating version

```
# $ORA_CRS_HOME/bin/crsctl query crs activeversion
```

5 Install the 10gR2 RDBMS
See "Installing Oracle 10g database software" on page 217.

Migrating the existing Oracle 10gR1 database to Oracle 10gR2

Upgrade the database to Oracle 10gR2. Refer to Oracle metalink 316889.1 and the patchset upgrade procedure for details.

Completing post-upgrade tasks

To complete post upgrade tasks

If CRS and Oracle are under VCS control unfreeze the service group

```
# haconf -makerw
# hagrp -unfreeze <cssd group> -persistent
# hagrp -unfreeze <oracle group> -persistent
# haconf -dump -makero
```

Chapter 13

Configuring Oracle 10g service groups

After you have installed Oracle and created your database, you can set up VCS to automate the Oracle RAC environment:

You can set up VCS to automate the Oracle RAC environment:

- About VCS service group for Oracle 10g dependencies
- Configuring CVM and Oracle Service Groups
- Location of VCS log files

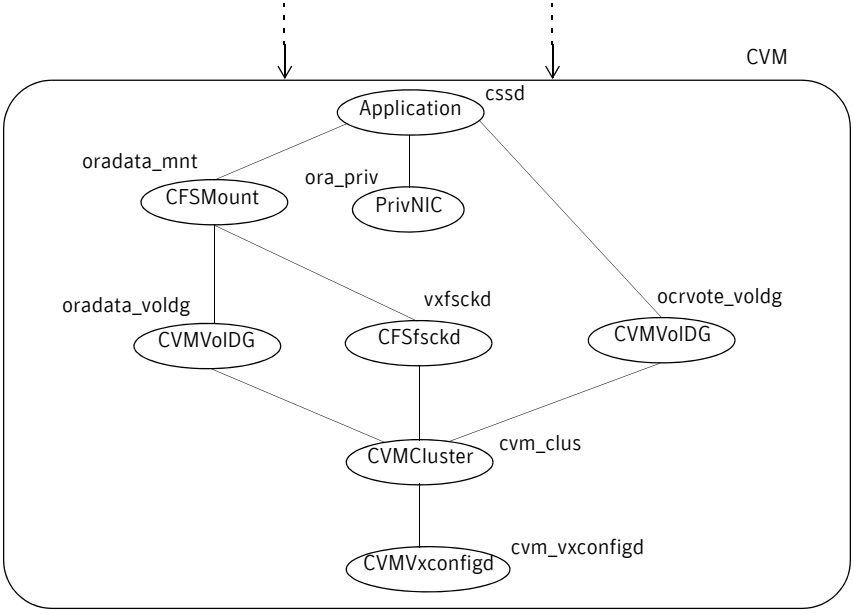
About VCS service group for Oracle 10g dependencies

Review the information on how to set up VCS to automate the Oracle 10g RAC environment and how VCS manages resources within a cluster.

VCS service group dependencies are based on whether you use the VCS Oracle agent or not. Figure 13-1 and Figure 13-2 illustrate the dependencies.

- In a configuration without the VCS Oracle agent, CRS controls the database. See Figure 13-1, "Configuration without the Oracle Agent."
- In a configuration with the VCS Oracle agent, VCS controls the Oracle database. An online local firm dependency exists between the Oracle group and the CVM group. For more details on service group dependencies, refer to the *Veritas Cluster Server User's Guide*.

See Figure 13-2, "Configuration with the Oracle Agent."



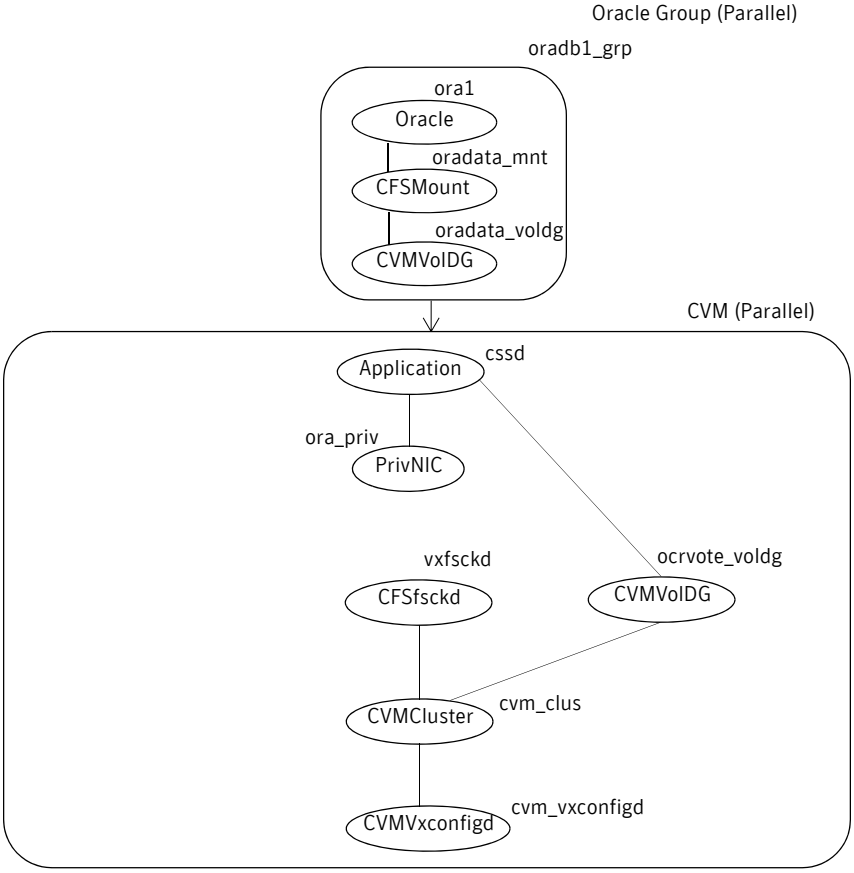

Oracle Group (Parallel)
oradb1_grp
ora1
Oracle
oradata_mnt
CFSMount
oradata_voldg
CVMVolDG
CVM (Parallel)
Application
cssd
ora_priv
PrivNIC
vxfsckd
ocrvote_voldg
CFSfsckd
CVMVolDG
cvm_clus
CVMCluster
cvm_vxconfigd
CVMVxconfigd

Figure 13-1 Configuration without the Oracle Agent

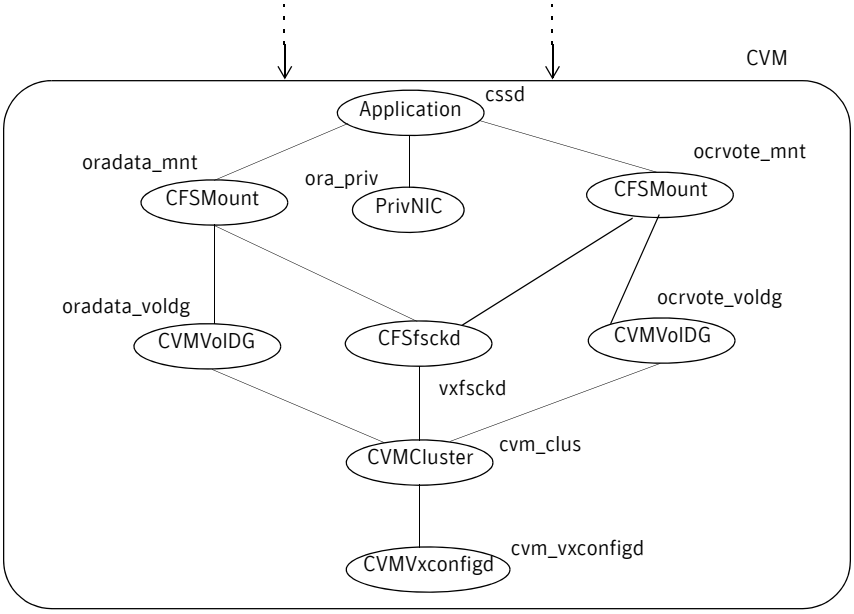
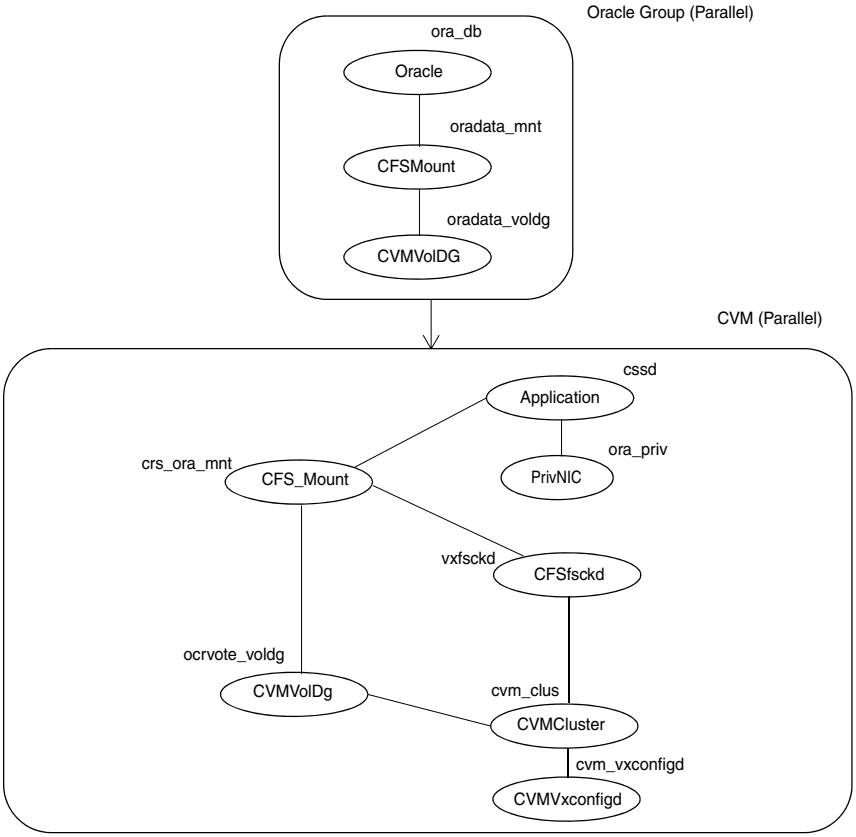


Figure 13-2 Configuration with the Oracle Agent



Configuring CVM and Oracle Service Groups

The CVM and Oracle service groups can be configured using the following two methods:

- By editing the VCS configuration file, main.cf, to define the service groups; see "Location of VCS log files" on page 247.
- By using a configuration wizard for Oracle RAC; see "Creating service groups using the configuration wizard" on page 239.

Configuring CVM Service Group for Oracle 10*g* Manually

This section describes how to manually edit the `main.cf` file to configure the CVM and Oracle service groups.

To configure CVM service group for Oracle 10*g* manually

This section describes how to manually edit the `main.cf` file to configure the CVM and Oracle service groups.

To configure CVM service group for Oracle 10*g* manually

1. Log in to one system as `root`.
2. Save your existing configuration to prevent any changes while you modify `main.cf`:

   ```
   # haconf -dump -makero
   ```

   If the configuration is not writable, a warning appears: “Cluster not writable.” You may safely ignore the warning.
3. Make sure VCS is not running while you edit `main.cf` by using the `hastop` command to stop the VCS engine on all systems and leave the resources available:

   ```
   # hastop -all -force
   ```
4. Make a backup copy of the main.cf file:

   ```
   # cd /etc/VRTSvcs/conf/config
   # cp main.cf main.orig
   ```
5. Using `vi` or another text editor, edit the `main.cf` file, modifying the `cvm` service group and creating `Oracle` service groups using the sample main.cf as a guideline.

Sample main.cf for *Oracle 10g*

This section provides the following sample files:

- “Oracle 10g configuration without Oracle agent” on page 429
- “Oracle 10g configuration with Oracle agent” on page 431

When you finish configuring the CVM and Oracle service groups by editing the `main.cf` file, verify the new configuration.

To save and check the configuration

1. Save and close the `main.cf` file.
2. Verify the syntax of the file `/etc/VRTSvcs/conf/config/main.cf`:

   ```
   # cd /etc/VRTSvcs/conf/config
   # hacf -verify .
   ```
3. Start the VCS engine on one system:

   ```
   # hastart
   ```
4. Type the command `hastatus`:

   ```
   # hastatus
   ```

5 When "LOCAL_BUILD" is listed in the message column, start VCS on the other system:

```
# hastart
```

6 Verify that the service group resources are brought online. On one system, enter:

```
# hagrp -display
```

To verify the state of newly added resources

1 Use `hargrp -state` to check status of the `cvm` group.

2 Use `hargrp -state` to check status of resources.

Modifying the VCS configuration

For additional information and instructions on modifying the VCS configuration by editing the `main.cf` file, refer to the *VERITAS Cluster Server User's Guide*.

Creating service groups using the configuration wizard

You can use a configuration wizard to configure the VCS service groups for Storage Foundation for Oracle RAC environment. The wizard enables you to modify the CVM service group to include the CRS resources. Note that the wizard for Oracle 10*g* does *not* create the service group for the Oracle database. To monitor the Oracle database using the Oracle Agent provided by VCS, you must edit the `main.cf` manually after you finish running the wizard. See "Location of VCS log files" on page 247 for details.

Before Starting the Wizard

Before starting the Wizard, you can verify that your Oracle installation can be configured. Review the requirements listed below. Also, you need to provide the wizard information as it proceeds. Make sure you have that information at hand.

Prerequisites

- Oracle RAC instances and listeners must be running on all cluster nodes.
- The database files of all instances must be on a cluster file system.

Note: The Wizard does not support using the same file system for the Oracle binary and Oracle datafiles.

- The OCR file and VOTE file location must be on a raw volume or a cluster file system.
- Each Oracle instance must be associated with a listener.

Note: The RAC configuration wizard requires that for the default listener, the listener parameter file, listener.ora, must reside in $ORACLE_HOME/network/admin. No such restriction applies for non-default listeners.

- The IP addresses and host names specified in the files listener.ora and tnsnames.ora must be the same.
- Virtual IPs required for CRS must be up.

Information Required From the User

- RAC database instances to be configured
- NICs for Private NIC resource
- Registry and vote disk location for CRS

Establishing graphical access for the wizard

The configuration wizard requires graphical access to the VCS systems where you want to configure service groups. If your VCS systems do not have monitors, or if you want to run the wizards from a remote HP system, do the following:

To establish graphical access from a remote system

1. From the remote system, (`jupiter`, for example), run `xhost +`

   ```
   # xhost +
   ```

2. Complete one of the following operations (depending on your shell):
   - If you are running `ksh`, run this step on one of the systems where the wizard will run (for example, `jupiter`):

     ```
     # export DISPLAY=jupiter:0.0
     ```

   - If you are running `csh`, run this step

     ```
     # setenv DISPLAY jupiter:0.0
     ```

3. Verify the DISPLAY environment variable is updated:

   ```
   # echo $DISPLAY
   jupiter:0.0
   ```

4. Make sure to set the JRE_HOME variable to `/opt/VRTSjre/jre1.4`. If VRTSjre1.4 is not installed, the hawizard exits after displaying an error message.

Creating service groups using the configuration wizard

Start the configuration wizard for Oracle 9i RAC at the command-line.

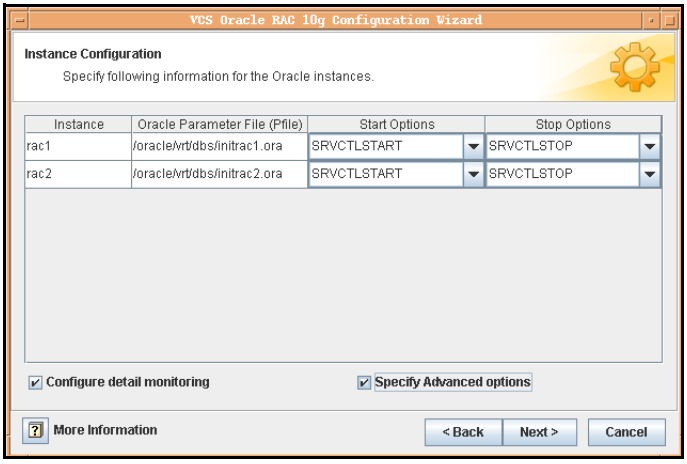
To create service groups using the configuration wizard

1. Log on to one of your VCS systems as superuser.
2. Start the configuration wizard.

   ```
   # /opt/VRTSvcs/bin/hawizard rac
   ```

3. Read the information on the Welcome screen.
   - If your configuration does not meet the requirements, click **Cancel** to stop the wizard. Start the wizard again after taking the necessary steps to meet the requirements.
   - If your configuration meets the requirements, click **Next**. The wizard begins discovering the current Oracle RAC information before proceeding.

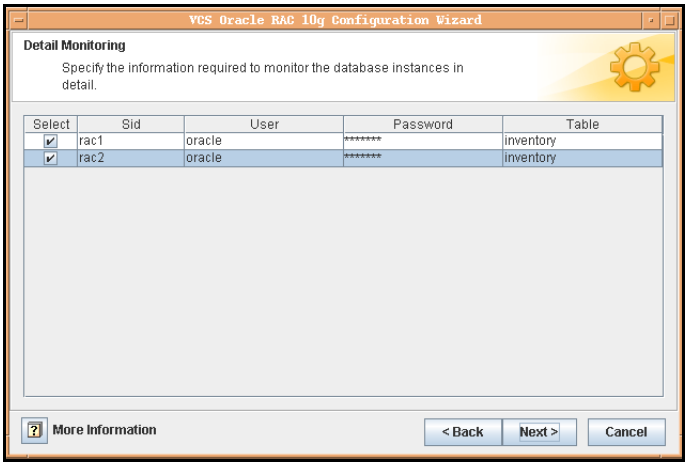
     If the wizard does not find all databases and listeners running on all nodes in the cluster, it halts with an error, indicating the problem. Click **Cancel**, and start the wizard again after you correct the problem.
4. In the Wizard Options dialog box, select the **Create RAC Service Group** option.
5. Enter a name for the RAC service group in the **Service group name** box and click **Next**.
6. In the Database Selection dialog box, select a database and click **Next**.
7. In the Instance Configuration dialog box, specify information for all instances of the database you selected.



Specify the following information for each Oracle instance that is displayed and click **Next**:

Oracle Parameter File (Pfile)	Verify the location of the Oracle Parameter File. Edit the information if necessary.
Start Options	Choose the Start options, if desired. Default is STARTUP_FORCE.
Stop Options	Choose the Stop options, if desired. Default is IMMEDIATE.
Configure detail monitoring	Select the check box if you want to monitor the database in detail. If you want to enable Detail Monitoring, be sure you have previously set up the database table, user, and password for the agent to use during monitoring.
Specify Advanced Options	Select the check box to enter advanced configuration information for the database instances.

8 If you chose to monitor the database in detail, the Detail Monitoring dialog box is displayed.



Specify the following information for the database instances that you want the agent to monitor in detail and click **Next**:

Select Select the check box corresponding to the database to be monitored in detail.

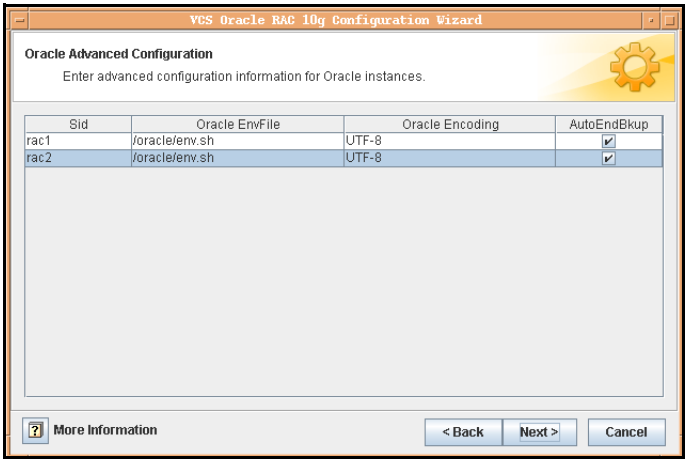
User Enter a valid user name for the database that the Oracle agent uses to log in to monitor the health of the database.

Password Enter a valid password for the database user.

Note: Do not encrypt passwords when entering them through the Agent Configuration Wizard; the wizard takes care of encrypting passwords.

Table Enter the name of a table that will be queried to validate the status of the database.

9 If you chose to specify advanced options, the Oracle Advanced Configuration dialog box is displayed.



Specify the following information for the Oracle instances that you want to configure advanced attributes and click **Next**:

Oracle EnvFile Enter the location of the Oracle Envfile, the source file used by the agent entry point scripts.

Oracle Encoding Enter the the operating system encoding that corresponds to Oracle encoding for the displayed Oracle output.

The encoding value must match the encoding value used by the Netlsnr configuration.

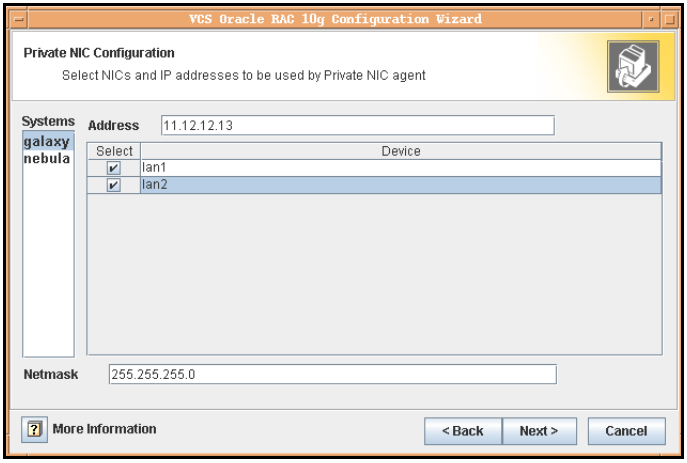
AutoEndBkup Select the check box, if desired.

Specifies that data files in the database are taken out of the backup mode when instance is brought online.

Refer to the *Veritas High Availability Agent for Oracle Installation and Configuration Guide* for a complete description of these attributes.

10 In the Monitor option Configuration dialog box, specify the monitor option for the Oracle instances, and click **Next**.
The default monitor option is **Process check**.

11 In the Private NIC Configuration dialog box, specify the NIC and IP address for Private NIC agent.



Specify the following information for each node in the cluster and click **Next**:

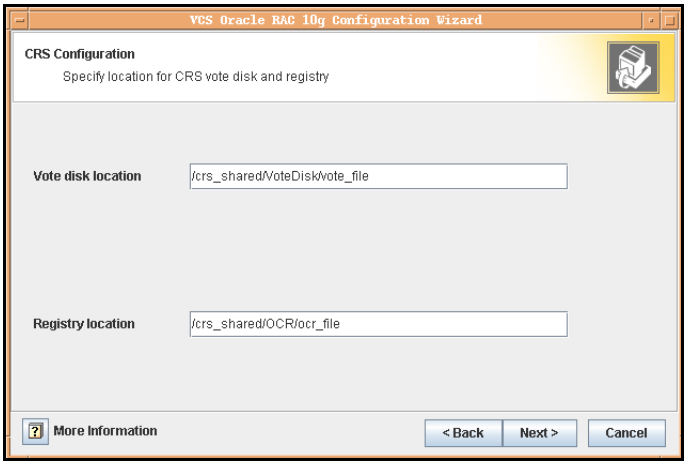
Address Enter the private IP address that is used by Oracle 10*g* CRS.

Select Select the checkbox against the network cards in the **Device** column. This NIC will be used by the PrivNIC agent.

Netmask Enter the netmask.

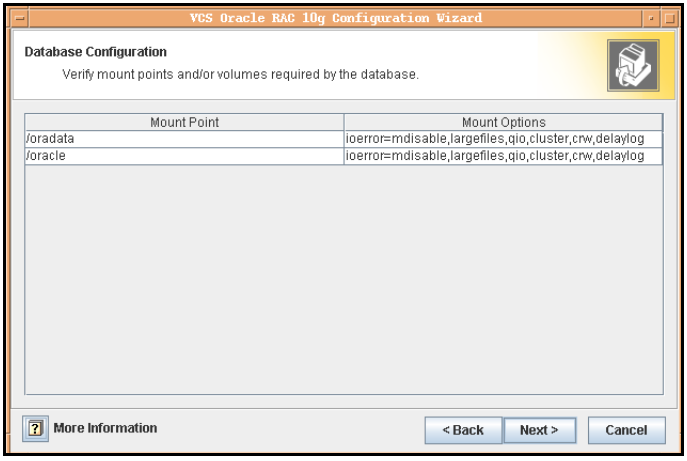
12 In the CRS Configuration dialog box, specify the location for CRS vote disk and OCR registry.
Enter the cluster file system or raw volume location for the CRS vote disk and registry. Example vote disk location:

- /ora_crs/VOTE-disk/vote_file (if you are using a cluster file system)
- /dev/vx/rdsk/crs_oradg/crsvol (if you are using raw volumes)



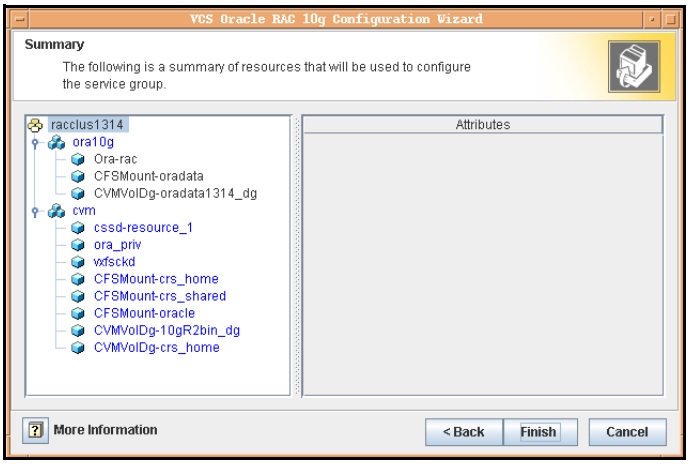
13 In the Database Configuration dialog box, verify the mount point of the database that the wizard displays. Confirm or modify the mount options displayed and click **Next**.

Note that the wizard discovers the mount point if the database is installed on a cluster file system. If the database exists on raw volumes, the wizard discovers the volumes.



14 In the Service Group Summary dialog, review your configuration.

Click on a resource to view its attributes and their configured values in the **Attributes** box.



- Click a resource within the service group to display its attributes and their values.

  For example, if you click on the name of the cssd application resource, cssd-resource, the wizard displays details of the cssd application resource.

  Attributes for the CFSMount resource show dependencies.

  The NetLsnr resource is configured as part of the CVM service group. The CVM service group also contains other resources, which may not be displayed by the wizard because the wizard does not control them.
- Change names of resources, if desired; the wizard assigns unique names to resources based on their respective name rules.

  To edit a resource name, select the resource name and click on it, press Enter after editing each attribute.

15 Review your configuration and click **Finish**.

The wizard starts running commands to create the Oracle RAC service group. Various messages indicate the status of these commands.

16 In the Completing the Oracle Configuration wizard dialog box, select the **Bring the service group online** check box to bring the service group online on the local system.

17 Click **Close**.

The wizard creates the Oracle RAC service group in your cluster and adds the Netlsnr resource to the CVM configuration.

Location of VCS log files

On all cluster nodes, look at the log files for any errors or status messages:

```
/var/VRTSvcs/log/engine_A.log
```

When large amounts of data are written, multiple log files may be required. For example, `engine_B.log`, `engine_C.log`, and so on, may be required. The `engine_A.log` contains the most recent data.

Chapter 14

Adding and removing cluster nodes for Oracle 10g

A cluster running Veritas *Storage Foundation for Oracle RAC* can have as many as eight systems. If you have a multi-node cluster running Oracle 10g, you can add or remove a node:

Adding a node to an Oracle 10g cluster

The examples used in these procedures describe adding one node to a two-system cluster.

Checking system requirements for new node

Ensure that the new systems meet all requirements for installing and using Storage Foundation for Oracle RAC.

✔ The new system must have the identical operating system and patch level as the existing systems.

✔ Use a text window of 80 columns minimum by 24 lines minimum; 80 columns by 24 lines is the recommended size for the optimum display of the `installsfrac` script.

Modifying /etc/pse.conf to enable the dlpi driver

The file /etc/pse.conf must be configured to enable the Streams DLPI driver. The following procedure describes checking the file and modifying it, if necessary, and loading the dlpi driver.

1. Check to see if the Ethernet driver is configured in the /etc/pse.conf file:
   ```
   # egrep 'ethernet driver' /etc/pse.conf
   ```
2. In the output, examine the line containing the "ethernetdriver" expression:
   ```
   #d+ dlpi en /dev/dlpi/en # streams dlpi ethernet driver
   ```
3. 3. If the comment symbol ("#") precedes the line, the DLPI driver is not configured. Using vi or another text editor, edit the file:
   ```
   # vi /etc/pse.conf
   ```
4. Find the section in the file labeled "#PSE drivers" and look for the line shown in step 2. Uncomment the line by removing the initial "#" symbol.
5. Save and close the file.
6. To load the dlpi driver, enter:
   ```
   # strload -f /etc/dlpi.conf
   ```

Physically adding a new system to the cluster

The new system must have the identical operating system and patch level as the existing systems. When you physically add the new system to the cluster, it must have private network connections to two independent switches used by the cluster and be connected to the same shared storage devices as the existing nodes. Refer to the *Veritas Cluster Server Installation Guide*.

After installing Storage Foundation for Oracle RAC on the new system and starting VxVM, the new system can access the same shared storage devices. The shared storage devices, including coordinator disks, must be exactly the same among all nodes. If the new node does not see the same disks as the existing nodes, it will be unable to join the cluster.

Installing Storage Foundation for Oracle RAC on the new system

Read the pre-installation instructions in this guide before proceeding.

To install Storage Foundation for Oracle RAC without configuration

1. Log in as root on one of the systems for installation.
2. Install the Veritas Storage Foundation for Oracle RAC software as described in Chapter 2 and Chapter 3 of this guide, but run the product installation script instead of the generic `installer` script. Enter the following command from the top-level directory of the mounted disc:

   ```
   # ./installsfrac -installonly [-rsh]
   ```

   The `-installonly` option is required to perform the installation without configuring the software. The `-rsh` option is required if you are using the remote shell (RSH) rather than the secure shell (SSH) to install the software simultaneously on several systems.
3. After the initial system checks are complete, press **Return** to start the requirements checks.
4. Enter the licenses when prompted.
5. After the requirements checks are complete, press **Return** to start installing the packages. If you are installing multiple nodes, you have the option of installing them simultaneously. You will be prompted after the installation is complete.
6. When installation is complete, note the locations of the summary, log, and response files indicated by the installer.

Note: Ignore the message advising that you must run `installsfrac -configure`. When adding a node to a cluster running Storage Foundation for Oracle RAC, you must manually configure the system using the following procedure.

Starting Volume Manager

As you run the utility, answer **n** to prompts about licensing. You installed the appropriate license when you ran the `installsfrac` utility.

To start Volume Manager

1. Run the installer:
   ```
   # vxinstall
   ```
2. Enter **n** to set up a systemwide disk group for the system. The installation completes.
3. Verify that the daemons are up and running. Enter the command:
   ```
   # vxdisk list
   ```
   The output should display the shared disks without errors.

Configuring LLT, GAB, VCSMM, and VXFEN drivers

To configure LLT, GAB, VCSMM, and VXFEN drivers

1. Edit the file `/etc/llthosts` on the two existing systems. Using `vi` or another text editor, add the line for the new node to the file. The file should resemble:
   ```
   1 galaxy
   2 nebula
   3 saturn
   ```
2. Copy the `/etc/llthosts` file from one of the existing systems over to the new system. The `/etc/llthosts` file must be identical on all nodes in the cluster.
3. Create an `/etc/lltab` file on the new system. For example:
   ```
   set-node saturn
   set-cluster 101
   link en1 /dev/dlpi/en:1 - ether - -
   link en2 /dev/dlpi/en:2 - ether - -
   ```
   The second line, the cluster ID, must be the same as in the existing nodes.
4. Use `vi` or another text editor to create the file `/etc/gabtab` on the new system. It should resemble the following example:
   ```
   /sbin/gabconfig -c -nN
   ```

Where *N* represents the number of systems in the cluster. For a three-system cluster, *N* would equal 3.

5 Edit the `/etc/gabtab` file on each of the existing systems, changing the content to match the file on the new system.

6 If you are adding the new node to a single node cluster, then fencing must be enabled and configured on the original node before proceeding tostep 7. See "Setting up I/O fencing" on page 105.

7 Set up the `/etc/vcsmmtab` and `/etc/vxfendg` files on the new system by copying them from one of the other existing nodes:

```
# rcp galaxy:/etc/vcsmmtab /etc
# rcp galaxy:/etc/vxfendg /etc
```

8 Run the commands to start LLT and GAB on the new node:

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

9 On the new node, start the VXFEN, VCSMM, and LMX drivers. Remove the `/etc/vxfenmode file` to enable fencing. Use the commands in the order shown:

```
# rm /etc/vxfenmode
# /etc/init.d/vxfen start
# /etc/init.d/vcsmm start
# /etc/init.d/lmx start
```

10 On the new node, start the GMS and ODM drivers. Use the commands in the order shown:

```
# /etc/rc.d/rc2.d/s99odm start
```

11 On the new node, verify that the GAB port memberships are a, b, d, and o. Run the command:

```
# /sbin/gabconfig -a
GAB Port Memberships
```

Preparing to add a node

Before configuring using the Oracle Add Node procedure, you must obtain IP addresses and configure CVM.

To prepare for installing Oracle

1 Obtain two IP addresses:

- one IP address for the private interconnect, which should be non-routable
- one public IP address to be plumbed as alias against the host interface, which *must* be on the same subnet as the system network interface

2 Create a local group and local user for Oracle. Be sure to assign the same group ID, user ID, and home directory as exists on the systems in the current cluster.

```
# mkgroup id='101' oinstall
# mkgroup id='102' dba
# mkgroup id='103' oper
# mkuser id=1001 pgrp='oinstall' groups='dba,oper'
    home='/opt/oracle' oracle
```

3 Create a password for the user oracle:

```
# passwd oracle
```

4 Create the directory structure for all shared mount points as defined in the `main.cf` configuration file. Include the Oracle OCR and Vote disk mount point if on the file system, the Oracle binaries if on CFS, and the Oracle database. The directory structure must be same as defined on the systems in the current cluster.

Example of mount point for OCR and Vote disk:

```
# mkdir –p /ora_crs/
```

Example of mount point for Oracle binaries

```
# mkdir-p /app/oracle/orahome
```

Example of mount point for Oracle database:

```
# mkdir –p /oradata
```

5 Change nership and group to Oracle user.

```
# chown –R oracle:oinstall app/oracle
# chown –R oracle:oinstall app/crshome
# chown –R oracle:oinstall oradata
# chown –R oracle:oinstall ocrvote
```

Configuring CVM

As root user, execute the following on the CVM master node only.

To configure the CVM group in the main.cf file

1 Determine the CVM master node:

```
# vxdctl –c mode
```

2 Make a backup copy of the `main.cf` file.

```
# cd /etc/VRTSvcs/conf/config
# cp main.cf main.cf.2node
```

3 Use the commands to reconfigure the CVM group. On the CVM master node, execute:

```
# haconf -makerw
# hasys -add saturn
# hagrp -modify cvm SystemList -add saturn 2
# hagrp -modify cvm AutoStartList -add saturn
```

```
# hares -modify ora_priv Device -add qfe0 0 -sys saturn
# hares -modify ora_priv Device -add qfe1 1 -sys saturn
```

4 Verify the syntax of main.cf file:

```
# hacf -verify .
```

5 Stop the VCS engine on all systems, leaving the resources available.

```
# hastop -all -force
```

6 Copy the new version of the `main.cf` to each system in the cluster including the newly added system.

```
# rcp (or scp) main.cf  nebula:/etc/VRTSvcs/conf/config
# rcp (or scp) main.cf  saturn:/etc/VRTSvcs/conf/config
```

In the example, galaxy is the system where `main.cf` is edited; it does not need a copy.

7 Start VCS on the CVM master.

```
# hastart
```

8 Verify the CVM group has come online.

```
# hastatus -sum
```

9 To enable the existing cluster to recognize the new node, execute on the current node:

```
# /etc/vx/bin/vxclustadm -m vcs -t gab reinit
# /etc/vx/bin/vxclustadm nidmap
```

10 Repeat steps 7 through 9 on each system in the existing cluster.

11 Start CVM on the newly added node.

- Determine the node ID:

  ```
  # cat /etc/llthost
  ```

- Verify this host ID is seen by the GAB module.

  ```
  # gabconfig -a
  ```

- Start the VCS engine.
  - If on the newly added node ports f, u, v, or w were present before `hastart`, then the newly added node must be rebooted to properly start the VCS:

    ```
    # /usr/sbin/shutdown -y
    ```

  - If on the newly added node ports f, u, v, or w were not present before `hastart`, then use the following command to start VCS:

    ```
    # hastart
    ```

12 Verify the CVM group has come online on the newly added node.

```
# hastatus -sum
```

Using the Oracle add node procedure

For the Oracle procedure for adding a node, see:

Metalink Article 270512.1, Adding a Node to a 10g RAC Cluster

In this procedure, Oracle copies the CRS_HOME and ORACLE_HOME from an existing node in the cluster.

Sample main.cf for adding an Oracle 10g node

Changes to the sample `main.cf` for adding a node are highlighted in red.

```
include "types.cf"
include "CFSTypes.cf"
include "CVMTypes.cf"
include "OracleTypes.cf"
include "PrivNIC.cf"

cluster ora_cluster (
        UserNames = { admin = dOPhOJoLPkPPnXPjOM }
        Administrators = { admin }
        HacliUserLevel = COMMANDROOT
        CounterInterval = 5
        UseFence = SCSI3
        )

system galaxy (
        )

system nebula (
        )

system saturn (
        )

group cvm (
        SystemList = { galaxy = 0, nebula = 1, saturn = 2 }
        AutoFailOver = 0
        Parallel = 1
        AutoStartList = { galaxy, nebula, saturn }
        )
        CFSMount ocrvote_mnt (
                Critical = 0
                MountPoint = "/ocrvote"
                BlockDevice = "/dev/vx/dsk/ocrvotedg/ocrvotevol"
                )

        CFSMount oradata_mnt (
                Critical = 0
                MountPoint = "/oradata"
                BlockDevice = "/dev/vx/dsk/oradatadg/oradatavol"
                )

        CVMVolDg oradata_voldg (
                Critical = 0
```

```
                CVMDiskGroup = oradatadg
                CVMVolume = { oradatavol }
                CVMActivation = sw
                )

        CVMVolDg orabin_voldg (
                Critical = 0
                CVMDiskGroup = orabindg
                CVMVolume = { orabinvol }
                CVMActivation = sw
                )

        CVMVolDg ocrvote_voldg (
                Critical = 0
                CVMDiskGroup = ocrvotedg
                               CVMVolume = { ocrvotevol }
                CVMActivation = sw
                )

        CFSfsckd vxfsckd (
                )

        CVMCluster cvm_clus (
                CVMClustName = ora_cluster
                CVMNodeId = { galaxy = 1, nebula = 2, saturn = 3 }
                CVMTransport = gab
                CVMTimeout = 200
                )

        CVMVxconfigd cvm_vxconfigd (
                Critical = 0
                CVMVxconfigdArgs = { syslog }
                )

        PrivNIC ora_priv (
                Critical = 0
                Device = { en1 = 0, en2 = 1}
                                                Address@galaxy =
"192.11.12.58"
                Address@nebula = "192.11.12.59"
                Address@saturn = "192.11.12.60"
                NetMask = "255.255.255.0"
                )

cvm_clus requires cvm_vxconfigd

oradata_voldg requires cvm_clus
ocrvote_voldg requires cvm_clus

ocrvote_mnt requires vxfsckd
oradata_mnt requires vxfsckd
```

```
ocrvote_mnt requires ocrvote_voldg
oradata_mnt requires oradata_voldg
```

Removing a node from an Oracle 10g cluster

The examples used in these procedures describe removing one node from a three-system cluster.

Removing a Node from an Oracle 10g Cluster

For the Oracle procedure for removing a node, see:

Metalink document ID#269320.1, Removing a Node from a 10g RAC Cluster

Follow the instructions provided by Oracle.

Running the uninstallsfrac utility

You can run the script from any node in the cluster, including a node from which you are uninstalling Storage Foundation for Oracle RAC.

Note: Prior to invoking the `uninstallsfrac` script, all service groups must be brought offline and VCS must be shut down.

For this example, Storage Foundation for Oracle RAC is removed from the node named `saturn`.

To run the uninstallsfrac utility

1. Before starting ./uninstallsfrac, execute:

   ```
   #/opt/VRTSvcs/bin/hastop -local
   ```

2. As root user, start the uninstallation from any node from which you are uninstalling Storage Foundation for Oracle RAC. Enter:

   ```
   # cd /opt/VRTS/install
   # ./uninstallsfrac
   ```

3. The welcoming screen appears, followed by a notice that the utility discovers configuration files on the system. The information lists all the systems in the cluster and prompts you to indicate whether you want to uninstall from *all* systems. You must answer " **n**." For example:

```
        VCS configuration files exist on this system with the
following
        information:

         Cluster Name: rac_cluster101
         Cluster ID Number: 7
         Systems: galaxy nebula saturn
         Service Groups: cvm oradb1_grp

Do you want to uninstall SFRAC from these systems? [y,n,q] (y) n
```

Caution: Be sure to answer N. Otherwise the utility begins the procedure to uninstall Storage Foundation for Oracle RAC from *all* systems.

4 The installer prompts you to specify the name of the system from which you are uninstalling Storage Foundation for Oracle RAC:

```
        Enter the system names separated by spaces on which to
uninstall
        SFRAC:saturn
```

5 The uninstaller checks for Storage Foundation for Oracle RAC packages currently installed on your system. It also checks for dependencies between packages to determine the packages it can safely uninstall and in which order.

6 Enter **y** when the uninstaller has completed checking.

7 When you press Enter to proceed, the uninstaller stops processes and drivers running on each system, and reports its activities.

8 When the installer begins removing packages from the systems, it indicates its progress by listing each step of the total number of steps required.

9 When the uninstaller is done, it describes the location of a summary file and a log of uninstallation activities.

Editing VCS configuration files on existing nodes

After running `uninstallsfrac`, modify the configuration files on the existing remaining nodes to remove references to the deleted node(s).

Edit /etc/llthosts

On the each of the existing nodes, using `vi` or another editor, edit the file `/etc/llthosts`, removing lines corresponding to the removed nodes. For example, if `saturn` is the node being removed from the cluster, remove the line "3 saturn" from the file:

```
1 galaxy
2 nebula
3 saturn
```

The file should now resemble:

```
1 galaxy
2 nebula
```

Edit /etc/gabtab

In the file `/etc/gabtab`, change the command contained in the file to reflect the number of systems after the node is removed:

```
/sbin/gabconfig -c -nN
```

where *N* is the number of nodes remaining. For example, with two nodes remaining, the file resembles:

```
/sbin/gabconfig -c -n2
```

Modify the VCS configuration to remove a system

You can modify the VCS configuration using one of three possible methods. You can edit `/etc/VRTSvcs/conf/config/main.cf` (the VCS configuration file) directly, you can use the VCS GUI (Cluster Manager), or you can use the command line, as illustrated in the following example. Please refer to the *Veritas Cluster Server User's Guide* for details about how to configure VCS.

At this point in the process, all Oracle binaries have been removed from the system to be deleted. The instance has been removed from the database, that is, the thread disabled, and the `spfile<SID>.ora` edited by Oracle to remove any references to this instance. The next step is to remove all references in the `main.cf` to the deleted node(s).

As root user execute the following on the CVM master node only.

To modify the CVM group in the main.cf file

1 To determine the CVM master node execute:

```
# vxdctl -c mode
```

2 Make a backup copy of the `main.cf` file.

```
# cd /etc/VRTSvcs/conf/config
# cp main.cf main.cf.3node.bak
```

3 Use the following commands to reconfigure the CVM group. Execute:

```
# haconf -makerw
# hagrp -modify cvm SystemList -delete saturn
# hares -modify cvm_clus CVMNodeId -delete saturn
# hasys -delete saturn
# haconf -dump -makero
```

Example of main.cf file: see "Sample main.cf for Removing an Oracle 10g Node" on page 261.

4 Verify the syntax of main.cf file:

```
# hacf -verify .
```

The main.cf file now should not contain entries for system saturn.

5 Stop the VCS engine on all systems, leaving the resources available.

```
# hastop -all -force
```

6 Copy the new version of the `main.cf` to each system in the cluster.

```
# rcp (or scp) main.cf  galaxy:/etc/VRTSvcs/conf/config
# rcp (or scp) main.cf  nebula:/etc/VRTSvcs/conf/config
```

7 Start the VCS engine on the current system.

```
# hastart
```

8 Verify the CVM group has come online.

```
# hastatus -sum
```

9 Repeat commands step 7 through step 8 on each system in the existing cluster.

Sample main.cf for Removing an *Oracle 10g* Node

Changes to the sample `main.cf` for adding a node are highlighted in red.

```
include "types.cf"
include "CFSTypes.cf"
include "CVMTypes.cf"
include "OracleTypes.cf"
include "PrivNIC.cf"

cluster ora_cluster (
        UserNames = { admin = dOPhOJoLPkPPnXPjOM }
        Administrators = { admin }
        HacliUserLevel = COMMANDROOT
        CounterInterval = 5
        UseFence = SCSI3
        )

system galaxy (
```

```
        )

system nebula (
        )

system saturn (
        )

group cvm (
        SystemList = { galaxy = 0, nebula = 1, saturn = 2 }
        AutoFailOver = 0
        Parallel = 1
        AutoStartList = { galaxy, nebula, saturn }
        )
        CFSMount ocrvote_mnt (
                Critical = 0
                MountPoint = "/ocrvote"
                BlockDevice = "/dev/vx/dsk/ocrvotedg/ocrvotevol"
                )

        CFSMount oradata_mnt (
                Critical = 0
                MountPoint = "/oradata"
                BlockDevice = "/dev/vx/dsk/oradatadg/oradatavol"
                )

        CVMVolDg oradata_voldg (
                Critical = 0
                CVMDiskGroup = oradatadg
                CVMVolume = { oradatavol }
                CVMActivation = sw
                )

        CVMVolDg orabin_voldg (
                Critical = 0
                CVMDiskGroup = orabindg
                CVMVolume = { orabinvol }
                CVMActivation = sw
                )

        CVMVolDg ocrvote_voldg (
                Critical = 0
                CVMDiskGroup = ocrvotedg
                                CVMVolume = { ocrvotevol }
                CVMActivation = sw
                )

        CFSfsckd vxfsckd (
                )

        CVMCluster cvm_clus (
                CVMClustName = ora_cluster
```

```
                CVMNodeId = { galaxy = 1, nebula = 2, saturn = 3 }
                CVMTransport = gab
                CVMTimeout = 200
                )

        CVMVxconfigd cvm_vxconfigd (
                Critical = 0
                CVMVxconfigdArgs = { syslog }
                )

        PrivNIC ora_priv (
                Critical = 0
                Device = { en1 = 0, en2 = 1}
                                                Address@galaxy =
"192.11.12.58"
                Address@nebula = "192.11.12.59"
                Address@saturn = "192.11.12.60"
                NetMask = "255.255.255.0"
                )

cvm_clus requires cvm_vxconfigd

oradata_voldg requires cvm_clus
ocrvote_voldg requires cvm_clus

ocrvote_mnt requires vxfsckd
oradata_mnt requires vxfsckd

ocrvote_mnt requires ocrvote_voldg
oradata_mnt requires oradata_voldg
```

Chapter 15

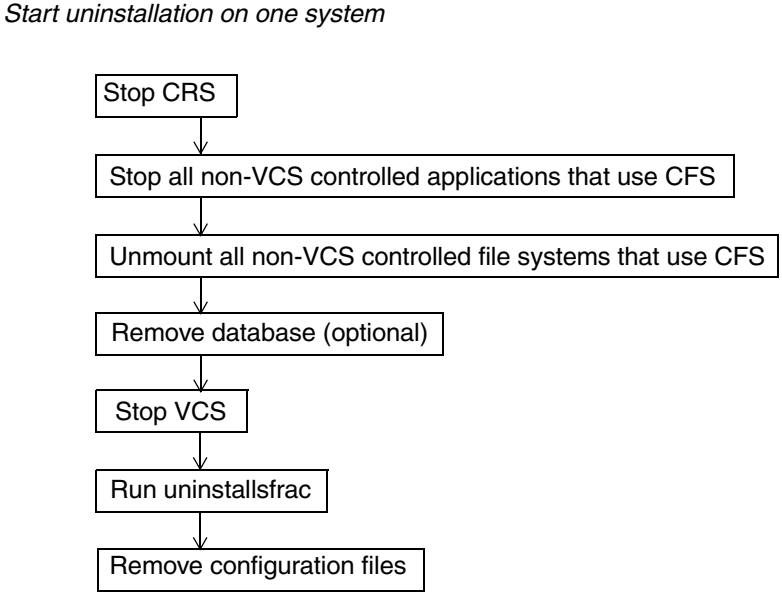
Uninstalling SF Oracle RAC from Oracle 10g systems

At the completion of the uninstallation procedure, you can continue to run Oracle using the single-instance binary generated when you unlink the Veritas binaries from Oracle.

Uninstalling SF Oracle RAC on Oracle 10*g*:

To uninstall SF Oracle RAC, you must remove all Veritas SF Oracle RAC software packages.

Figure 15-1 Uninstalling SF Oracle RAC



Offlining service groups

Offline all service groups and shutdown VCS prior to launching the `uninstallsfrac` script:

```
# /etc/init.crs stop
# /opt/VRTSvcs/bin/hastop -all
```

Do not use the `-force` option when executing `hastop`. This will leave all service groups online and shut down VCS, causing undesired results during the uninstallation of SF Oracle RAC.

Stopping Applications Using CFS (Outside of VCS Control)

All Oracle users must stop all applications using the CFS mounts *not* under VCS control.

To verify that no processes are using the CFS mount point

1 Enter:

```
# fuser -c mount_point
```

2 Stop any processes using a CFS mount point.

Unmounting VxFS File Systems (Outside of VCS Control)

All Oracle users must unmount any CFS file systems *not* under VCS control on all nodes.

To unmount CFS file systems not under VCS control

1 Determine the file systems to unmount by checking the output of the mount file. For example, type:

```
# mount | grep vxfs | grep cluster
```

2 By specifying its mount point, unmount each file system listed in the output:

```
# umount mount_point
```

Removing the Oracle Database (Optional)

You can remove the Oracle database after safely relocating the data as necessary.

Unlinking Veritas libraries from Oracle 10g binaries

If you have uninstalled Oracle, skip this procedure. If you have not uninstalled Oracle, unlink the Veritas libraries, using the following procedure, which generates a single-instance Oracle binary.

To unlink Oracle 10*g* binaries from Veritas libraries

1 Log in as the `oracle` user.

```
# su -oracle
```

2 Change to the ORACLE_HOME directory.

```
$ cd $ORACLE_HOME/lib
```

3 Restore the original Oracle libraries from their backup copies. There could be multiple backup copies as <library_name>.XX_XX_XX-XX_XX_XX. Run the following command to verify that it is an oracle library.

```
$strings <library_name>.XX_XX_XX-XX_XX_XX | grep -i vcs
```

The output should not have “vcs” string. Out of the libraries which do not have the “vcs” string select the library with the latest timestamp and use the following commands to restore them:

```
$ rm libskgxp10.a libskgxpu.a
$ cp libskgxpu.a.XX_XX_XX-XX_XX_XX libskgxpu.a
$ cp libskgxp10.a. XX_XX_XX-XX_XX_XX libskgxp10.a
$ rm libskgxn2.a libskgxn2.so
$ cp libskgxn2.a. XX_XX_XX-XX_XX_XX libskgxn2.a
$ cp libskgxn2.so. XX_XX_XX-XX_XX_XX libskgxn2.so
$ rm libodm10.so
$ cp libodmd10.so libodm10.so
```

4 Restore the original CRS library. Use the method in step 3 to identify the library to restore.

```
$ rm libskgxn2.so libskgxn2.a
$ cp libskgxn2.so. XX_XX_XX-XX_XX_XX libskgxn2.so
$ cp libskgxpn2.a. XX_XX_XX-XX_XX_XX libskgxn2.a
$ rm libskgxp10.a
$ cp libskgxp10.a. XX_XX_XX-XX_XX_XX libskgxp10.a
```

5 Change to the ORACLE _HOME directory and update the library. Use the method in step 3 to identify the library to restore.

```
$ cd $ORACLE_HOME/lib32
$ rm libskgxn2.so libskgxn2.a
$ cp libskgxn2.so. XX_XX_XX-XX_XX_XX libskgxn2.so
$ cp libskgxpn2.a. XX_XX_XX-XX_XX_XX libskgxn2.a
```

6 Change to the CRS_HOME directory and update the library. Use the method in step 3 to identify the library to restore.

```
$ cd $CRS_HOME/lib32
$ rm libskgxn2.so libskgxn2.a
$ cp libskgxn2.so. XX_XX_XX-XX_XX_XX libskgxn2.so
$ cp libskgxpn2.a. XX_XX_XX-XX_XX_XX libskgxn2.a
```

7 Relink Oracle.

```
$ cd $ORACLE_HOME/rdbms/lib
$ cp env_rdbms.mk.XX_XX_XX-XX_XX_XX env_rdbms.mk
$ make -f ins_rdbms.mk ioracle
```

Removing SF Oracle RAC packages

The `uninstallsfrac` script removes packages installed by `installsfrac` on all systems in the cluster. The installer removes all SF Oracle RAC `rpms`, regardless of the version of Oracle used.

To run uninstallsfrac

1 As root user, navigate to the directory containing the `uninstallsfrac` program.

```
# cd /opt/VRTS/install
```

2 Start `uninstallsfrac`:

```
# ./uninstallsfrac [-rsh]
```

The utility reports the cluster and systems for uninstalling.

3 Enter **y** if the cluster information is correct.
After entering the systems where the uninstallation will take place, the script checks the operating system on each system, verifies system-to-system communication, and sets up a log file.
The script checks for SF Oracle RAC packages currently installed the nodes. This process involves identifying system uninstall requirements and dependencies between packages to determine the safety and order of uninstalling packages.

4 Confirm to uninstall SF Oracle RAC. packages.

5 Review the output as the script stops processes and drivers running on each node, and reports its activities.

6 Review the output as the script indicates the progress of removing packages from the nodes by listing the steps that are completed. The total number of steps depends on the nature of the installation.

7 If necessary, review the summary and log files of uninstallation activities.

Removing other configuration files (optional)

You can remove the following configuration files:

```
/etc/vcsmmtab
/etc/vxfentab
/etc/vxfendg
/etc/llttab
/etc/gabtab
/etc/llthosts
```

Rebooting the Nodes

After uninstalling SF Oracle RAC, reboot each node:

```
# /usr/sbin/shutdown -Fr
```

Section V

Wide area disaster recovery for SF Oracle RAC

After installing and configuring SF Oracle RAC and Oracle,you can use these procedures to set up global clustering and replication for your RAC database:

Chapter 16

Preparing for global clustering

You can create a global cluster environment with SF Oracle RAC and volume replication capability. VCS provides the Global Cluster Option (GCO) for wide-area failover and disaster recovery, and Veritas Volume Replicator provides the volume replication capability.

If you are currently administering an Oracle database in a cluster running SF Oracle RAC, you can configure a global clustering environment with a new cluster on a secondary site. SF Oracle RAC 5.0 supports up to four nodes each on two clusters. You will need to set up a secondary site and modify the existing cluster configuration to support replication.

Preparing for global clustering requires:

- ✔ Creating a cluster on a secondary site with hardware set up.
- ✔ Installing SF Oracle RAC.
- ✔ Installing Oracle.
- ✔ Configuring VCS service groups.

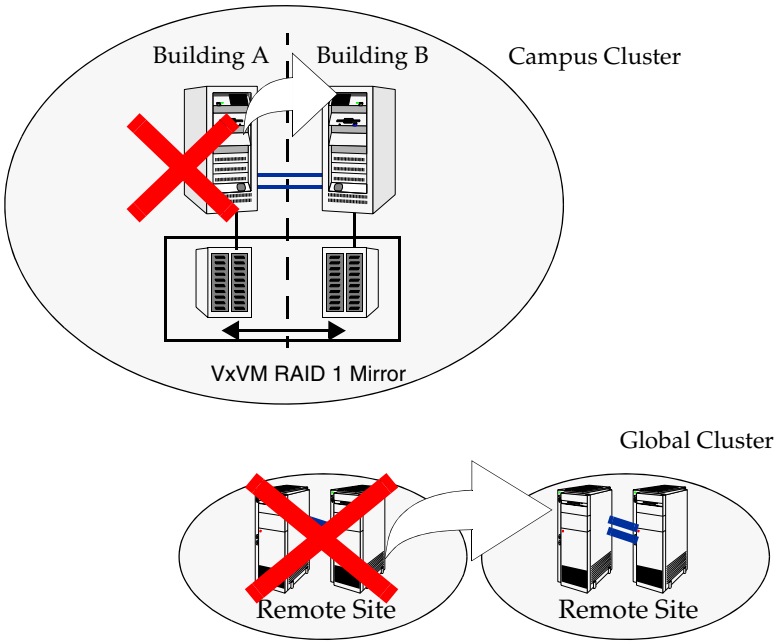
You do not need to create a database for the secondary site, since it will be replicated from the primary site.

Global clustering concepts

VCS provides the global clustering to enable wide-area failover and disaster recovery. Local clustering provides local failover for each site or building. Campus and replicated cluster configurations offer some degree of protection against disasters within limited geographic regions. These types of configurations do 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.

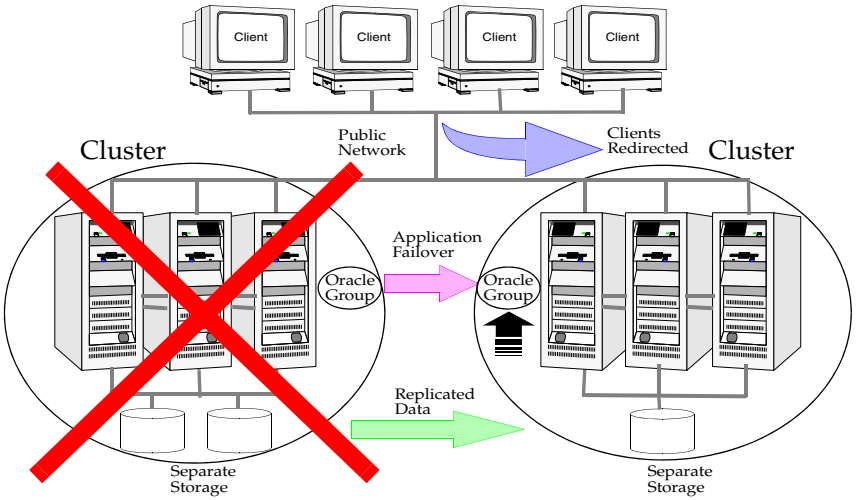


Global clustering for Oracle RAC

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 Oracle service groups are online on a node in cluster A and are configured to fail over on cluster A and cluster B.

Figure 16-1 Global clusters



VCS continuously monitors and communicates events between clusters. Cluster-to-cluster communications ensures that the global cluster environment is aware of the state of global service group at all times.

In the event of a local node or application failure, the Oracle RAC service groups become available on other cluster nodes. 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. For complete details on VCS global clusters:

See the *Veritas Cluster Server User's Guide*.

Replication in a shared disk environment

Veritas Volume Replicator (VVR) enables you to replicate data volumes on a shared disk group in SF Oracle RAC. In this environment, a two-node cluster on the primary site exists with a shared disk group. A two-node or single-node cluster exists on the secondary site; the disk group does not need be a shared disk group.

The VVR feature is provided with SF Oracle RAC. For configuring instructions for VVR:

See "Configuring global clustering" on page 281..

For complete details of VVR in a shared disk environment:

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

Setting up a secondary site

Setting up SF Oracle RAC in a global cluster environment to prepare for replication requires preparing two sites:

- ✔ Obtaining and installing license keys.
- ✔ Creating two clusters on two sites, setting up the hardware.
- ✔ Installing SF Oracle RAC.
- ✔ Installing Oracle software.
- ✔ Configuring VCS service groups for the secondary site.

Obtaining and installing license keys for VVR

Make sure you have licenses for the following products:

- Storage Foundation for Oracle RAC
- Veritas Volume Replicator (VVR)

See "Obtaining license keys" on page 71 for information on how to obtain license keys.

To install VVR license keys

1. Install the VVR license:

```
# vxlicinst -k xxxx-xxxx-xxxx-xxxx-xxxx-xxxx-xxxx-xx (CVM/VVR)
```

2. Make sure that "VVR" is enabled:

```
# vxlicrep | grep -e "CVM_FULL" -e "VVR"
```

Installing SF Oracle RAC on the secondary site

Important requirements for global clustering:

- Cluster names on the primary and secondary sites must be unique.
- Node and resource names must be unique within a cluster but not across clusters.

You can set up a multi-node or single-node cluster on the secondary site. The only difference between setting up a single node-cluster and a multi-node cluster is the fencing configuration.

To install SF Oracle RAC on a multi-node cluster

1. Prepare to install SF Oracle RAC by planning your installation, reviewing the requirements, and performing pre-installation tasks.

   "SF Oracle RAC prerequisites" on page 67.

"Performing preinstallation tasks" on page 70.

2 Install SF Oracle RAC.
   - Global clustering requires different names for the clusters on the primary and secondary sites.
   - To install a multi-node cluster:
     See "Installing and Configuring SF Oracle RAC Software" on page 83.

3 Set up storage on the secondary site and configure I/O fencing.
   See "Setting up I/O fencing" on page 105.
   - Verify the shared storage on the secondary site supports SCSI-3 reservations.
   - Set up the coordinator disks.
   - Configure I/O fencing for two-node clusters.

4 Update the `main.cf` file on the secondary site.
   See "Enabling fencing in the VCS configuration" on page 112.

5 Shut down and restart all nodes on the secondary site. After starting the cluster nodes, check the `cvm` group is online:

```
# hagrp -state cvm
```

   "Setting up I/O fencing" on page 105.

To install SF Oracle RAC on a one-node cluster

1 Prepare to install SF Oracle RAC by planning your installation, reviewing the requirements, and performing pre-installation tasks.
   "SF Oracle RAC prerequisites" on page 67.
   "Performing preinstallation tasks" on page 70

2 Install SF Oracle RAC.
   - Global clustering requires different names for the clusters on the primary and secondary sites.
   - To install a single-node cluster on the secondary site:
     See "Installing and Configuring SF Oracle RAC Software" on page 83.

3 Set up storage on the secondary site.

   Note: For a single node cluster, the steps to create a co-ordinator disk group and editing the fencing configuration files need not be performed. Fencing will run in disabled mode.

4 Shut down and restart the node on the secondary site. After starting the cluster node, check the `cvm` group is online:

```
# hagrp -state cvm
```

Installing Oracle on the secondary site

Make sure you have Oracle pre-configured and installed and configured on the secondary site.

To pre-configure for Oracle

1. Pre-configure Oracle.
   - For Oracle 10*g*:
     "Preparing to Install Oracle 10g RAC" on page 201.
   - For Oracle9*i*:
     Chapter 5, "Preparing to Install Oracle9i RAC" on page 129
2. Install Oracle on the secondary site.
   - Use the same name for the database disk group and the database volume on the secondary site as the one on the primary site.
   - Set the same capacity for the database disk group and the database volume on the secondary site as the one on the primary site.
   - Do not create a database on the secondary site. The disk group will contain the database replicated from the primary site.
   - For Oracle9i:
     Chapter 6, "Installing Oracle9i RAC" on page 137.
   - For Oracle 10g:
     See "Installing Oracle 10g RAC" on page 213.
3. On the secondary site, set up the disk group and volumes for the Oracle database, but do not create a database. It will be replicated from the primary site.
4. Make sure that the database disk group and database volume on the secondary site have the same name and size as that on the primary site.

Configuring VCS service groups for the secondary site

Make sure you have configured VCS service groups for Oracle on the secondary site.

- For Oracle 10*g*
  Chapter 13, "Configuring Oracle 10g service groups" on page 233
- For Oracle9*i*
  "Configuring Oracle9i service groups" on page 147.
- Each cluster requires a virtual IP address associated with the cluster. The VCS installation and creation of the `ClusterService` group typically involves defining this IP address. If you did not configure the

`ClusterService` group when you installed SF Oracle RAC, configure it when you configure global clustering.

See "Configuring global clustering" on page 281.

Chapter 17

Configuring global clustering

Once you have set up a secondary cluster running SF Oracle RAC, you can configure a global cluster environment. Modify both cluster configurations to operate with replication in the global cluster environment.

Configuring SF Oracle RAC for global clusters requires:

- ✔ Setting up both clusters as part of a global cluster environment.
- ✔ Setting up replication on both clusters.
- ✔ Starting replication of the database.
- ✔ Configuring VCS on the primary site for replication.
- ✔ Configuring VCS on the secondary site for replication.

Preparing clusters for replication

Before configuring clusters for global clustering, make sure both clusters have:

- SF Oracle RAC and Oracle installed and configured
- License keys installed for CVR
- VVR and global clustering enabled

Preparing clusters for replication in both sites requires:

- Adding the VVR types to the VCS configuration
- Configuring global clustering

Adding the VVR types to the VCS configuration

After SF Oracle RAC is installed on two clusters and fencing is configured, add the VVR types to the VCS configuration.

To add VVR types to the VCS configuration on each cluster

1. On the first cluster, `rac_cluster101`, make sure that CVM is up. For example:

```
LOCAL_CLUSTER NAME:  rac_cluster101 (galaxy,nebula)
REMOTE_CLUSTER_NAME: rac_cluster102 (mercury,jupiter)
```

2. Make sure you have installed CVR license keys with "VVR" enabled.
   See "Obtaining and installing license keys for VVR" on page 276.

3. On any node in one cluster, enable write access to the VCS configuration:

```
# haconf -makerw
```

4. Run the script to add definitions for VVR types:

```
# cd /etc/VRTSvcs/conf/sample_vvr
# ./addVVRTypes.sh
```

5. Save the VCS configuration and change the access to read-only:

```
# haconf -dump -makero
```

6. Perform step 3 through step 5 on the other cluster, `rac_cluster102`.

Configuring global clustering

Review the requirements for global clustering:

- ✔ Cluster names on the primary and secondary sites must be unique.
- ✔ Node and resource names must be unique within a cluster but not across clusters.
- ✔ Each cluster requires a virtual IP address associated with the cluster. The VCS installation and creation of the `ClusterService` group typically involves defining this IP address. If you did not configure the `ClusterService` group when you installed SF Oracle RAC, configure it when you configure global clustering.
- ✔ One WAN (Wide Area Network) heartbeat must travel between clusters, assuming each cluster has the means to monitor the health of the remote cluster. Configure the heartbeat resource manually.
- ✔ All oracle s must be the same on all nodes.
- ✔ The Oracle database, which VVR replicates from the storage on the primary site to the secondary site, must be defined in a global group having the same name on each cluster. Each resource in the group may differ from cluster to

cluster, but clients redirected to a remote cluster after a wide-area failover must see the same application as the one in the primary cluster.

You can modify the global clustering configuration using the following methods:

- global clustering wizard
- manual modification of the main.cf file

The global clustering wizard results is generally more accurate for basic configurations. The manual modification method provides greater opportunity for customazation but also greater opportunity for errors.

See the *Veritas Cluster Server User's Guide* for complete details on global clustering.

Modifying the global clustering configuration using the wizard

The global clustering wizard completes the following tasks:

- Validates the ability of the current configuration to support a global cluster environment.
- Creates the components that enable the separate clusters, each of which contains a different set of GAB memberships, to connect and operate as a single unit.
- Creates the `ClusterService` group, or updates an existing `ClusterService` group.

Run the global clustering configuration wizard on each of the clusters; you must have the global clustering license in place on each node in the cluster.

To use the global clustering wizard

1. On a node in the primary site, start the global clustering configuration wizard:

   ```
   # /opt/VRTSvcs/bin/gcoconfig
   ```

2. After discovering the NIC devices on the local node, specify or confirm the device for the cluster joining the global cluster environment.
3. Indicate whether the NIC you entered is for all cluster nodes. If you enter **n**, enter the names of NICs on each node.
4. Enter or confirm the virtual IP address for the local cluster.
5. When the wizard discovers the net mask associated with the virtual IP address, accept the discovered value or enter another value.

   With NIC and IP address values configured, the wizard creates a `ClusterService` group or updates an existing one. After modifying the VCS configuration file, the wizard brings the group online.

6 Perform through step 1 through step 5 on the secondary cluster.

Modifying the global clustering configuration using the main.cf

After you run the global clustering configuration wizard, the modifications to the `main.cf` file typically involve specifying the virtual IP address for the local cluster and defining the `ClusterService` group for the local cluster.

The example global clustering configuration shows the `rac_cluster101` cluster on the primary site. The additions to the configuration appear in bold text. Use the example as guidelines to add a cluster service group to main.cf.

```
include "types.cf"
include "CFSTypes.cf"
include "CVMTypes.cf"
include "OracleTypes.cf"
include "VVRTypes.cf"

cluster rac_cluster101 (
    UserNames = { admin = "cDRpdxPmHpzS." }
        ClusterAddress = "10.10.10.101"
    Administrators = { admin }
    CounterInterval = 5
    UseFence = SCSI3
    )


system galaxy (
    )

system nebula (
    )

group ClusterService (
    SystemList = { galaxy = 0, nebula = 0 }
    AutoStartList = { galaxy, nebula }
    OnlineRetryLimit = 3
    OnlineRetryInterval = 120
    )

    Application wac (
        StartProgram = "/opt/VRTSvcs/bin/wacstart"
        StopProgram = "/opt/VRTSvcs/bin/wacstop"
        MonitorProcesses = "/opt/VRTSvcs/bin/wac" }
        RestartLimit = 3
        )

    IP gcoip (
        Device =en0
        Address = "10.10.10.101"
```

```
        NetMask = "255.255.240.0"
        )

    NIC csgnic (
        Device =en0
        )

    gcoip requires csgnic
        wac requires gcoip
.
.
.

group cvm (
.
.
.
```

After using the global clustering wizard, the `main.cf` for the secondary site has a similar configuration to the one above.

Defining the remote cluster and heartbeat Cluster Objects

After configuring global clustering, add the `remotecluster` cluster object to define the IP address of the cluster on the secondary site, and the `heartbeat` object to define the cluster-to-cluster heartbeat.

Heartbeats monitor the health of remote clusters. VCS can communicate with the remote cluster only after you set up the `heartbeat` resource on both clusters.

To define the remote cluster and heartbeat

1. On the primary site, enable write access to the configuration:

   ```
   # haconf -makerw
   ```

2. Define the `remotecluster` and its virtual IP address. In this example, the remote cluster is `rac_cluster102` and its IP address is `10.11.10.102`:

   ```
   # haclus -add rac_cluster102 10.11.10.102
   ```

3. Complete step 1 and step 2 on the secondary site using the name and IP address of the primary cluster (`rac_cluster101` and `10.10.10.101`).

4. On the primary site, add the `heartbeat` object for the cluster. In this example, the heartbeat method is `ICMP` ping.

   ```
   # hahb -add Icmp
   ```

5. Define the following attributes for the `heartbeat` resource:

   - `ClusterList` lists the remote cluster.
   - `Arguments` enables you to define the virtual IP address for the remote cluster.

For example:

```
# hahb -modify Icmp ClusterList rac_cluster102
# hahb -modify Icmp Arguments 10.11.10.102 -clus
rac_cluster102
```

6 Save the configuration and change the access to read-only on the local cluster:

```
# haconf -dump -makero
```

7 Complete step 4 through step 6 on the secondary site using appropriate values to define the cluster on the primary site and its IP as the remote cluster for the secondary cluster.

8 Verify clusters in the global setup by executing haclus -list command.

```
# haclus -list
    rac_cluster101
    rac_cluster102
```

Example additions to the `main.cf` file on the primary site:

```
.
.
remotecluster rac_cluster102 (
Cluster Address = "10.11.10.102"
)
heartbeat Icmp (
    ClusterList = { rac_cluster102 }
    Arguments @rac_cluster102 = { "10.11.10.102" }
    )

system galaxy (
    )
.
.
Example additions to the main.cf file on the secondary site:
.
.
remotecluster rac_cluster101 (
    Cluster Address = "10.190.88.188"
    )

heartbeat Icmp (
    ClusterList = { rac_cluster101 }
    Arguments @rac_cluster102 = { "10.190.88.188" }
    )

system galaxy (
    )
.
.
```

See the *Veritas Cluster Server User's Guide* for details for configuring the required and optional attributes of the `heartbeat` object.

Setting up replication

Setting up replication in a global cluster environment involves the following tasks:

✔ Creating the SRL in the disk group for the database.

✔ Creating the RVG on the primary site.

✔ Setting up replication objects on the secondary site.

Creating the SRL volume on the primary site

Create the Storage Replicator Log (SRL), a volume in the Replicated Volume Group (RVG). The RVG also holds the data volumes for replication.

- The data volume on the secondary site has the same name and the same size as the data volume on the primary site.
  See "Setting up a secondary site" on page 276.
- The SRL on the secondary site has the same name and the same size as the SRL on the primary site.
- The data volume and the SRL should exist in the same disk group.
- If possible, create SRLs on disks without other volumes.
- Mirror SRLs and data volumes in the absence of hardware-based mirroring.

After determining the size of the SRL volume, create the volume in the shared disk group for the Oracle database. If hardware-based mirroring does not exist in your setup, use the `nmirror` option to mirror the volume. In this example, the Oracle database is in the `oradatadg` shared disk group on the primary site and the size required for the SRL volume is 1.5 GB:

To create the SRL volume on the primary site

1. On the primary site, determine the size of the SRL volume based on the configuration and use.
   See the Veritas Volume Replicator documentation for details.
2. Determine whether a node is the master or the slave:
   ```
   # vxdctl -c mode
   ```
3. From the master node, issue the following command:
   ```
   # vxassist -g oradatadgmake rac1_srl 1500M nmirror=2 disk4
   disk5
   ```
   Make sure that the data disk has a minimum of 500M of free space after creating the SRL volume.
4. Start the SRL volume by starting all volumes in the disk group:

```
# vxvol -g oradatadg startall
```

Setting up replication objects on the primary site

Before creating the RVG on the primary site, make sure the replication objects are active and online.

To review the status of replication objects on the primary site

1. Verify the volumes you intend to include in the group are active.
2. Review the output of the `hagrp -state cvm` command.
3. Check that the `cvm` group is online.

To create the RVG

The command to create the primary RVG takes the form:

```
vradmin -g disk_group createpri rvg_name data_volume srl_volume
```

where:

- `disk_group` is the name of the disk group containing the database
- `rvg_name` is the name for the RVG
- `data_volume` is the volume that VVR replicates
- `srl_volume` is the volume for the SRL

For example, to create the `rac1_rvg` RVG, enter:

```
# vradmin -g oradatadgcreatepri rac1_rvg rac1_vol rac1_srl
```

The command creates the RVG on the primary site and adds a Data Change Map (DCM) for each data volume. In this case, a DCM exists for `rac1_vol`).

Configuring replication for the secondary site

To create objects for replication on the secondary site, use the `vradmin` command with the `addsec` option. To set up replication on the secondary site:

✔ Creating a disk group on the storage with the same name as the equivalent disk group on the primary site if you have not already done so.
See "Installing Oracle on the secondary site" on page 278.

✔ Creating volumes for the database and SRL on the secondary site.
See "Creating the data and SRL volumes on the secondary site" on page 289.

✔ Editing the `/etc/vx/vras/.rdg` file on the secondary site.
See "Editing the /etc/vx/vras/.rdg files" on page 289.

✔ Resolvable virtual IP addresses that set network RLINK connections as host names of the primary and secondary sites.
See "Setting up IP addresses for RLINKs on each cluster" on page 290.

✔ Creating the replication objects on the secondary site.
See "Setting up disk group on secondary site for replication" on page 290.

Creating the data and SRL volumes on the secondary site

When creating volumes for the data and SRL:

- The sizes and names of the volumes must reflect the sizes and names of the corresponding volumes in the primary site.
- Create the data and SRL volumes on different disks in the disk group. Use the `vxdisk -g diskgroup list` command to list the disks in the disk group.
- Mirror the volumes.

To create the data and SRL volumes on the secondary site

1. In the disk group created for the Oracle database, create a volume for data; in this case, the `rac_vol1` volume on the primary site is 6.6 GB:

```
# vxassist -g oradatadg make rac_vol1 6600M nmirror=2 disk1
disk2
```

2. Create the volume for the SRL, using the same name and size of the equivalent volume on the primary site. Create the volume on a different disk from the disks for the database volume:

```
# vxassist -g oradatadg make rac1_srl 1500M nmirror=2 disk4
disk6
```

Editing the /etc/vx/vras/.rdg files

Editing the `/etc/vx/vras/.rdg` file on the secondary site enables VVR to replicate the disk group from the primary site to the secondary site. On each node, VVR uses the `/etc/vx/vras/.rdg` file to check the authorization to replicate the RVG on the primary site to the secondary site. The file on each node in the secondary site must contain the primary disk group ID, and likewise, the file on each primary system must contain the secondary disk group ID.

1. On a node in the primary site, display the primary disk group ID:

```
# vxprint -l diskgroup
.....
```

2. On each node in the secondary site, edit the `/etc/vx/vras/.rdg` file and enter the primary disk group ID on a single line.
3. On each cluster node of the primary cluster, edit the file and enter the primary disk group ID on a single line.

Setting up IP addresses for RLINKs on each cluster

Creating objects with the `vradmin` command requires resolvable virtual IP addresses that set network RLINK connections as host names of the primary and secondary sites.

To set up IP addresses for RLINKS on each cluster

1. For each RVG running on each cluster, set up a virtual IP address on one of the nodes of the cluster. These IP addresses are part of the RLINK. The example assumes that the public network interface is`en0:1`, the virtual IP address is `10.10.9.101`, and the net mask is `255.255.240.0` for the cluster on the primary site:

```
# ifconfig en0 10.180.13.140 netmask 255.255.248.0 broadcast
    10.180.95.255 alias
# ifconfig en0 up
```

2. Use the same commands with appropriate values for the interface, IP address, and net mask on the secondary site. The example assumes the interface is `en0:1`, virtual IP address is `10.11.9.102`, and the net mask is `255.255.240.0` on the secondary site.
3. Define the virtual IP addresses to correspond to a virtual cluster host name on the primary site and a virtual cluster host name on the secondary site. For example, update `/etc/hosts` file on all nodes in each cluster. The examples assume `rac_clus101_priv` has IP address `10.10.9.101` and `rac_clus102_priv` has IP address `10.11.9.102`.
4. Use the `ping` command to verify the links are functional.

Setting up disk group on secondary site for replication

Create the replication objects on the secondary site from the master node on the primary site, using the `vradmin` command.

To set up the disk group on the secondary site for replication

1. Issue the command in the following format from the cluster on the primary site:

   `vradmin -g` *`dg_pri`* `addsec` *`rvg_pri pri_host sec_host`*

   where:

   - `dg_pri` is the disk group on the primary site that VVR will replicate. For example: *`oradatadg`*
   - `rvg_pri` is the RVG on the primary site. For example: *`rac1_rvg`*
   - `pri_host` is the virtual IP address or resolvable virtual host name of the cluster on the primary site. For example: *`10.10.9.101`* or *`rac_clus101_priv`*

- `sec_host` is the virtual IP address or resolvable virtual host name of the cluster on the secondary site. For example: *`10.11.9.102`* or *`rac_clus102_priv`*

For example, the command to add the cluster on the primary site to the RDS is:

```
# vradmin -g oradatadg addsec rac1_rvg rac_clus101_priv
rac_clus102_priv
```

2 On the secondary site, the command:

- Creates an RVG within the specified disk group using the same name as the one for the primary site
- Associates the data and SRL volumes that have the same names as the ones on the primary site with the specified RVG
- Adds a data change map (DCM) for the data volume
- Creates cluster RLINKS for the primary and secondary sites with the default names; for example, the "primary" RLINK created for this example is `rlk_rac_clus102_priv_rac1_rvg` and the "secondary" RLINK created is `rlk_rac_clus101_priv_rac1_rvg`.

3 Verify the list of RVGs in the RDS by executing the following command.

```
# # vradmin -g oradg -l printrvg
```

For example:

```
Reeplicated Data Set: rac1_rvg
Primary:
HostName: 10.180.88.187 <localhost>
RvgName:  rac1_rvg
DgName: oradatadg
datavol_cnt: 1
vset_cnt: 0
srl: rac1_srl
RLinks:
name=rlk_10.11.9.102_ rac1_rvg, detached=on,
synchronous=off
Secondary:
HostName: 10.190.99.197
RvgName: rac1_rvg
DgName: oradatadg
datavol_cnt: 1
vset_cnt: 0
srl: rac1_srl
RLinks:
name=rlk_10.10.9.101_ rac1_rvg, detached=on,
synchronous=off
```

Note: Once the replication is started the value off detached flag will change the status from OFF to ON.

Starting replication of Oracle database volume

When you have both the primary and secondary sites set up for replication, you can start replication from the primary site to the secondary site.

Start with the default replication settings:

- Mode of replication: `synchronous=off`
- Latency Protection: `latencyprot=off`
- SRL overflow protection: `srlprot_autodcm`
- Packet size: `packet_size=8400`
- Network protocol: `protocol=UDP`

Method of initial synchronization:

- Automatic synchronization
- Full synchronization with Checkpoint

For for guidelines on modifying these settings and information on choosing the method of replication for the initial synchronization:

See the *Veritas Volume Replicator Administrator's Guide*

Starting replication using automatic synchronization

To start replication using automatic synchronization

From the primary site, automatically synchronize the RVG on the secondary site:

```
vradmin -g disk_group -a startrep pri_rvg sec_host
```

where:

- `disk_group` is the disk group on the primary site that VVR will replicate
- `pri_rvg` is the name of the RVG on the primary site
- `sec_host` is the virtual host name for the secondary site

For example:

```
# vradmin -g oradatadg -a startrep rac1_rvg rac_clus102_priv
```

Because the cluster on the secondary site uses only one host name, the command does not require the *sec_host* argument. The command starts replication or the transfer of data from the primary site to the secondary site over the network.

Starting replication using full synchronization with Checkpoint

To start replication using full synchronization with Checkpoint

1. From the primary site, synchronize the RVG on the secondary site with full synchronization (using the `-c checkpoint` option):

   ```
   vradmin -g disk_group -full -c ckpt_name syncrvg pri_rvg
   sec_host
   ```

   where:

   - `disk_group` is the disk group on the primary site that VVR will replicate
   - `ckpt_name` is the name of the checkpoint on the primary site
   - `pri_rvg` is the name of the RVG on the primary site
   - `sec_host` is the virtual host name for the secondary site

   For example:

   ```
   # vradmin -g oradatadg -c rac1_ckpt syncrvg rac1_rvg
   rac_clus102_priv
   ```

2. To start replication after full synchronization, enter:

   ```
   # vradmin -g oradatadg -c rac1_ckpt startrep rac1_rvg
   rac_clus102_priv
   ```

Verifying replication status

Verify replication is functioning properly:

1. Use the `vxprint` command on the primary site:

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

2. Review the `flags` output for the status. The output may appear as `connected` and `consistent`. For example:

   ```
   # vxprint -g oradatadg -l rlk_10.182.13.221_oradatadg
   Rlink: rlk_10.182.13.221_oradatadg
   info: timeout=500 packet_size=8400 rid=0.1078
       latency_high_mark=10000 latency_low_mark=9950
       bandwidth_limit=none
   state: state=ACTIVE
       synchronous=off latencyprot=off srlprot=autodcm
   .
   .
   protocol: UDP/IP
   checkpoint: rac1_ckpt
   flags: write enabled attached consistent connected
   asynchronous
   ```

Configuring VCS to Replicate the Database Volume

After configuring both clusters for global clustering and setting up the Oracle database for replication, configure VCS to provide high availability for the database. Specifically, configure VCS agents to control the resources in the clusters, including resources for replication.

For sample main.cf files to illustrate the changes to the VCS configuration when you set up the existing Oracle database for replication, see:

"Oracle 10g configuration for CVM/VVR primary site" on page 434

"Oracle 10g configuration for CVM/VVR secondary site" on page 437

Modifying the VCS configuration for replication

Resources that must be configured or modified for replication:

- Log owner group
- RVG group
- Oracle database service group
- `RVGSharedPri` resource
- The `CVMVolDg` resource from the existing Oracle database service group.

About VCS resources for replication

Log owner group

A log owner group including the `RVGLogowner` resources must be created. The `RVGLogowner` resources are used by the RLINKs for the RVG, and the `RVGLogowner` resource, for which the RVG and its associated disk group are defined as attributes. The RVG log owner service group has an *online local firm* dependency on the service group containing the RVG.

RVG group

An RVG group including the `RVGShared` resource replication objects must be created. Define the `RVGShared` resource and `CVMVolDg` resource together within a *parallel* service group. The group is defined as parallel because it may be online at the same time on all cluster nodes. The CVMVolDg resource does not have volumes specified for the `CVMVolume` attribute; the volumes are contained in the RVG resource. The `CVMVolume` attribute for the `CVMVolDg` resource is empty because all volumes in the RVG are defined by the `RVG` attribute of the `RVGShared` resource. The `RVG` service group has an *online local firm* dependency on the `CVM` service group.

For a detailed description of the RVGLogowner agent, which VCS uses to control the `RVGLogowner` resource, and the RVGShared agent, which VCS uses to control the RVGShared resource:

See the *Veritas Cluster Server Agents for Veritas Volume Replicator Configuration Guide*.

Oracle database service group

The `RVGSharedPri` resource must be added to the existing Oracle database service group. The `CVMVolDg` resource must be removed from the existing Oracle database service group.

The existing Oracle database service group is a *parallel* group consisting of the Oracle database resource, `CVMVolDg` resource, and `CFSMount` resource (if the database resides in a cluster file system). Define the Oracle service group as a *global* group by specifying the clusters on the primary and secondary sites as values for the `ClusterList` group attribute

For a detailed description of the CVMVolDg agent in this guide:

"CVMVolDg and CFSMount resources" on page 454

For more information on replication resources:

See the *Veritas Cluster Server Agents for Veritas Volume Replicator Configuration Guide*.

Configuration before and after modification

Review the illustrations that show the changes to the VCS configuration when you set up the existing Oracle database for replication.

- Configuration before modification:
  - See Figure 17-1, "Illustration of Dependencies: Configuration Before Modification for Replication (Oracle9i)."
  - See "Illustration of Dependencies: Configuration Before Modification for Replication (Oracle 10g)" on page 297.
- Configuration after modification:
  - See "Illustration of Dependencies: Configuration After Modification for Replication (Oracle9i)" on page 298.
  - See "Illustration of Dependencies: Configuration After Modification for Replication (Oracle 10g)" on page 299.

Note that all of the dependencies between parent and child groups are *online local firm*. The CVM service group is the same in all illustrations because its definition requires no changes. For Oracle9*i*, the CVM service group may be different on your configuration, depending on the type of IP you set up for the Listener.

Figure 17-1 Illustration of Dependencies: Configuration Before Modification for Replication (Oracle9i)

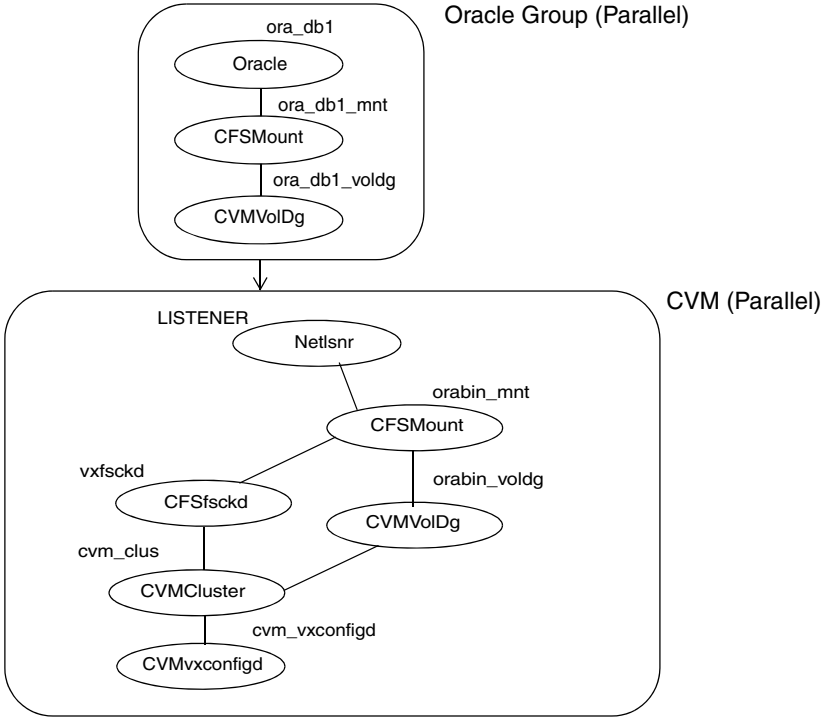


Illustration of Dependencies: Configuration Before Modification for Replication (Oracle 10g)

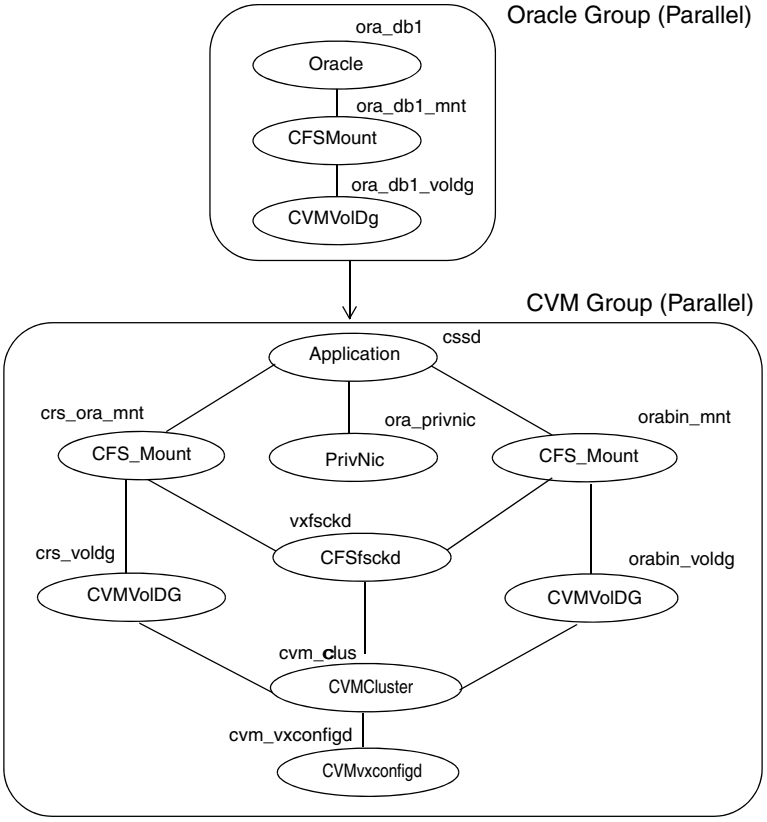


Illustration of Dependencies: Configuration After Modification for Replication (Oracle9i)

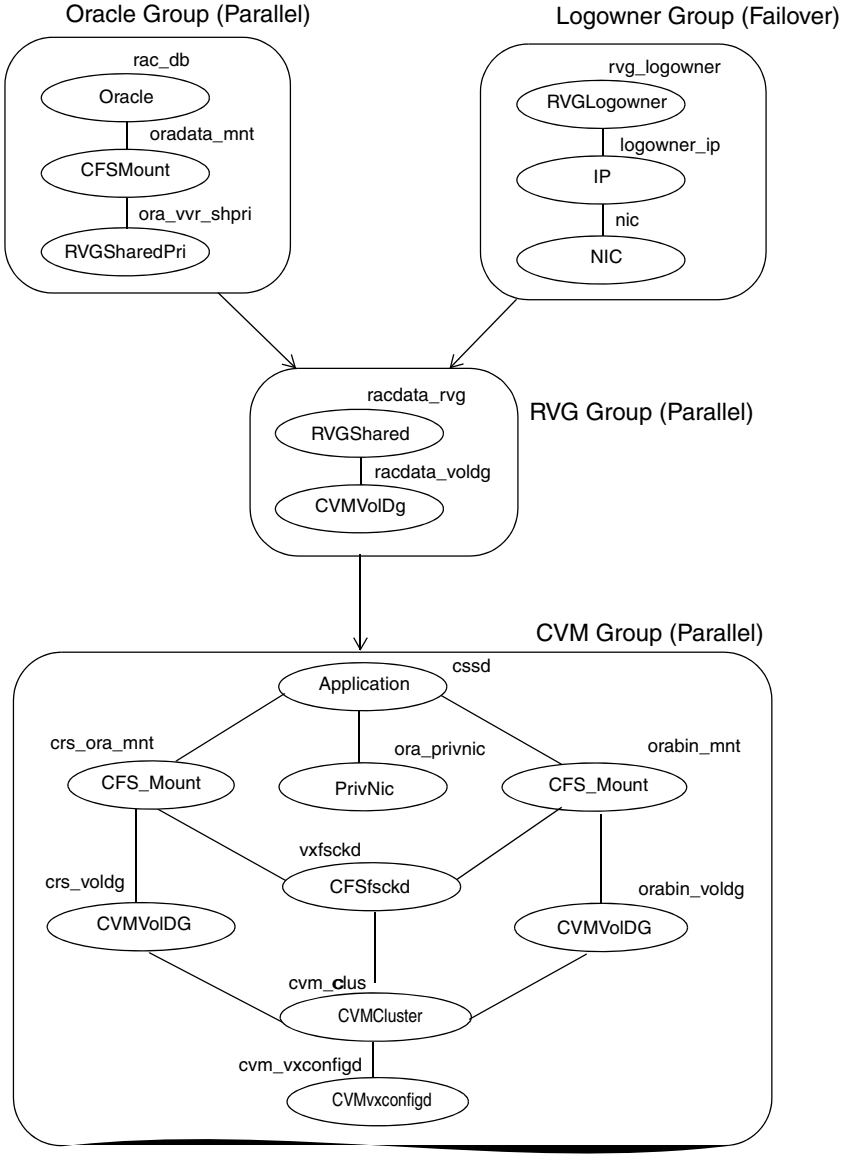
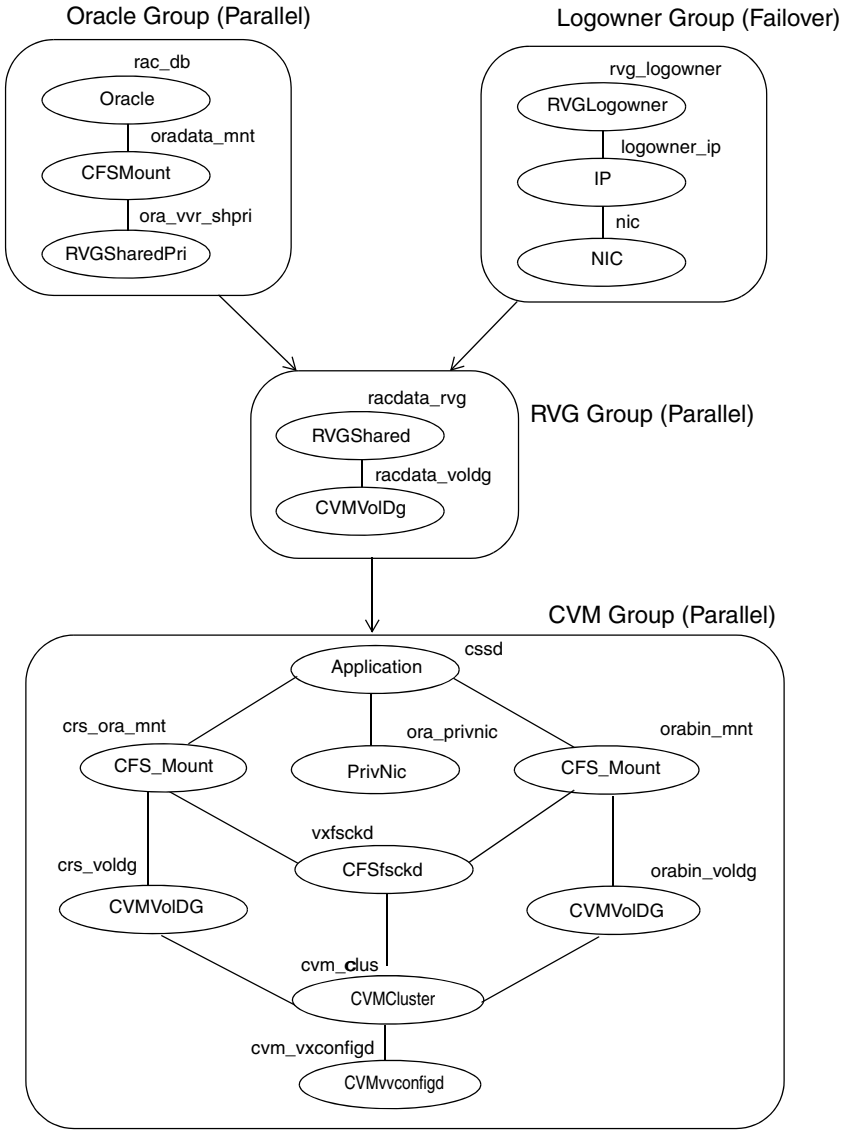


Illustration of Dependencies: Configuration After Modification for Replication (Oracle 10g)



Modifying the VCS Configuration on the Primary Site

Highlights of the procedure to modify the existing VCS configuration on the primary site include:

- Configure two service groups:
  - A log owner group including the `RVGLogowner` resource.
  - An RVG group including the `RVGShared` resource replication objects.
- Add the `RVGSharedPri` resource to the existing Oracle database service group and define this group as a global group by setting the `ClusterList` and `ClusterFailOverPolicy` attributes.
- Move the `CVMVolDg` resource from the existing Oracle database service group to the newly created `RVGShared` service group.

To modify VCS on the primary site

1. Log into one of the nodes on the primary cluster.
2. Save the existing configuration to disk and make the configuration read-only while you are make the changes:

   ```
   # haconf -dump -makero
   ```

3. Make sure VCS is not running while you edit `main.cf` by stopping the VCS engine on all nodes and leave the resources available:

   ```
   # hastop -all -force
   ```

4. Make a backup copy of the `main.cf` file:

   ```
   # cd /etc/VRTSvcs/conf/config
   # cp main.cf main.orig
   ```

5. Use `vi` or another text editor to edit the `main.cf` file.
6. Add a failover service group using the appropriate values for your cluster and nodes. Include:
   - `RVGLogowner` resource. The node on which the group is online functions as the log owner (node connected to the second cluster for the purpose of replicating data).
   - IP resource
   - `NIC` resources

   Example `RVGLogowner` service group:

   ```
   group rlogowner (
       SystemList = { galaxy = 0, nebula = 1 }
       AutoStartList = { galaxy, nebula }
       )

       IP logowner_ip (
           Device =en0
   ```

```
        Address = "10.10.9.101"
        NetMask = "255.255.240.0"
        )

    NIC nic (
        Device =en0
        NetworkType = ether
        )

    RVGLogowner logowner (
        RVG = rac1_rvg
        DiskGroup = oradatadg
        )
requires group RVGgroup online local firm
logowner requires logowner_ip
logowner_ip requires nic
```

7 Add the RVG service group using the appropriate values for your cluster and nodes.

Example `RVGgroup` service group:

```
group RVGgroup (
    SystemList = { galaxy = 0, nebula = 1 }
    Parallel = 1
    AutoStartList = { galaxy, nebula }
    )

RVGShared racdata_rvg (
    RVG = rac1_rvg
    DiskGroup = oradatadg
    )
    CVMVolDg racdata_voldg (
        CVMDiskGroup = oradatadg
        CVMActivation = sw
        )
requires group cvm online local firm
racdata_rvg requires racdata_voldg
```

8 Modify the Oracle service group using the appropriate values for your cluster and nodes:

- Define the Oracle service group as a *global* group by specifying the clusters on the primary and secondary sites as values for the `ClusterList` group attribute (see the bolded attribute in the example that follows).
- Add the `ClusterFailOverPolicy` cluster attribute. Symantec recommends using the `Manual` value. See the bolded attribute in the example.
- Add the `RVGSharedPri` resource to the group configuration.

- Remove the CVMVolDg resource, if it has been configured in your previous configuration. This resource is now part of the RVG service group.
- Specify the service group to depend (online, local, firm) on the RVG service group.

Example Oracle database service group configured for replication:

```
group oradb1_grp
    SystemList = { galaxy = 0, nebula = 1 }
    ClusterList = { rac_cluster101 = 0, rac_cluster102 = 1 }
    Parallel = 1
    ClusterFailOverPolicy = Manual
    Authority = 1
    AutoStartList = { galaxy, nebula }
    )

    CFSMount oradata_mnt
        MountPoint = "/oradata"
        BlockDevice = "/dev/vx/dsk/oradatadg/racdb_vol"
        )

    RVGSharedPri ora_vvr_shpri (
        RvgResourceName = racdata_rvg
        OnlineRetryLimit = 0
        )


Oracle rac_db (
    Sid @galaxy = vrts1
    Sid @nebula = vrts2
    Owner = Oracle
    Home = "/oracle/orahome/dbs"
    Pfile @galaxy = "/oracle/orahome/dbs/initvrts1.ora"
    Pfile @nebula = "/oracle/orahome/dbs/initvrts2.ora"
    ShutDownOpt = SRVCTLSTOP
    MonScript = "./bin/Oracle/SqlTest.pl"
    )
requires group RVGgroup online local firm
oradata_mnt requires ora_vvr_shpri
rac_db requires oradata_mnt
```

9 Save and close the `main.cf` file.

10 Verify the syntax of the `/etc/VRTSvcs/conf/config/main.cf` file:

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

Modifying the VCS Configuration on the Secondary Site

Highlights of the procedure to modify the existing VCS configuration on the secondary site include:

- Add the log owner and RVG service groups.

- Add a service group to manage the Oracle database and the supporting resources.
- Define the replication objects and agents, such that the cluster at the secondary site can function as a companion to the primary cluster.

The steps are similar to what you performed on the primary site.

To modify VCS on the secondary site

1. Log into one of the nodes on the secondary site as `root`.
2. Save the existing configuration to disk and make the configuration read-only while you are make the changes:

   ```
   # haconf -dump -makero
   ```

3. Ensure VCS is not running while you edit the `main.cf` by stopping the VCS engine on all systems and leave the resources available:

   ```
   # hastop -all -force
   ```

4. Make a backup copy of the `main.cf` file:

   ```
   # cd /etc/VRTSvcs/conf/config
   # cp main.cf main.orig
   ```

5. Use `vi` or another text editor to edit the `main.cf` file.
6. Edit the CVM group on the secondary site. After the SF Oracle RAC installation, the CVM file resembles the example in: Appendix A, "Sample VCS configuration files for SF Oracle RAC" on page 425.

   In our example, the secondary site has `rac_cluster102` consisting of the nodes `mercury` and `jupiter`. To modify the CVM service group on the secondary site, use the CVM group on the primary site as your guide.
7. Add a failover service group using the appropriate values for your cluster and nodes. Include:
   - `RVGLogowner` resource. The node on which the group is online functions as the log owner (node connected to the second cluster for the purpose of replicating data).
   - IP resource
   - `NIC` resources

   Example `RVGLogowner` service group:

   ```
   group rlogowner (
          SystemList = { galaxy = 0, nebula = 1 }
          AutoStartList = { galaxy, nebula }
          )

          IP logowner_ip (
                  Device =en0
   Address = "10.11.9.102"
                  NetMask = "255.255.240.0"
   ```

```
                )

        NIC nic (
                Device =en0
NetworkType = ether
                )

        RVGLogowner logowner (
                RVG = rac1_rvg
                DiskGroup = oradatadg
                )

requires group RVGgroup online local firm
logowner requires logowner_ip
logowner_ip requires nic
```

8 Add the RVG service group using the appropriate values for your cluster and nodes.

Example `RVGgroup` service group:

```
group RVGgroup (
        SystemList = { mercury = 0, jupiter = 1 }
        Parallel = 1
        AutoStartList = { mercury, jupiter }
        )

RVGShared racdata_rvg (
               RVG = rac1_rvg
               DiskGroup = oradatadg
               )

        CVMVolDg racdata_voldg
            CVMDiskGroup = oradatadg
            CVMActivation = sw
            )
requires group cvm online local firm
racdata_rvg requires racdata_voldg
```

9 Add an Oracle service group. Use the Oracle service group on the primary site as a model for the Oracle service group on the secondary site.

- Define the Oracle service group as a *global* group by specifying the clusters on the primary and secondary sites as values for the `ClusterList` group attribute.
- Assign this global group the same name as the group on the primary site; for example, `oradb1_grp`.
- Include the `ClusterList` and `ClusterFailOverPolicy` cluster attributes. Symantec recommends using the `Manual` value.
- Add the `RVGSharedPri` resource to the group configuration.

- Remove the CVMVolDg resource, if it has been configured in your previous configuration. This resource is now part of the RVG service group.
- Specify the service group to depend (online, local, firm) on the RVG service group.

Below is an example of the Oracle group on the secondary site:

```
.
group oradb1_grp
    SystemList = { mercury = 0, jupiter = 1 }
    ClusterList = { rac_cluster102 = 0, rac_cluster101 = 1 }
    Parallel = 1
    OnlineRegryInterval = 300
    ClusterFailOverPolicy = Manual
    Authority = 1
    AutoStartList = { mercury, jupiter }
    )

        CFSMount oradata_mnt
        MountPoint = "/oradata"
        BlockDevice = "/dev/vx/dsk/oradatadg/racdb_vol"
        )

RVGSharedPri ora_vvr_shpri (
        RvgResourceName = racdata_rvg
        OnlineRetryLimit = 0
        )

    Oracle rac_db (
        Sid @mercury = vrts1
        Sid @jupiter = vrts2
        Owner = Oracle
        Home = "/oracle/orahome/dbs"
        Pfile @mercury = "/oracle/orahome/dbs/initvrts1.ora"
        Pfile @jupiter = "/oracle/orahome/dbs/initvrts2.ora"
        StartUpOpt = SRVCTLSTART
        ShutDownOpt = SRVCTLSTOP
        MonScript = "./bin/Oracle/SqlTest.pl"
        )

requires group RVGgroup online local firm
oradata_mnt requires ora_vvr_shpri
rac_db requires oradata_mnt
```

10 Save and close the `main.cf` file.

11 Verify the syntax of the `/etc/VRTSvcs/conf/config/main.cf` file:

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

Starting VCS on All Nodes in Both Clusters

Start VCS on both clusters:

1. From the primary site, start the VCS engine on one node:

   ```
   # hastart
   ```

2. Type:

   ```
   # hastatus
   ```

3. When `LOCAL_BUILD` or `RUNNING` is listed in the message column, start VCS on the other node:

   ```
   # hastart
   ```

4. Verify that VCS brings all resources online. On one node, enter:

   ```
   # hagrp -display
   ```

   The `Oracle`, `RVG`, and `CVM` groups are online on both nodes of the primary site. The `RVGLogOwner` group is online on one node of the cluster. If either the `RVG` group or the `RVGLogOwner` group is partially online, manually bring the groups online using the `hagrp -online` command. This information applies to the secondary site, except for the Oracle group which must be offline.

5. On the secondary site, start VCS from one node:

   ```
   # hastart
   ```

6. Type:

   ```
   # hastatus
   ```

7. When `LOCAL_BUILD` or `RUNNING` is listed in the message column, start VCS on the other node:

   ```
   # hastart
   ```

8. Verify the service groups and their resources that are brought online. On one node, enter:

   ```
   # hagrp -display
   ```

   The `Oracle` service group is offline on the secondary site, but the `CVM`, `RVG log owner`, and `RVG` groups are online.

Migration and takeover of primary replication role

Migration refers to the planned transfer of the role of primary replication host from one cluster to a remote cluster. This transfer enables the application on the remote cluster to actively use the replicated data. The former primary cluster becomes free for maintenance or other activity.

Takeover occurs when an unplanned event (such as a disaster) causes a failure, making it necessary for the applications using the replicated data to be brought online on the remote cluster.

Migrating the role of primary site to the remote site

After configuring the replication objects within VCS, you can use VCS commands to migrate the role of the cluster on the primary site to the remote cluster. In the procedure below, VCS takes the replicated Oracle RAC database service group, `oradb1_grp`, offline on the primary site and brings it online on the secondary site; the secondary site now assumes the role of the primary site.

Note: The `hagrp -switch` command cannot migrate a parallel group within a cluster or between clusters in a global cluster environment.

---

1. From the primary site, take the Oracle service group offline on all nodes.

   ```
   # hagrp -offline oradb1_grp -any
   ```

   Wait for VCS to take all Oracle service groups offline on the primary site.

2. Verify that the RLINK between the primary and secondary is up to date. Use the `vxrlink -g` command with the `status` option and specify the RLINK for the primary cluster (`rlk_rac_clus102_priv_rac1_rvg`, in this example). You can use the command from any node on the primary cluster. For example:

   ```
   # vxrlink -g oradatadg status rlk_rac_clus102_priv_rac1_rvg
   ```

3. On Secondary make sure that CRS is up, add listener resource using `netca`. Make changes to `tnsnames.ora`.

   Example `tnsnames.ora` (Here vrts is the database name)

   ```
   LISTENERS_VRTS =
     (ADDRESS_LIST =
       (ADDRESS = (PROTOCOL = TCP)(HOST = mercury-vip)(PORT =
   1521))
       (ADDRESS = (PROTOCOL = TCP)(HOST = jupiter-vip)(PORT =
   1521))
     )

   VRTS2 =
     (DESCRIPTION =
       (ADDRESS = (PROTOCOL = TCP)(HOST = jupiter-vip)(PORT =
   1521))
       (CONNECT_DATA =
         (SERVER = DEDICATED)
         (SERVICE_NAME = vrts)
         (INSTANCE_NAME = vrts2)
       )
     )

   VRTS1 =
     (DESCRIPTION =
       (ADDRESS = (PROTOCOL = TCP)(HOST = mercury-vip)(PORT =
   1521))
   ```

```
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = vrts)
      (INSTANCE_NAME = vrts1)
    )
  )

VRTS =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = mercury-vip)(PORT =
1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = jupiter-vip)(PORT =
1521))
    (LOAD_BALANCE = yes)
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = vrts)
    )
  )
```

4 Register the database using `srvctl` command. (On Secondary)

```
# srvctl add database -d <database_name> -o <oracle_home> -p
<spfile-on-shareddisk>
```

To prevent automatic database instance restart, change Management policy for the database (automatic, manual) to MANUAL using `srvctl` command:

```
# srvctl add database -d <db-name> -p
<location-of-parameter-file> -y manual
```

5 Register the instances using `srvctl` command. Execute the following command on each node:

```
# srvctl add instance -d <database_name> -i <instance_name> -n
<node-name>
```

6 Create directories `adump`, `bdump`, `cdump`, `dpdump`, `hdump`, udump in `$ORACLE_HOME/admin/<db_name>`.

7 Create pfile on each node as follows:

```
# echo "SPFILE=<location of spfile on shared volume>"
>>$ORACLE_HOME/dbs/init<db_name><instance-number>.ora
```

8 On the secondary site, bring the Oracle service group online on all nodes:

```
# hagrp -online oradb1_grp -any
```

9 Verify CRS resources by executing the command `crs_stat -t`. All resources should be online.

10 Make sure that All CRS resources are online, and switch back the group *OracleRAC* to primary. Use the following command to switch back the oradb1_grp to original primary site:

```
# hagrp -offline oradb1_grp -any
```

11 Make sure that "oradb1_grp" is offline and execute the following command on Original Primary cluster to bring the "oradb1_grp" online.

```
#hagrp -online oradb1_grp -any
```

Taking over the primary role by the remote cluster

Takeover occurs when the remote cluster on the secondary site starts the application that uses replicated data. This situation may occur if the secondary site perceives the primary site as dead, or when the primary site becomes inaccessible (perhaps for a known reason). See the *Veritas Volume Replicator Administrator's Guide* for detailed description of concepts of taking over the primary role.

Before enabling the secondary site to take over the primary role, the administrator on the secondary site must "declare" the type of failure at the remote (primary, in this case) site. Designate the failure type using one of the options for the `haclus` command, are discussed in the following sections.

disaster

When the cluster on the primary site is inaccessible and appears dead, the administrator declares the failure type as *disaster*. For example, fire may destroy a data center, including the primary site and all data in the volumes. After making this declaration, the administrator can bring the service group online on the secondary site, which now has the role as "primary" site.

outage

When the administrator of a secondary site knows the primary site is inaccessible for a known reason, such as a temporary power outage, the administrator may declare the failure as an *outage*. Typically, an administrator expects the primary site to return to its original state.

After the declaration for an outage occurs, the RVGSharedPri agent enables DCM logging while the secondary site maintains the primary replication role. After the original primary site becomes alive and returns to its original state, DCM logging makes it possible to use fast fail back resynchronization when data is resynchronized to the original cluster.

Before attempting to resynchronize the data using the fast fail back option from the current primary site to the original primary site, take the precaution at the original primary site to make a snapshot of the original data. This action provides a valid copy of data (see "replica" on page 310) at the original primary site for use in the case the current primary site fails before the resynchronization is complete.\

disconnect

When both clusters are functioning properly and the heartbeat link between the clusters fails, a split-brain condition exists. In this case, the administrator can declare the failure as *disconnect*, meaning no attempt will occur to take over the role of the primary site at the secondary site. This declaration is merely advisory, generating a message in the VCS log indicating the failure results from a network outage rather than a server outage.

replica

In the rare case where the current primary site becomes inaccessible while data is resynchronized from that site to the original primary site using the fast fail back method, the administrator at the original primary site may resort to using a data snapshot (if it exists) taken before the start of the fast fail back operation. In this case, the failure type is designated as *replica*.

Example of takeover for an outage

To take over after an outage

1. From any node of the secondary site, issue the `haclus` command:

   ```
   # haclus -declare outage -clus rac_cluster101
   ```

2. After declaring the state of the remote cluster, bring the Oracle service group online on the secondary site. For example:

   ```
   # hagrp -online -force oradb1_grp -any
   ```

Example of resynchronization after an outage

To resynchronize after an outage

1. On the original primary site, create a snapshot of the RVG before resynchronizing it in case the current primary site fails during the resynchronization. Assuming the disk group is `oradatadg` and the RVG is `rac1_rvg`, type:

   ```
   # vxrvg -g oradatadg -F snapshot rac1_rvg
   ```

   See the *Veritas Volume Replicator Administrator's Guide* for details on RVG snapshots.

2. Resynchronize the RVG. From the CVM master node of the current primary site, issue the `hares` command and the `-action` option with the `fbsync` action token to resynchronize the `RVGSharedPri` resource. For example:

   ```
   # hares -action ora_vvr_shpri fbsync -sys mercury
   ```

   To determine which node is the CVM master node, type:

   ```
   # vxdctl -c mode
   ```

3 Perform one of the following commands, depending on whether the resynchronization of data from the current primary site to the original primary site is successful:

   a If the resynchronization of data is successful, use the `vxrvg` command with the `snapback` option to reattach the snapshot volumes on the original primary site to the original volumes in the specified RVG:

```
# vxrvg -g oradatadg snapback rac1_rvg
```

   b A failed attempt at the resynchronization of data (for example, a disaster hits the primary RVG when resynchronization is in progress) could generate inconsistent data. You can restore the contents of the RVG data volumes from the snapshot taken in step 1:

```
# vxrvg -g oradatadg snaprestore rac1_rvg
```

Section VI

Backup and recovery

Use Checkpoints and FlashSnap procedures to back up your RAC database.

Chapter 18

Configuring the repository database for Oracle

After installing SF Oracle RAC, you can create and configure the repository database using the sfua_db_config script.This repository database configuration enables you to use SF Oracle RAC features such as Checkpoint, flashsnap and storage mapping. The script detects that the system is running in an HA configuration and automatically configures the repository database.

Creating and configuring the repository database for Oracle

Before running the sfua_db_config script, review the following requirements:

- Make sure a disk group exists with at least one volume, which should not be shared. A VxFS file system must be created on the disk group.
- The volume must be started and the file system must be mounted.
- Obtain an unique virtual IP address for public NIC interface.
- Obtain a device name for the public NIC interface (for example : en0).
- Obtain a subnet mask for the public NIC interface.

Note: The volume is used to store the repository database.

Table 2-1 indicates the options available for the sfua_db_config script.

Table 18-1 sfua_db_config options

Options	Description
`-ssh`	Use this option in a high availability (HA) configuration. The option indicates that ssh and scp are to be used for communication between systems. 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	Drops the repository database.
-o unconfig_cluster	Use this option in a high availability (HA) configuration. Unconfigures the repository database from the VCS cluster.
-o dbstatus	Verifies the status of the database and database server.

Table 18-2 sfua_db_config options

Option	Description
-o stopserver	Stops the database server.
-o startserver	Starts the database server.
-o serverstatus	Reports the databas server status.
-o stopdb	Detaches the repository database from the database server.
-o startdb	Attaches the repository database to the database server.

To create and configure the repository database

1 Run the sfua_db_config script:

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

2 The following is an example of configuring 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 NIC for system galaxy for HA Repository
configuration:en0

Enter the NIC for system nebula for HA Repository
configuration:en0

Enter the Virtual IP address for repository
failover:10.182.186.249
Enter the netmask for public NIC interface:255.255.0.0

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-en0 nebula-en0
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 sucessfully 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.

Setting administrative permissions

To allow database administrators to administer a database using SF Oracle RAC, you must change permission settings. During SF Oracel RAC installation, you are asked if you want to allow database administrators access. If you did not change permissions installation, you can do so at a later time.

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

To enable access for users other than root

1 To enable the user “oracle” access to the /opt/VRTSdbed directory, use the `chown` and `chmod` commands, as follows:

```
# chown -R oracle /opt/VRTSdbed
```

```
# chmod -R 500 /opt/VRTSdbed
```

2 To allow users in the group “dba” access to the /opt/VRTSdbed directory, use the `chgrp` and `chmod` commands, as follows:

```
# chgrp -R dba /opt/VRTSdbed
# chmod -R 550 /opt/VRTSdbed
```

Chapter 19

Using Checkpoints and Storage Rollback with Storage Foundation for Oracle RAC

Note: Storage Foundation for Oracle RAC only supports the SFDB features described in this guide.

Veritas Storage Checkpoint enables 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. This chapter describes Storage Checkpoints and Storage Rollback and how to use them through Storage Foundation for Oracle RAC.

Topics covered in this chapter include:

- "Backing up and recovering the database using Storage Checkpoints" on page 324
- "Cloning the Oracle instance using dbed_clonedb" on page 328
- "Guidelines for Oracle recovery" on page 331

Storage Checkpoints and Storage Rollback concepts

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. This chapter describes what Storage Checkpoints and storage rollback are and how to make use of these technologies through SF ORacle RAC.

SF ORacle RAC provides a Storage Checkpoint facility is similar to the snapshot file system mechanism; however, a Storage Checkpoint persists after a system reboot. 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 Veritas Storage Foundation command line interface (CLI).

A direct application of the Storage Checkpoint facility is Storage Rollback. 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 restoring the database faster. For more information on Storage Checkpoints and Storage Rollback, see the *Veritas File System Administrator's Guide*.

How Storage Checkpoints and Storage Rollback work

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. 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—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 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 *n* 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.

You can set a quota to limit how much space a file system will give to all storage checkpoints, to prevent the checkpoints from consuming all free space. See the command `dbed_ckptquota` for more information.

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.

Note: 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`.

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.

If you mount a Storage Checkpoint as read-write, roll back to this Storage Checkpoint will not be permitted. 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 enables the database to still be rolled back to the original Storage Checkpoint.

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).

When you use VERITAS NetBackup to back up your database, VERITAS NetBackup creates one set of Storage Checkpoints to provide a consistent view of the file systems for the database backups. The space required to hold this additional set of Storage Checkpoints depends on how busy the database load is when the backup is running. If the database is offline during the entire backup window, there is no additional space required.

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 (say, 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.

See the `fsadm_vxfs`(1) and `chfs`(1) manual pages for more information.

Performance of Storage Checkpoints

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 like 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

Storage Checkpoints can be created by specifying one of the following options: online, offline, or instant. To create a Storage Checkpoint with the online option, the database should be online and you must enable ARCHIVELOG mode for the database. 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.

For the instant option, the database should be online and it can be running in either ARCHIVELOG or NOARCHIVELOG mode. You can only roll back the entire database to an instant Storage Checkpoint. Rolling back individual tablespaces or datafiles to an instant Storage Checkpoint is not possible. After the rollback is complete, you need to perform database recovery. Rolling the database forward is not supported; that is, you cannot apply archived redo logs.

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 using the command line

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 using the procedure below.

Usage Notes

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

To verify that a Storage Checkpoint is error-free using the command line

1 Create and mount a Storage Checkpoint:

```
$ /opt/VRTS/bin/dbed_ckptcreate-S PROD -H
/oracle/product -o online
Creating online Storage Checkpoint of database PROD.
Storage Checkpoint Checkpoint_903937870 created.

$ mkdir /tmp/ckpt_ro

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

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

---

2 Examine the content of the Storage Checkpoint:

```
$ ls -l /tmp/ckpt_ro/dbvol_82/dbinst1
drwxr-xr-x                3       oracle             dba
1024                      Nov 11 2000 .
drwxr-xr-x                3       oracle  dba        512
Nov 16 11:00 ..
-rw-r--r--                1       oracle             dba
209747968                 Nov 16 10:58 .tstmp
-rw-r--r--                1       oracle  dba
209747968                 Nov 16 10:58 .tstab
lrwxrwxrwx                1       oracle  dba        18
Nov 11 2000 tstmp -> \

.tstmp::cdev:vxfs:
lrwxrwxrwx                1       oracle     dba
18                        Nov 11 2000 tstab -> \

.tstab::cdev:vxfs:
```

3 Run `dbv` tool against Quick I/O file `tstmp`:

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. 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.

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.

To back up a frozen database image using the command line

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

---

1. Create a Storage Checkpoint using the `dbed_ckptcreate` command:

```
$ /opt/VRTS/bin/dbed_ckptcreate -S PROD1 -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 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. 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
   # **hagrp -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.
4. Start up the database instance if it is down.
5. Unfreeze the service group
   # **hagrp -unfreeze *Service_Group***
6. Use **recover database until cancel**, **recover database until change**, or **recover database until time** to re-apply archive logs to the point before the table was deleted to recover the database to 4:00 p.m.
7. Open the database with **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.

Cloning the Oracle instance using dbed_clonedb

You can use the `dbed_clonedb` command to clone an Oracle instance using mountable and writable Storage Checkpoints to the same or different instance so the instance can coexist. You can also create a clone instance using a Storage Checkpoint that is not mounted.

You have the option to manually or automatically recover the Oracle database when using the `dbed_clonedb` command:

- Manual (interactive) recovery, which requires using the `-i` option, of the clone instance 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 `dbed_clonedb` command, recovers the entire database and replays all of the archive logs. You will not be prompted for any archive log names.

Caution: If the `-o cio` option was used with the `mount` command to mount your primary database file systems, Concurrent I/O will not be preserved when using `db2ed_clonedb` to create a clone database.

Prerequisites

- You may be logged in as either the database administrator or `root`.
- Make sure you have enough space to create a clone database on your system. A clone database takes up as much memory and machine resources as the primary database.
- You must first create a writable Storage Checkpoint. See the *Veritas Storage Foundation for Oracle Administration Guide*.
- If you choose to use an existing Storage Checkpoint to create the clone database, the Storage Checkpoint needs to be online.

Usage notes

- The `dbed_clonedb` command is used to create a copy of an Oracle database, cloning all existing database files to new locations. This is required when using mountable, writable Storage Checkpoints, where a new Oracle database needs to be started on the same host as an existing database.
- The utility requires that the current environment be configured correctly for the existing Oracle database which has had a Storage Checkpoint created underneath it. This means that the ORACLE_SID and ORACLE_HOME environment variables must be set correctly.

- the dbed_clonedb command cannot use instant checkpoint to clone a RAC database.
- When cloning an Oracle instance using the dbed_clonedb or dbed_vmclonedb command, the clone database's ORACLE_SID can only be eight characters or less. You will receive an error (ERROR V-81-5713) if the ORACLE_SID is more than eight characters.
- It is assumed that the user has a basic understanding of the Oracle recovery process.
- See the dbed_clonedb(1M) manual page for more information.

Options

-S Specifies the name of the new Oracle SID, which will be the name of the new database instance.

-m Indicates the new mount point of the Storage Checkpoint.

-c Indicates the name of the Storage Checkpoint.

-i Runs the command in interactive mode where you must respond to prompts by the system. The default mode is non-interactive. (Optional)

-d This option is only for use 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.

-o The -o umount option shuts down the clone database and unmounts the Storage Checkpoint file system. The -o restartdb option mounts the Storage Checkpoint file system and starts the clone database.

-p The pfile_modification_file option

To clone an Oracle instance with manual Oracle recovery

1 Use the dbed_clonedb command as follows:

```
$ /opt/VRTS/bin/dbed_clonedb -S NEW9 -m /local/oracle9/1 \
    -c Checkpoint_988813047 -i
```

Output resembles the following:

```
Primary Oracle SID is TEST9i
New Oracle SID is NEW9
Checkpoint_988813047 not mounted at /local/oracle9/1
Mounting Checkpoint_988813047 at /local/oracle9/1
 Using environment-specified parameter file
        /local/oracle9/links/dbs/initTEST9i.ora
 Default Oracle parameter file found:
        /local/oracle9/links/dbs/initTEST9i.ora
 Copying /local/oracle9/links/dbs/initTEST9i.ora
        to /local/oracle9/1/testvol
```

```
     Control file 'ora_control2'
           path not explicitly specified in init file; assuming
           ORACLE_HOME/dbs

     All redo-log files found
     Copying initTEST9i.ora to initNEW9.ora
           in /local/oracle9/1/testvol
     Altering db_name in initNEW9.ora
     Altering control file locations in initNEW9.ora
     Creating new link for clone database init file
     Creating archive log directory

     About to start up new database and begin reconfiguration

     Database NEW9 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/oracle9/1/testvol
2. ORACLE_SID=NEW9; export ORACLE_SID # sh and ksh, OR
       setenv ORACLE_SID NEW9 #csh
3. /local/oracle9/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 NEW9
     Database instance NEW9 is up and running
```

To clone an Oracle instance with automatic Oracle recovery

1 Use the `dbed_clonedb` command as follows:

```
$ /opt/VRTS/bin/dbed_clonedb -S NEW9 -m /local/oracle9/1 \
    -c Checkpoint_988813047
```

Output resembles the following:

```
     Primary Oracle SID is TEST9i
     New Oracle SID is NEW9
     Checkpoint_988813047 not mounted at /local/oracle9/1
     Mounting Checkpoint_988813047 at /local/oracle9/1
     Using environment-specified parameter file
           /local/oracle9/links/dbs/initTEST9i.ora
     Default Oracle parameter file found:
```

```
      /local/oracle9/links/dbs/initTEST9i.ora
Copying /local/oracle9/links/dbs/initTEST9i.ora
      to /local/oracle9/1/testvol
Control file 'ora_control2'
      path not explicitly specified in init file; assuming
      ORACLE_HOME/dbs

All redo-log files found
Copying initTEST9i.ora to initNEW9.ora
      in /local/oracle9/1/testvol
Altering db_name in initNEW9.ora
Altering control file locations in initNEW9.ora
Creating new link for clone database init file
Creating archive log directory

About to start up new database and begin reconfiguration
Database NEW9 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 NEW9
Database instance NEW9 is up and running
```

Guidelines for Oracle recovery

For optimal Oracle recovery, follow these guidelines:

- Back up all control files before Storage Rollback 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. It is also a good idea to back up the control files before and after making structural changes to databases.

Note: The SFDB utility automatically saves control file and log information when you create a Storage Checkpoint. See [[find xref for creating a Storage Checkpoint]] for more information.

- Make sure 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.

Note: If you intend to roll back the database to recover from structural changes that you do not want to maintain, you may want to roll back control files. The SFDB utility saves control file and log information and provides the capability to roll back control files. See [[find xrefs for Managing Storage Rollback and Showing Backup Control File List - old VxDBQA chapter?]] and for more information.

- Make sure 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.
  For 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.

- To perform a complete media recovery:

```
SET AUTORECOVERY ON;
RECOVER DATABASE;
```

- To perform an incomplete media recovery, use one of the following:
  - **`RECOVER DATABASE UNTIL CANCEL;`**
  - **`RECOVER DATABASE UNTIL TIME 'yyyy-mm-dd:hh:mm:ss';`**
    (You can confirm the time of error by checking the `$ORACLE_HOME/rdbms/log/alert*.log` file.)

- **RECOVER DATABASE UNTIL TIME 'yyyy-mm-dd:hh:mm:ss' using**
  backup controlfile;
- **RECOVER DATABASE UNTIL CHANGE scn;**

- To open the database after an incomplete media recovery, use the following:
  - **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.

Caution: 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.

Chapter 20

Using database FlashSnap for backup and off-host processing

This chapter describes how to use Database FlashSnap to create a point-in-time copy of a database for backup and off-host processing. Database FlashSnap enables you to make backup copies of your volumes online with minimal interruption to users.

Topics covered in this chapter include:

About Database FlashSnap

Database FlashSnap lets you capture an online image of an actively changing database at a given instant. You can perform backups and off-host processing tasks on snapshots while providing continuous availability of your critical data.

Execute Database FlashSnap commands from the command line.

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.

Database FlashSnap enables database administrators to create a consistent copy of a database without root privileges by creating 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 can significantly reduce the time it takes to backup your database, increase the availability of your production database, and still maintain your production database's performance.

Note: You must have Veritas Storage Foundation Enterprise Edition on all systems on which you intend to use Database FlashSnap.

To use Database FlashSnap, you must first configure the volumes used by the database.

See "Preparing hosts and storage for Database FlashSnap" on page 341.

Solving typical database problems with Database FlashSnap

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 not central to their job function or 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 it does not require root privileges, Database FlashSnap overcomes these obstacles by enabling database administrators to create consistent snapshots of the database more easily. The snapshots can be utilized for repetitive use.

About Database FlashSnap applications

The following are typical applications of Database FlashSnap:

- *Database Backup and Restore*: Enterprises require 24/7 online data availability. They 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

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 341.

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 their snapshot requirements. For example, they need 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 a file called a *snapplan*. The snapplan specifies snapshot options that will be used when creating a snapshot image (such as whether the snapshot mode will be `online`, `offline`, or `instant`).

See "Creating a snapplan (dbed_vmchecksnap)" on page 359.

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.

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.

Database FlashSnap can also be used to recover the primary copy of the database if it becomes corrupted by overwriting it with the snapshot. You can recover the primary database with a snapshot using the reverse resynchronization functionality of Database FlashSnap.

Using Database FlashSnap commands

The Database FlashSnap feature consists of three commands:

- `dbed_vmchecksnap` (used on the primary host)
  Creates and validates the snapshot plan used to create a snapshot image of an Oracle database. You can also use `dbed_vmchecksnap` to copy, list, or remove a snapplan or make sure the storage is configured properly for the task. `dbed_vmchecksnap` is also used on the secondary host to list the snapplan.
- `dbed_vmsnap` (used on the primary host)
  Creates a snapshot image of an Oracle database by splitting the mirror volumes used by the database. You can also use `dbed_vmsnap` to resynchronize snapshot volumes with their original volumes. The command also allows you to resynchronize the original volumes from the data in the snapshot volumes, which is useful if the original volumes become corrupted. Resynchronizing the original volumes from the snapshot volumes is known as *reverse resynchronization*.
- `dbed_vmclonedb` (used on the primary or secondary host)
  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. The snapshot image can be brought up on the same host running the primary database or on a secondary host.

All of these commands can be executed by the Oracle database administrator and do not require superuser (`root`) privileges.

Using Database FlashSnap options

Database FlashSnap offers three options for creating database snapshots. The option you choose is specified in the snapplan.

- *online*
  With this option, the tablespaces are put into online backup mode before the snapshot is created. This type of snapshot is also a valid database backup. Select this option if you are performing a point-in-time recovery from logical errors.
- *instant*
  With this option, the database can be up and running, and the tablespaces do not need to be put into online backup mode before the snapshot is created. However, all the file systems used by the database (including those containing the online redo logs and control files) are temporarily frozen and the cache is flushed before the snapshot is created. By freezing the file

systems, the snapshot will be a consistent point-in-time image of the database from which a database clone can be created.

An instant snapshot can be used to guard against data corruption or for off-host decision-support queries. However, it *is not* a valid database backup and cannot be used to perform a point-in-time recovery or off-host backup since tablespaces are not put into online backup mode before the snapshot is created. The instant option is much faster than the online option.

- *offline*

  The `offline` option can be used to clone or back up a database. With this option, the database must be shut down when the snapshot is created. and online redo logs are required. This type of snapshot is a valid database backup.

Note: 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

Before using Database FlashSnap, you must first determine your intended application. You will then need to make the following decisions:

- Which snapshot mode is appropriate: online, offline, or instant?
- Will you need one or two hosts?

Selecting the snapshot mode

If your purpose is to use the snapshot for backup or to recover the database after logical errors have occurred, choose the online option. In the event that your production database is offline, choose offline. If you intend to use the snapshot for decision-support analysis, reporting, development, or testing, choose instant. An instant snapshot is not suitable for recovery because it is not necessarily an exact copy of the primary database.

Selecting one or two hosts

If maintaining the performance of your primary database is critical, you can offload processing of the snapshots to a secondary host. For off-host processing, storage must be shared between the primary and secondary hosts.

If cost savings is most important, you can choose to do the processing on the same host as the primary database to save on hardware costs.

Preparing hosts and storage for Database FlashSnap

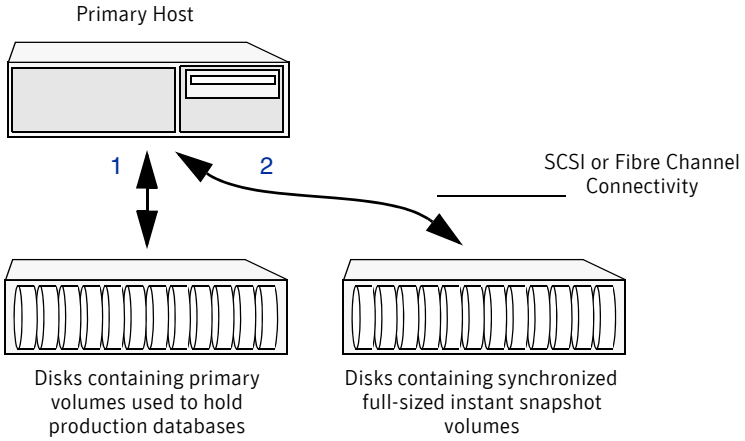
Setting up hosts

Database FlashSnap requires sufficient Veritas Volume Manager disk space, and can be used on the same host that the database resides on (the primary host) or on a secondary host. 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.

Single-host configuration

Figure 20-1 on page 341shows the suggested arrangement for implementing Database FlashSnap solutions on the primary host to avoid disk contention.

Figure 20-1 Example of a Database FlashSnap solution on a primary host

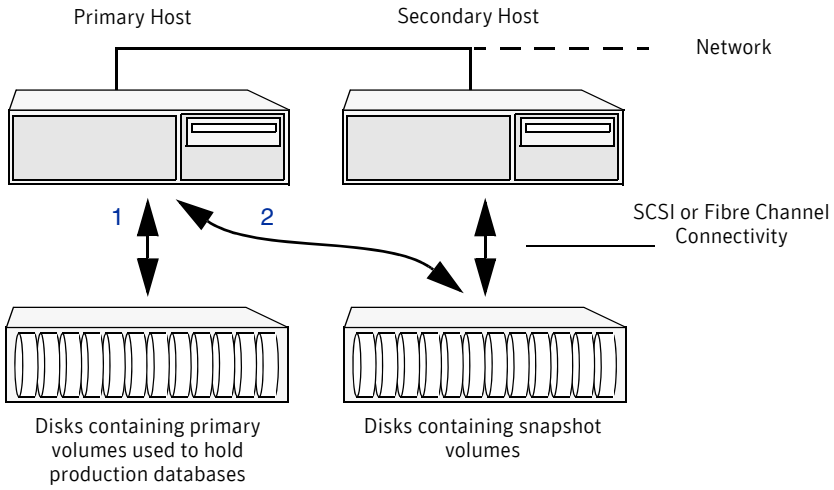


Two-host configuration

As shown in Figure 20-2 on page 342, a Database FlashSnap configuration with two hosts 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. A two-host configuration also allows the snapshot database to avoid contending for I/O resources on the primary host.

For off-host processing applications, both the primary and secondary hosts need to share the storage in which the snapshot database is created. Both the primary and secondary hosts must be able to access the disks containing the snapshot volumes.

Figure 20-2 Example of an off-host Database FlashSnap solution



Host and storage requirements

Before using Database FlashSnap, ensure that:

- 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.

In addition, before attempting to use Database FlashSnap with two hosts, ensure that:

- The versions of Veritas Storage Foundation on the primary and secondary hosts are the same.
- The same version of Oracle is installed on both hosts the Oracle binaries and datafiles are on different volumes and disks.
- The UNIX login for the database user and group must be the same on both hosts.
- You have a Veritas Storage Foundation Enterprise Edition license on both hosts.

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` CLI 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 `vxdg`(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.

Caution: The procedure given in this section is for existing volumes without existing snapshot plexes or associated snapshot volumes.

To create a snapshot mirror of a volume or volume set

Note: In the following procedure, *volume_name* is the name of either a volume or a volume set.

1. To prepare the volume for being snapshot, use the `vxsnap prepare` command:

   ```
   # vxsnap -g diskgroup prepare volume \
   alloc="storage_attribute ..."
   ```

   Note: 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 5.0, either from the command line or from within a script, use the `vxsnap prepare` command as described above.

2. 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`.

3. 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`.

4. 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.

5. List the available mirrors:

   ```
   # vxprint -g diskgroup -F%name -e"pl_v_name in \"volume_name\""
   ```

Note: The following two steps enable database FlashSnap to locate the correct mirror plexes when creating snapshots.

6 Set the dbed_flashsnap 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
```

7 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.

This example shows the steps involved in creating a snapshot mirror for the volume `data_vol` belonging to the disk group `PRODdg`.

1 Prepare the volume `data_vol` for mirroring:

```
# vxsnap -g PRODdg prepare data_vol alloc=PRODdg01
```

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
```

Note: 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.

6 Decide which data plex you want to use and set the *dbed_flashsnap* tag for it:

```
# vxedit -g PRODdg set putil2=dbed_flashsnap data_vol-02
```

7 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
```

8 To verify that the snapshot volume was created successfully, use the `vxprint -g <dg>` command as follows:

```
# vxprint -g PRODdg
TY NAME         ASSOC        KSTATE   LENGTH   PLOFFS   STATE    TUTIL0
PUTIL0
dg PRODdg       PRODdg       -        -        -        -        -       -
dm PRODdg01     Disk_1       -        71117760 -        -        -       -
dm PRODdg02     Disk_2       -        71117760 -        -        -       -
dm PRODdg03     Disk_3       -        71117760 -        -        -       -

v  data_vol     fsgen
ENABLED4194304        -        ACTIVE   -       -
pl data_vol-01  data_vol
ENABLED4194304        -        ACTIVE   -       -
sd PRODdg03-01  data_vol-01
ENABLED4194304        0        -        -       -
pl data_vol-02  data_vol
ENABLED4194304        -        SNAPDONE -       -
sd PRODdg02-01  data_vol-02
ENABLED4194304        0        -        -       -
dc data_vol_dco data_vol                                            ---
-        -       -
v  data_vol_dcl gen
ENABLED560            -        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 described 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 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.

Note: Existing snapshot volumes created by the `vxassist` command are not supported. A combination of snapshot volumes created by `vxassist` and `vxsnap` are not supported.

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=dco
   ```

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.

Note: 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 dco_volume_name
[alloc=disk_name]
```

where *dco_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=dco_plex_name convert \
state=SNAPDONE data_plex
```

where *dco_plex_name* is the name of the DCO plex you are creating.

In this example, the volume, `data_vol`, is upgraded to make use of VxVM 5.0 features.

Upgrade the disk group, `PRODdg`.

```
# vxdg upgrade PRODdg
```

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
```

Reattach any snapshot volume back to the primary volume to be upgraded.

```
# vxsnap -g PRODdg snapback SNAP-data_vol
```

Turn off FastResync on the volume to be upgraded.

```
# vxvol -g PRODdg set fastresync=off data_vol
```

Disassociate and remove any older DCO object and DCO volumes.

```
# vxassist -g PRODdg remove log  data_vol logtype=dco
```

Upgrade the volume by associating a new DCO object and DCO volume.

```
# vxsnap -g PRODdg prepare data_vol alloc="PRODdg01 PRODdg02"
```

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
TY NAME         ASSOC        KSTATE   LENGTH   PLOFFS   STATE    TUTIL0
PUTIL0
dg PRODdg       PRODdg       -        -        -        -        -       -
dm PRODdg01     Disk_1       -        71117760 -        -        -       -
dm PRODdg02     Disk_2       -        71117760 -        -        -       -
dm PRODdg03     Disk_3       -        71117760 -        -        -       -

v  data_vol     fsgen
ENABLED4194304        -          ACTIVE   -       -
pl data_vol-01  data_vol
ENABLED4194304        -          ACTIVE   -       -
sd PRODdg01-01  data_vol-01
ENABLED4194304        0          -        -       -
pl data_vol-04  data_vol
ENABLED4194304        -          SNAPDONE -       -
sd PRODdg02-03  data_vol-04
ENABLED4194304        0          -        -       -
dc data_vol_dco data_vol                                          ---
-        -       -
v  data_vol_dcl gen
ENABLED560            -          ACTIVE   -       -
pl data_vol_dcl-01 data_vol_dcl
ENABLED560            -          ACTIVE   -       -
sd PRODdg01-02  data_vol_dcl-01
ENABLED560            0          -        -       -
pl data_vol_dcl-02 data_vol_dcl
ENABLED560            -          ACTIVE   -       -
sd PRODdg02-02  data_vol_dcl-02
ENABLED560            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
TY NAME         ASSOC        KSTATE   LENGTH   PLOFFS   STATE    TUTIL0
PUTIL0
dg PRODdg       PRODdg       -        -        -        -        -       -
dm PRODdg01     Disk_1       -        71117760 -        -        -       -
dm PRODdg02     Disk_2       -        71117760 -        -        -       -
dm PRODdg03     Disk_3       -        71117760 -        -        -       -
```

```
pl data_vol-03  -
DISABLED4194304        -          -        -       -
sd PRODdg02-01  data_vol-03
ENABLED4194304         0          -        -       -

v  data_vol     fsgen
ENABLED4194304         -          ACTIVE   -       -
pl data_vol-01  data_vol
ENABLED4194304         -          ACTIVE   -       -
sd PRODdg01-01  data_vol-01
ENABLED4194304         0          -        -       -
pl data_vol-04  data_vol
ENABLED4194304         -          SNAPDONE -       -
sd PRODdg02-03  data_vol-04
ENABLED4194304         0          -        -       -
dc data_vol_dco data_vol                                          ---
-        -       -
v  data_vol_dcl gen
ENABLED560             -          ACTIVE   -       -
pl data_vol_dcl-01 data_vol_dcl
ENABLED560             -          ACTIVE   -       -
sd PRODdg01-02  data_vol_dcl-01
ENABLED560             0          -        -       -
pl data_vol_dcl-02 data_vol_dcl
DISABLED560            -          DCOSNP   -       -
sd PRODdg02-02  data_vol_dcl-02
ENABLED560             0          -        -       -
```

Scenario 2

In this scenario, there are fewer DCO plexes than data plexes.

```
# vxprint -g PRODdg
TY NAME         ASSOC        KSTATE   LENGTH   PLOFFS   STATE    TUTIL0
PUTIL0
dg PRODdg       PRODdg       -        -        -        -        -       -
dm PRODdg01     Disk_1       -        71117760 -        -        -       -
dm PRODdg02     Disk_2       -        71117760 -        -        -       -
dm PRODdg03     Disk_3       -        71117760 -        -        -       -

pl data_vol-03  -
DISABLED4194304        -          -        -       -
sd PRODdg02-01  data_vol-03
ENABLED4194304         0          -        -       -

v  data_vol     fsgen
ENABLED4194304         -          ACTIVE   -       -
pl data_vol-01  data_vol
ENABLED4194304         -          ACTIVE   -       -
sd PRODdg01-01  data_vol-01
ENABLED4194304         0          -        -       -
pl data_vol-04  data_vol
ENABLED4194304         -          ACTIVE   -       -
```

```
sd PRODdg02-03  data_vol-04
ENABLED4194304        0          -        -       -
dc data_vol_dco data_vol                                          ---
-        -       -
v  data_vol_dcl gen
ENABLED560            -          ACTIVE   -       -
pl data_vol_dcl-01 data_vol_dcl
ENABLED560            -          ACTIVE   -       -
sd PRODdg01-02  data_vol_dcl-01
ENABLED560            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.

```
# vxplex -g PRODdg -o dcoplex=data_vol_dcl-02 convert
state=SNAPDONE -V data_vol-04

# vxprint -g PRODdg
TY NAME         ASSOC        KSTATE   LENGTH   PLOFFS   STATE    TUTIL0
PUTIL0
dg PRODdg       PRODdg       -        -        -        -        -       -
dm PRODdg01     Disk_1       -        71117760 -        -        -       -
dm PRODdg02     Disk_2       -        71117760 -        -        -       -
dm PRODdg03     Disk_3       -        71117760 -        -        -       -

pl data_vol-03  -
DISABLED4194304       -          -        -       -
v  data_vol     fsgen
ENABLED4194304        -          ACTIVE   -       -
pl data_vol-01  data_vol
ENABLED4194304        -          ACTIVE   -       -
sd PRODdg01-01  data_vol-01
ENABLED4194304        0          -        -       -
pl data_vol-04  data_vol
ENABLED4194304        -          SNAPDONE -       -
sd PRODdg02-03  data_vol-04
ENABLED4194304        0          -        -       -
dc data_vol_dco data_vol                                          ---
-        -       -
v  data_vol_dcl gen
ENABLED560            -          ACTIVE   -       -
pl data_vol_dcl-01 data_vol_dcl
ENABLED560            -          ACTIVE   -       -
sd PRODdg01-02  data_vol_dcl-01
ENABLED560            0          -        -       -
pl data_vol_dcl-02 data_vol_dcl
DISABLED560           -          DCOSNP   -       -
sd PRODdg02-02  data_vol_dcl-02
ENABLED560            0          -        -       -
```

Summary of database snapshot steps

You can use Database FlashSnap commands to create a snapshot of your entire database on the same host or on a different one. Three types of snapshots can be created: `online`, `offline`, or `instant`.

If the `SNAPSHOT_MODE` specified in the snapplan is set to `online`, `dbed_vmsnap` 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.

If the `SNAPSHOT_MODE` is set to `offline`, the database must be shut down before the snapshot is created. Online redo logs and control files are required and will be used to ensure a full database recovery.

If the `SNAPSHOT_MODE` is set to `instant`, tablespaces are not put into and out of backup mode. Online redo logs and control files are required and will be used to ensure a full database recovery.

Both online and offline snapshots provide a valid backup copy of the database. You can use the snapshot as a source for backing up the database or creating a clone database for decision-support purposes. Instant snapshots *do not represent* a valid backup copy for point-in-time recovery.

The sections that follow explain 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. For more information about the `vxsnap` command, see *Veritas Volume Manager Administrator's Guide.*

Note: Make sure the volumes used by the database are configured properly before attempting to take a snapshot. This requires superuser (`root`) privileges.

Note: Anytime 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`.

Note: Database FlashSnap commands must be run by the Oracle database administrator.

To create a snapshot image of a database

1. Perform the steps in "To create a snapshot mirror of a volume or volume set" on page 345.

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* (`dbed_vmchecksnap -o list`).
   - Remove the snapplan from the SFDB repository (`dbed_vmchecksnap -o remove -f` *SNAPPLAN*).
   - Copy a snapplan from the SFDB repository to your local directory (`dbed_vmchecksnap -o copy -f` *SNAPPLAN*).

   See "Creating a snapplan (dbed_vmchecksnap)" on page 359..

3. Use the `dbed_vmsnap` command to create snapshot volumes for the database.
   See "Creating a snapplan (dbed_vmchecksnap)" on page 359.

4. On the secondary host, use the `dbed_vmclonedb` command to create a clone database using the disk group deported from the primary host.
   See "Cloning a database (dbed_vmclonedb)" on page 376. for more information.
   If the primary and secondary hosts specified in the snapplan are different, the `dbed_vmclonedb` command imports the disk group that was deported from the primary host, recovers the snapshot volumes, mounts the file systems, recovers the database, and 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.

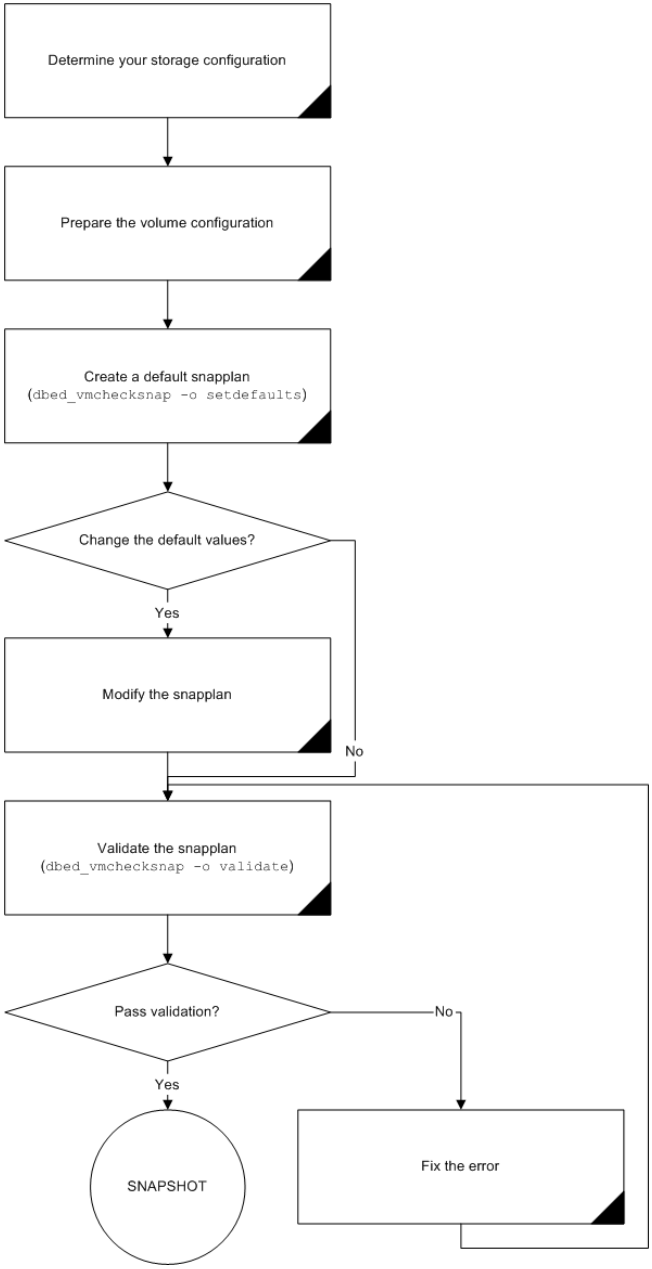
5. You can now use the clone database to perform database backup and other off-host processing work.

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.

Figure 20-3, "Prerequisites for creating a snapshot of your database," depicts the sequence of steps leading up to taking a snapshot using Database FlashSnap.

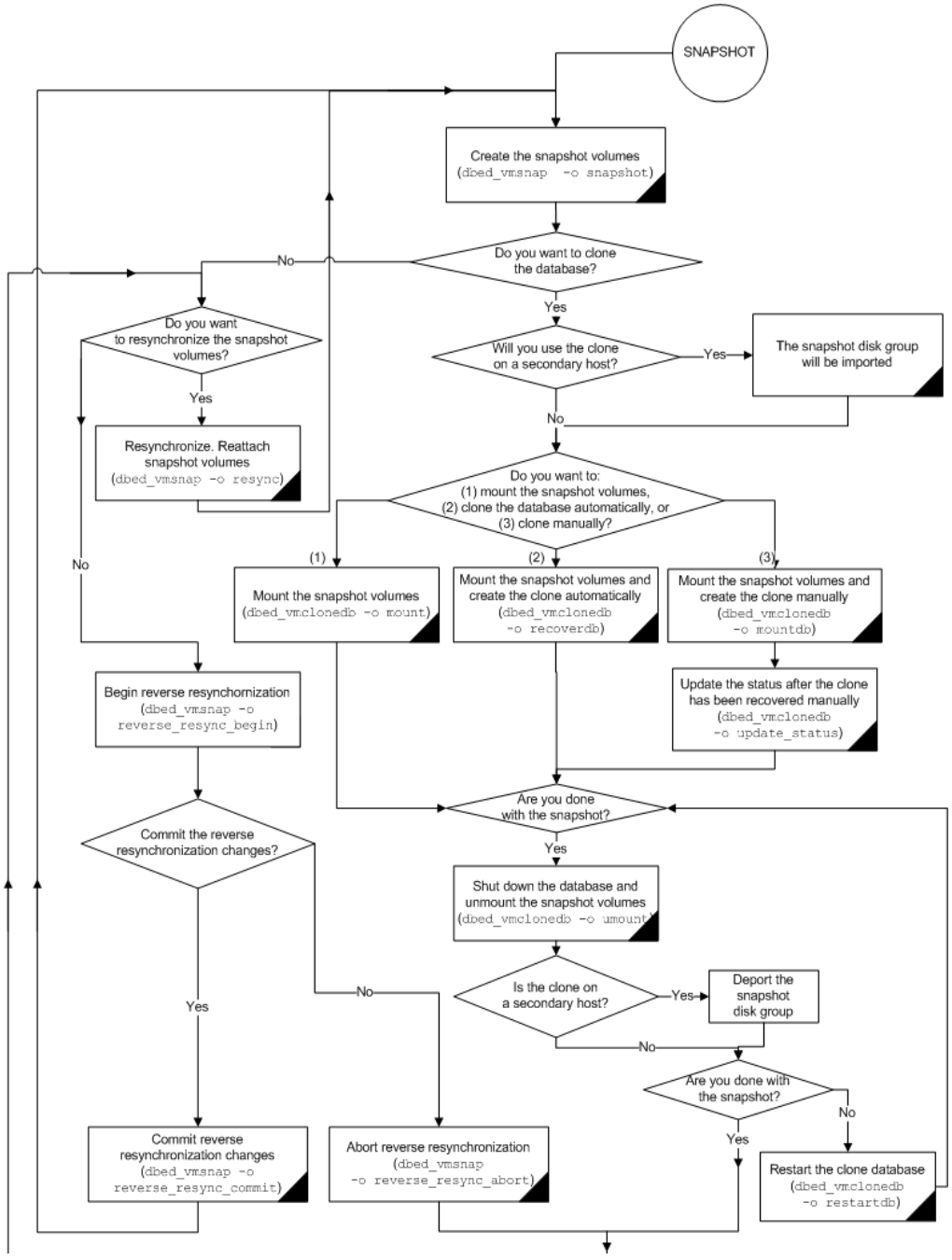
Figure 20-3 Prerequisites for creating a snapshot of your database



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.

The following flow chart depicts the actions you can perform after creating a snapshot of your database using Database FlashSnap.

Figure 20-4 Actions you can perform after creating a snapshot of your database



Creating a snapplan (dbed_vmchecksnap)

The `dbed_vmchecksnap` command creates a snapplan that `dbed_vmsnap` uses to create a snapshot of an Oracle database. The snapplan specifies snapshot scenarios (such as `online`, `offline`, or `instant`).

You can name a snapplan file whatever you choose. Each entry in the snapplan file is a line in *`parameter=argument`* format.

When using `dbed_vmchecksnap` to create or validate a snapplan, the following parameters are set:

Table 20-1 Parameter values for dbed_vmchecksnap

Parameter	Value
`SNAPSHOT_VERSION`	Specifies the snapshot version for this major release of Storage Foundation for 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.
`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 muliplexing 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 `SNAPSHOT_ARCHIVE_LOG` 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`**.

Table 20-1 Parameter values for dbed_vmchecksnap

Parameter	Value
SNAPSHOT_MODE	**online** or **offline** or **instant** Specifies whether the database snapshot should be online, offline, or instant. If the snapshot is created while the database is online, the dbed_vmsnap command will put the tablespaces into backup mode. After dbed_vmsnap 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. If the database if offline, it is not necessary to put the tablespaces into backup mode. The database must be shut down before creating an offline snapshot. If the database snapshot is instant, dbed_vmsnap will skip putting the tablespace into backup mode. **Note:** If SNAPSHOT_MODE is set to offline or instant, a two-host configuration is required and the -r *relocate_path* option is not allowed.
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 plex_tag is 15 characters. The default plex tag is dbed_flashsnap.
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.
ALLOW_REVERSE_RESYNC	**yes** or **no** By default, reverse resynchronization is off (set equal to no). If it is set to yes, data from the snapshot volume can be used to update the primary volume.
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: 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 in"Preparing hosts and storage for Database FlashSnap" on page 341.
- 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 `vxdg`(1M) 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 (`init.ora`).
- The Oracle database must have at least one mandatory archive destination.
  See "Establishing a mandatory archive destination" on page 364..
- *`ORACLE_HOME`* cannot reside on disk which will be used for snapshot.

Usage Notes

- The snapplan must be created on the primary host.
- After creating the snapplan using the `dbed_vmchecksnap` 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 `dbed_vmchecksnap` (1M) online manual page for more information.
- If the `SNAPSHOT_MODE` for the database is set to `online`, the primary and secondary hosts can be the same. If the `SNAPSHOT_MODE` is set to `offline` or `instant`, the primary and secondary hosts must be different.

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.

In this example, a snapplan, `snap1`, is created for a snapshot image in a single-host configuration and default values are set. The host is named `host1` and the working directory is `/export/snap_dir`.

```
$ cd /export/snap_dir
$ /opt/VRTS/bin/dbed_vmchecksnap -S PROD \
-H /oracle/product/9i -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 other example, a snapplan, `snap2`, is created for a snapshot image in a two-host configuration, and default values are set. The primary host is `host1`, the secondary host is `host2`, and the working directory is `/export/snap_dir`.

```
$ cd /export/snap_dir
$ /opt/VRTS/bin/dbed_vmchecksnap -S PROD \
-H /oracle/product/9i -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 $ORACLE_SID -H $ORACLE_HOME -O setdefaults \
    -p my_tag -f snap1 -t PROD
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 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 gets disabled. Snapshot volumes can be mounted and the entire database snapshot is usable even if one of the mirror gets disabled. The multi-mirror snapshots are enabled via `SNAPSHOT_MIRROR=<n>` in the snapplan.

Note: There are no changes to the Command Line usage or arguments for the Flashsnap tools.

Note: Before taking the snapshot, make sure all tagged snapshot mirrors are in SNAPDONE state.

The following sample explains the setup and the procedure for taking multi-mirror snapshots:

1 Add the second mirror and DCO log. When allocating storage for the second mirror and DCO logs, make sure the snap volumes are splittable. If snap volumes are not splittable, `dbed_vmchecksnap` fails with appropriate errors. Tag the newly added mirror with the same tag as that of the first mirror. Assume that the volume has `fastresync = on`, has `dcolog = on`, and already has one `SNAPDONE` mirror and is tagged with `dbed_flashsnap`.

```
# vxsnap -g dg_a  addmir dg_a_vol1 alloc=dg_a03
# vxedit -g dg_a set putil2=dbed_flashsnap
dg_a_vol1-03
```

2 Addkeyword to the snapplan.Here is a sample snapplan.

```
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=2
```

Establishing a mandatory archive destination

When cloning a database using Database FlashSnap, the Oracle database must have at least one mandatory archive destination.

See "Cloning a database (dbed_vmclonedb)" on page 376.

If no mandatory archive destination is set, `dbed_vmchecksnap` results in this error message:

```
SFORA dbed_vmchecksnap ERROR V-81-5677 Could not find a mandatory,
primary and valid archive destination for database PROD.

Please review the LOG_ARCHIVE_DEST_n parameters and check
v$archive_dest.
```

This example shows how to establish a mandatory archive destination using SQL*Plus:

```
alter system set log_archive_dest_1 =
'LOCATION=/ora_mnt/oracle/oradata/PROD/archivelogs MANDATORY
[REOPEN]' [scope=both];
```

For more information about Oracle parameters for archiving redo logs, see your Oracle documentation.

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 `dbed_vmchecksnap` 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.
- The default behavior is to force validation. Use the `-n` option if you want to skip validation.
- When using `dbed_vmchecksnap` 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.

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
```

Note: In HA environment, you must modify the default snapplan, use the virtual host name defined for the resource group for the PRIMARY_HOST and/or SECONDARY_HOST, and run validation.

In the following example, a snapplan, snap1, is validated for a snapshot image in a single-host configuration. The primary host is host1 and the working directory is /export/snap_dir.

```
$ cd /export/snap_dir
$ /opt/VRTS/bin/dbed_vmchecksnap -S PROD -H /oracle/product/9i \
-f snap1 -o validate

PRIMARY_HOST is host1

SECONDARY_HOST is host1

The version of PRIMARY_DG-PRODdg is 110.

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.
```

Displaying, copying, and removing a snapplan (dbed_vmchecksnap)

Consider these 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 re-validate it.
- If the database schema or disk group is modified, you must revalidate.

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
```

Note: 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=yes
SNAPSHOT_MIRROR=1
STORAGE_INFO
PRODdg02
SNAP_PLEX=prod_ar-02

STATUS_INFO
SNAP_STATUS=init_full
DB_STATUS=init
```

To copy a snapplan from the SFDB repository to your current directory

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.

Use the dbed_vmchecksnap command as follows:

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S ORACLE_SID \
-f SNAPPLAN -o copy
```

Example

In the following example, the snapplan, snap1, is copied from the SFDB repository to the current directory.

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S PROD \
-f snap1 -o copy
Copying 'snap1' to '/export/snap_dir'
```

To remove a snapplan from the SFDB repository

A snapplan can be removed from a local directory or repository if the snapplan is no longer needed.

Use the dbed_vmchecksnap command as follows:

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S ORACLE_SID -f\
SNAPPLAN -o remove
```

Example

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. dbed_vmsnap 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 *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 `dbed_vmchecksnap` before you can create a snapshot image with `dbed_vmsnap`.

Usage Notes

- The `dbed_vmsnap` 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 `$ORACLE_HOME/dbs` 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 `dbed_vmsnap`(1M) manual page for more information.

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 command:

   ```
   $ /opt/VRTS/bin/dbed_vmsnap -S ORACLE_SID -f SNAPPLAN \
       -o snapshot
   ```

Note: 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.

Note: 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-host 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.

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 371.

You can also create another Oracle database for decision-support purposes.

See "Cloning a database (dbed_vmclonedb)" on page 376.

In this example, a snapshot image of the database, `PROD`, is created for a single-host configuration. In this case, the `SECONDARY_HOST` parameter is set the same as the `PRIMARY_HOST` parameter in the snapplan.

```
$ /opt/VRTS/bin/dbed_vmsnap -S PROD -f snap1 -o snapshot

dbed_vmsnap started at 2004-04-02 14:15:27
SFDB 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 2004-04-02 14:16:11
```

In this example, a snapshot image of the primary database, PROD, is created for a two-host 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 2004-04-09 23:01:10
SFDB 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.
SFDB 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:

dbed_vmsnap ended at 2004-04-09 23:02:58
```

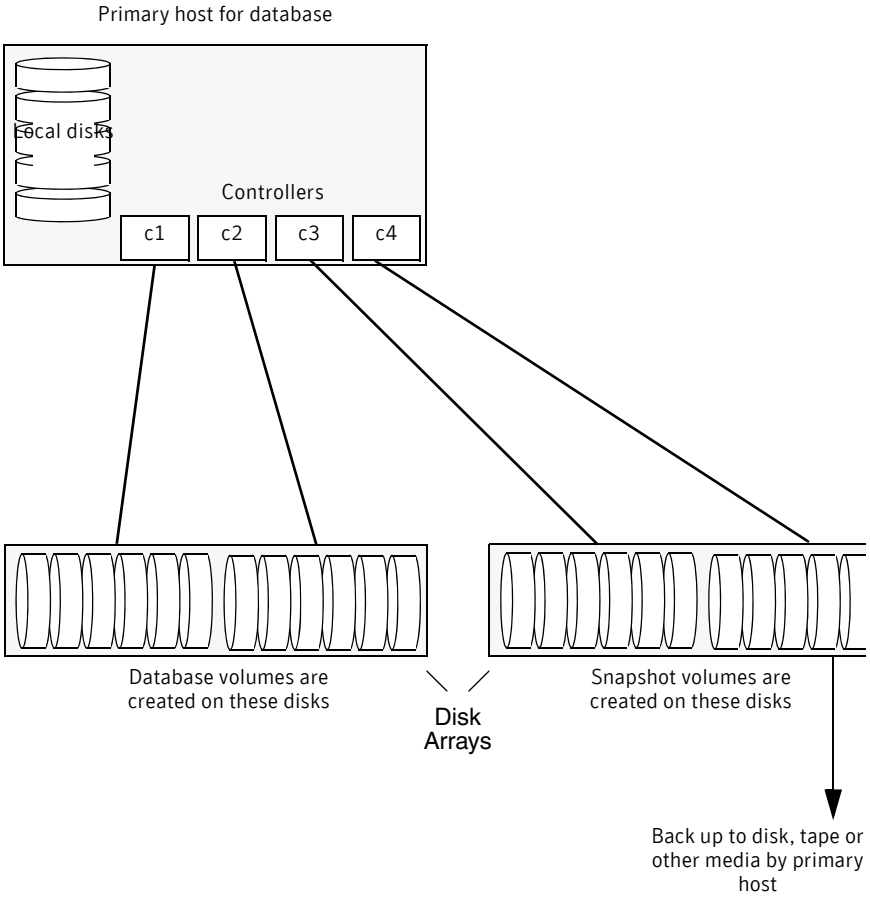
Backing up the database from snapshot volumes (dbed_vmclonedb)

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 20-5, "Example system configuration for database backup on the primary host," shows a typical configuration when snapshot volumes are located on the primary host.

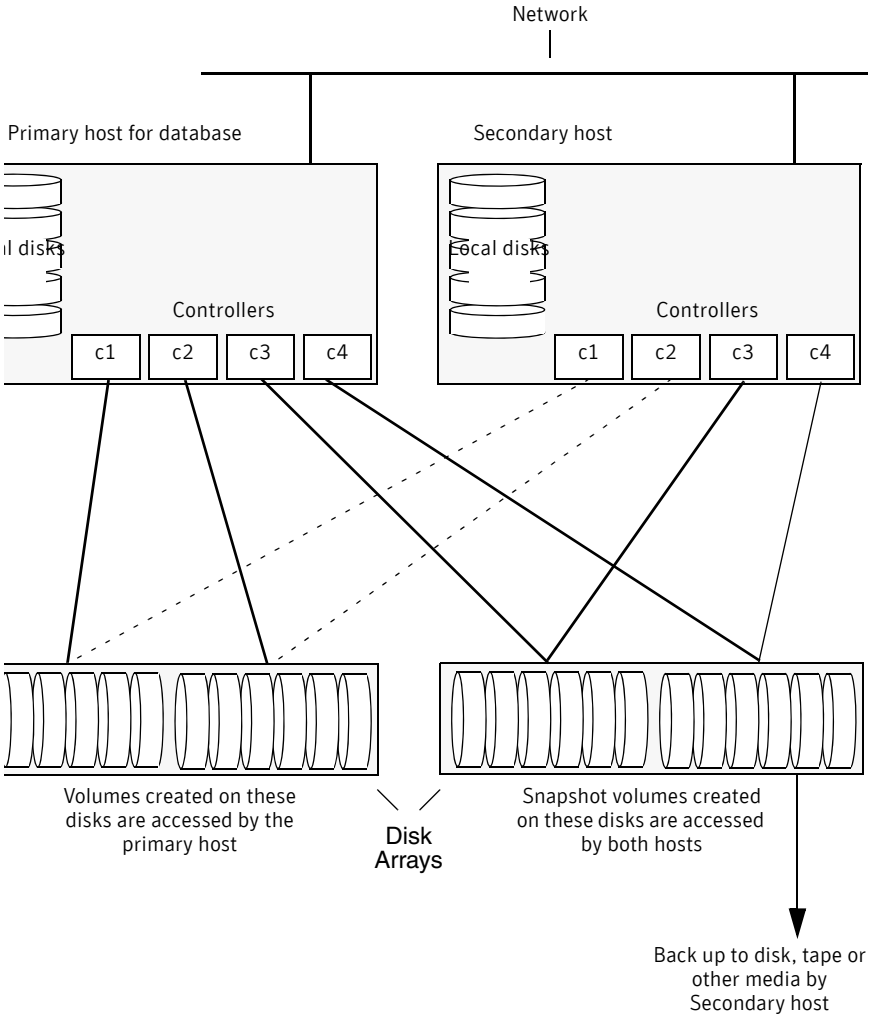
Figure 20-5 Example system configuration for database backup on the primary



host

Figure 20-6, "Example system configuration for database backup on a secondary host,"shows a typical configuration when snapshot volumes are used on a secondary host.

Figure 20-6 Example system configuration for database backup on a secondary



host

Prerequisites

- You must be logged in as the Oracle database administrator to use `dbed_vmclonedb` command.
- Before you can use the `dbed_vmclonedb` command, you must complete the steps in "Summary of database snapshot steps" on page 353, "Validating a snapplan (dbed_vmchecksnap)" on page 365, and "Creating a snapshot (dbed_vmsnap" on page 368.
- The volume snapshot must contain the entire database.
- Before you can use the `dbed_vmclonedb` command with the `-r relocate_path` 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 `dbed_vmclonedb` command can be used on the secondary host.
- In a single-host configuration, the primary and secondary hosts are the same.
- In a single-host configuration, `-r relocate_path` is required.
- In a two-host configuration, the `SFDBvol=`*`vol_name`* option is required.
- If `SNAPSHOT_MODE` is set to `offline` or `instant`, a two-host configuration is required and `-r` *`relocate_path`* is not allowed.
- See the `dbed_vmclonedb`(1M) 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.

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 -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 -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
```

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:

   ```
   $ /opt/VRTS/bin/dbed_ckptdisplay -S ORACLE_SID -n
   ```

2. You can mount one of the listed Storage Checkpoints using:

   ```
   $ /opt/VRTS/bin/dbed_ckptmount -S ORACLE_SID -c CKPT_NAME \
       -m MOUNT_POINT
   ```

Limitations

- Any mounted Storage Checkpoints must be unmounted before running the following commands:

  ```
  $ /opt/VRTS/bin/dbed_vmclonedb -o umount,new_sid=new_sid \
      -f SNAPPLAN
  ```

- It is only possible to mount a Storage Checkpoint carried over with the snapshot volumes in a two-host 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.

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.

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 371.

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 single-host 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 single-host 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-host 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.

Caution: 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 `dbed_vmclonedb` command, you must complete the steps in "Summary of database snapshot steps" on page 353, "Validating a

snapplan (dbed_vmchecksnap)" on page 365, and "Creating a snapshot (dbed_vmsnap" on page 368.

- 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 `dbed_vmclonedb` command with the `-r relocate_path` 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.
- If `SNAPSHOT_MODE` is set to `offline` or `instant`, a two-host configuration is required and `-r` *`relocate_path`* is not allowed.
- The Oracle database must have at least one mandatory archive destination. See "Establishing a mandatory archive destination" on page 364.

Usage Notes

- The `dbed_vmclonedb` command can be used on the secondary host.
- In a single-host configuration, `-r` *`relocate_path`* is required. This command is also needed if the name of the clone database is different than the primary database.
- In a two-host configuration, the `SFDBvol=`*`vol_name`* 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 `init.ora` file that has reduced memory requirements. If the host where `dbed_vmclonedb` 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 `dbed_vmclonedb`(1M) manual page for more information.

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[,SFDBvol=vol_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
\
-f SNAPPLAN [-r relocate_path]
```

Example

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.

```
$ /opt/VRTS/bin/dbed_vmclonedb -S PROD -g SNAP_PRODdg \
-o mountdb,new_sid=NEWPROD -f snap1 -r /clone
dbed_vmclonedb started at 2004-04-02 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.
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 2004-04-02 15:34:59
```

The database is recovered manually using `dbinitdb`.

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
\
-f snap1 -r /clone
dbed_vmclonedb started at 2004-04-02 15:19:16
The snapshot status has been updated.
dbed_vmclonedb ended at 2004-04-02 15:19:42
```

Example

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 -g SNAP_PRODdg \
-o mountdb,new_sid=NEWPROD,SFDBvol=SNAP_arch -f snap2
dbed_vmclonedb started at 2004-04-09 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.
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 2004-04-09 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
\
-f snap2
dbed_vmclonedb started at 2004-04-09 23:34:01
The snapshot status has been updated.
dbed_vmclonedb ended at 2004-04-09 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[,SFDBvol=vol_name] -f SNAPPLAN \
[-H ORACLE_HOME] [-r relocate_path]
```

Where:

- *ORACLE_SID*is the name of the Oracle database used to create the snapshot.
- *snap_dg* is the name of the diskgroup that contains all the snapshot volumes.
- `new_sid` specifies the *ORACLE_SID* for the clone database.
- `SFDBvol` is the volume that contains the snapplan data. This name is provided after you run `dbed_vmsnap -o snapshot`.
- *SNAPPLAN* is the name of the snapplan file.
- *ORACLE_HOME* is the *ORACLE_HOME* setting for the *ORACLE_SID* database.
- *relocate_path* is the name of the initial mount point for the snapshot image.

Note: 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 -f snap1 -r /clone
dbed_vmclonedb started at 2004-04-02 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.
Database NEWPROD (SID=NEWPROD) is running.
dbed_vmclonedb ended at 2004-04-02 14:43:05
```

Example

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,SFDBvol=SNAP_arch -f snap2
dbed_vmclonedb started at 2004-04-09 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.
Database NEWPROD (SID=NEWPROD) is running.
dbed_vmclonedb ended at 2004-04-09 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 \
-f SNAPPLAN [-r relocate_path]
```

Example

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 single-host configuration).

```
$ /opt/VRTS/bin/dbed_vmclonedb -o umount,new_sid=NEWPROD \
-f snap1 -r /clone
dbed_vmclonedb started at 2004-04-02 15:11:22
NOTICE: Umounting /clone/prod_db.
NOTICE: Umounting /clone/prod_ar.
dbed_vmclonedb ended at 2004-04-02 15:11:47
```

Example

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 (a two-host configuration).

```
$ /opt/VRTS/bin/dbed_vmclonedb -o umount,new_sid=NEWPROD \
-f snap2
dbed_vmclonedb started at 2004-04-09 23:09:21
```

```
NOTICE: Umounting /clone/arch.
NOTICE: Umounting /clone/prod_db.
dbed_vmclonedb ended at 2004-04-09 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 -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 (a single-host configuration).

```
$ /opt/VRTS/bin/dbed_vmclonedb -S PROD -g SNAP_PRODdg \
-o restartdb,new_sid=NEWPROD -f snap1 -r /clone
dbed_vmclonedb started at 2004-04-02 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 2004-04-02 15:15:19
```

In this example, the clone database is re-started on the secondary host (a two-host configuration).

```
$ /opt/VRTS/bin/dbed_vmclonedb -S PROD -g SNAP_PRODdg \
-o restartdb,new_sid=NEWPROD,SFDBvol=SNAP_arch -f snap2
dbed_vmclonedb started at 2003-04-09 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 2003-04-09 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, `dbed_vmclonedb` will display the `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 $ORACLE_SID is at
/tmp/add_tf.$ORACLE_SID.sql.
```

Note: *$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.

Note: 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.

There are two choices when resynchronizing the data in a volume:

- Resynchronizing the snapshot from the original volume. This option is explained in this section.
- Resynchronizing the original volume from the snapshot. This choice is known as *reverse resynchronization*. Reverse resynchronization may be necessary to restore a corrupted database and is usually much quicker than using alternative approaches such as full restoration from backup media.

Prerequisites

- You must be logged in as the Oracle database administrator.
- Before you can resynchronize the snapshot image, you must complete the steps in "Summary of database snapshot steps" on page 353, "Validating a snapplan (dbed_vmchecksnap)" on page 365, and "Creating a snapshot (dbed_vmsnap" on page 368.
- If a clone database has been created, shut it down and unmount the file systems using the `dbed_vmclonedb -o umount` command. This command also deports the disk group if the primary and secondary hosts are different.
  See "Shutting Down the Clone Database and Unmounting File Systems" on page 380..

Usage Notes

- The `dbed_vmsnap` command can only be executed on the primary host.

- In a two-host 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.
- You cannot access the Database FlashSnap commands `dbed_vmsnap`, (`dbed_vmchecksnap`, and `dbed_vmclonedb`) with the SFDB menu utility.

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 2004-04-02 16:19:05
The option resync has been completed.
dbed_vmsnap ended at 2004-04-02 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 root.
- 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:

a 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:

b 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 -D db -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
```

Section VII

Performance and troubleshooting

Chapter 21

Investigating I/O performance using storage mapping

Veritas Storage Foundation for Oracle RAC provides storage mapping which enables you to map datafiles to physical devices. To obtain and view detailed storage topology information, use the `vxstorage_stats` command. You can also use the Oracle Enterprise Manager to access storage mapping information.

This chapter contains the following topics:

Understanding storage mapping

Storage mapping enables you to map datafiles to physical devices. You may obtain and view detailed storage topology information using the `vxstorage_stats` and `dbed_analyzer` commands. You may also use the Oracle Enterprise Manager to access storage mapping information.

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 Veritas Storage Foundation for Oracle storage mapping option, you can map datafiles to physical devices. 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 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 Foundation for Oracle 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. 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 `vxstorage_stats` command. This command displays the complete I/O topology mapping of specific datafiles through intermediate layers like logical volumes down to actual physical devices.
- The `dbed_analyzer` 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.

In addition, you can also use the Oracle Enterprise Manager GUI to display storage mapping information after file mapping has occurred. Oracle Enterprise Manager does not display I/O statistics information. Unlike the information displayed using the Veritas command line, the information displayed in Oracle Enterprise Manager may be "stale," that is, it may not be the latest information.

For information on the command line options or the Oracle Enterprise Manager, see the chapter on using storage mapping in the *Veritas Storage Foundation for Oracle Administrator's Guide*.

Verifying the storage mapping setup

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 "VWERITAS Mapping Services" -f \
"Found_Edi_map"
Found_Edi_map feature is licensed
```

2. Verify that the `VRTSvxmsa` package is installed.

```
# lslpp -l VRTSvxmsa
Fileset                    Level      State          Description
----------------------------------------------------------------
Path: /usr/lib/objrepos
VRTSvxmsa                  4.4.0.10   COMMITTED      Veritas -
VxMS Mapping Service, Application Libraries
```

Using vxstorage_stats

The `vxstorage_stats` command displays detailed storage mapping information and I/O statistics about an individual VxFS file. 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 root.

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.
- For more information, see the `vxstorage_stats`(1m) online manual page.
- 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.
- For more information, see the `vxstorage_stats`(1m) online manual page.
- The `-m` option displays the I/O topology for the specified file.

Displaying 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:

```
$ /opt/VRTSdbed/bin/vxstorage_stats -m -f /oradata/system01.dbf
```

Output similar to the following is displayed:

```
TY  NAME                        NSUB    DESCRIPTION     SIZE(sectors)  OFFSET(sectors)
PROPERTIES
fi  /oradata/system01.dbf       1       FILE            262148096(B)   11534336
Extents:2 Sparse Extents:0
v   /dev/vx/rdsk/mapdg/oradata  1       MIRROR          12582912           0
pl vxvm:mapdg/oradata-01        2       STRIPE          12582912           0
Stripe_size:256
rd  /dev/vx/rdmp/EMC0_16        2       PARTITION       6291456            0
sd  /dev/rhdisk13               2       MIRROR          17846400           0
da  EMC000184502242:02:0c:01    0       DISK            143112192          0
```

```
da  EMC000184502242:31:0c:01  0      DISK          143112192        0
sd  /dev/rhdisk41             2      MIRROR        17846400         0
da  EMC000184502242:02:0c:01  0      DISK          143112192        0
da  EMC000184502242:31:0c:01  0      DISK          143112192        0
rd  /dev/vx/rdmp/EMC0_12      2      PARTITION     6291456          0
sd  /dev/rhdisk14             2      MIRROR        17846400         0
da  EMC000184502242:01:0d:01  0      DISK          143112192        0
da  EMC000184502242:32:0d:01  0      DISK          143112192        0
sd  /dev/rhdisk42             2      MIRROR        17846400         0
da  EMC000184502242:01:0d:01  0      DISK          143112192        0
da  EMC000184502242:32:0d:01  0      DISK          143112192        0
```

Note: For file type (`fi`), the `SIZE` column is number of bytes, and for volume (`v`), plex (`pl`), 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

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:

$ **/opt/VRTSdbed/bin/vxstorage_stats -s -f **
/data/system01.dbf

Output similar to the following is displayed:

```
                I/O OPERATIONS     I/O BLOCKS(512 byte)   AVG TIME(ms)

OBJECT                      READ       WRITE      B_READ     B_WRITE    AVG_RD  AVG_WR
/oradata/system01.dbf       2246       7199       18192      57592      1.12    138.04
/dev/vx/rdsk/ramdg/oradata  7662       30082      65609      1834893    0.67    38.54
vxvm:ramdg/oradata-01       7662       30082      65609      1834893    0.67    38.54
/dev/rhdisk13               1816       8410       16313      864243     0.86    68.69
EMC000184502242:02:0c:01    5543529    15683649   118054696  64581888   -       -
EMC000184502242:31:0c:01    6023668    15528423   151474336  962147391  -       -
/dev/rhdisk4                1816       8410       16313      864243     0.86    68.69
EMC000184502242:02:0c:01    543529     15683649   118054696  964581888  -       -
EMC000184502242:31:0c:01    6023668    15528423   151474336  962147391  -       -
/dev/rhdisk14               6747       22095      56504      970717     0.69    26.37
EMC000184502242:01:0d:01    5327497    13762427   94735284   596679746  -       -
EMC000184502242:32:0d:01    22610445   30183683   868190315  1319215657 -       -
/dev/rhdisk42               6747       22097      56504      970733     0.69    26.36
EMC000184502242:01:0d:01    5327497    13762427   94735284   596679746  -       -
EMC000184502242:32:0d:01    22610445   30183683   868190315  1319215657 -       -
```

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 to display storage mapping and I/O statistics information at repeated intervals. 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, to display statistics twice with a time interval of two seconds:

```
$ /opt/VRTSdbed/bin/vxstorage_stats -s -i2 -c2 \
-f /data/system01.dbf
```

Output similar to the following is displayed:

```
                        OPERATIONS     FILE BLOCKS(512 byte)         AVG TIME(ms)

OBJECT                  READ    WRITE      B_READ     B_WRITE    AVG_RD     AVG_WR
/data/system01.dbf      615     19         20752      152        3.53    24.74
/dev/vx/rdsk/mapdg/data_vol19386  33227    895692     1376438    9.27    16.18
vxvm:mapdg/data_vol-01  19386   33227      895692     1376438    9.26    14.03
/dev/rdsk/c1t10d0s2     19386   33227      895692     1376438    9.26    14.03
c1t10d0                 19386   33227      895692     1376438    9.26    14.03
vxvm:mapdg/data_vol-03      0   33227          0      1376438    0.00    14.21
/dev/rdsk/c1t13d0s2         0   33227          0      1376438    0.00    14.21
c1t13d0                     0   33227          0      1376438    0.00    14.21
                        I/O OPERATIONS     I/O BLOCKS(512 byte)   AVG TIME(ms)

OBJECT                  READ       WRITE      B_READ     B_WRITE    AVG_RD  AVG_WR
/oradata/system01.dbf   287021     275609     2296504    2898416    2.02    98.02
/dev/vx/rdsk/ramdg/oradata307856   372064     3690425    6426299    1.98    77.61
vxvm:ramdg/oradata-01   307869     372065     3690529    6426511    1.98    77.61
/dev/rhdisk13           146504     158258     1763433    2966464    2.16    92.61
EMC000184502242:02:0c:01 5546124   15721781   118224296  967486277     -       -
EMC000184502242:31:0c:01 6024254   15566930   151512680  965106560     -       -
/dev/rhdisk41           148643     158463     1780545    2985152    2.15    92.49
EMC000184502242:02:0c:01 5546124   15721781   118224296  967486277     -       -
EMC000184502242:31:0c:01 6024254   15566930   151512680  965106560     -       -
/dev/rhdisk14           169737     224798     1955720    3535868    2.23    67.45
EMC000184502242:01:0d:01 5330453   13888409   94918220   603917193     -       -
EMC000184502242:32:0d:01 22612024  30295506   868273331  1326216512    -       -
/dev/rhdisk42           169737     226953     1955720    3723823    2.23    69.03
EMC000184502242:01:0d:01 5330453   13888409   94918220   603917193     -       -
EMC000184502242:32:0d:01 22612024  30295506   868273331  1326216512    -       -
```

Using dbed_analyzer

Effectively performing a parallel backup requires an understanding of which tablespaces reside on which disks. If two tablespaces reside on the same disk, for example, backing them up in parallel will not reduce their downtime.

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 `-o sort=tbs` option provides the layout of the specified tablespaces on the physical disk as well as the amount of disk space they are using.
- The `-o sort=disk` option provides the name of the disks containing the specified tablespaces as well as the amount of disk space the tablespaces are using.
- The `-f` *`filename`* option specifies the name of a file containing a list of the tablespaces for which to obtain mapping information.
- The `-t` *`tablespace`* option specifies the name of a tablespace for which to obtain mapping information.
- If `-f` *`filename`* or `-t` *`tablespace`* is not specified then all the tablespaces in the database will be analyzed.
- For more information, see the `dbed_analyzer`(1M) online manual page.

Obtaining 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,

```
$ /opt/VRTSdbed/bin/dbed_analyzer -S PROD -H /usr1/oracle \
-o sort=tbs -f /tmp/tbsfile
```

Output similar to the following is displayed in the file `tbsfile`:

```
TBSNAME     DATAFILE                  DEVICE             SIZE(sectors)
SYSTEM      /usr1/oracle/rw/DATA/PROD.dbf c3t21000020379DBD5Fd0819216
TEMP        /usr1/oracle/rw/DATA/temp_20000c3t21000020379DBD5Fd0 1021968
TEMP        /usr1/oracle/rw/DATA/temp_20001c3t21000020379DBD5Fd0 2048016
SYSAUX      /usr1/oracle/rw/DATA/sysaux.dbfc3t21000020379DBD5Fd0 819216
ITEM        /usr1/oracle/rw/DATA/item_1000c3t21000020379DBD5Fd0  1021968
ITM_IDX     /usr1/oracle/rw/DATA/itm_idx_2000c3t21000020379DBD5Fd0
1021968
PRODID_IDX  /usr1/oracle/rw/DATA/prodid_idx_3000c3t21000020379DBD5Fd0
1021968
QTY_IDX     /usr1/oracle/rw/DATA/qty_idx_7000 c3t21000020379DBD5Fd0
1021968
ROLL_1      /usr1/oracle/rw/DATA/roll_1_5000c3t21000020379DBD5Fd01021968
ROLL_2      /usr1/oracle/rw/DATA/roll_2_6000c3t21000020379DBD5Fd01021968
ORDERS      /usr1/oracle/rw/DATA/orders_4000c3t21000020379DBD5Fd01021968
ORD_IDX     /usr1/oracle/rw/DATA/ord_idx_10000c3t21000020379DBD5Fd0
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_6001c3t21000020379DBD5Fd0
1024016
ITEM        /usr1/oracle/rw/DATA/item_1001c3t21000020379DBD5Fd0  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,

```
$ /opt/VRTSdbed/bin/dbed_analyzer -S PROD -H /usr1/oracle \
-o sort=disk -f /tmp/tbsfile
```

Output similar to the following is displayed in the file `tbsfile`:

```
DEVICE              TBSNAME    DATAFILE                  SIZE(sectors)
c3t21000020379DBD5Fd0SYSTEM   /usr1/oracle/rw/DATA/PROD.dbf 819216
c3t21000020379DBD5Fd0TEMP     /usr1/oracle/rw/DATA/temp_200001021968
c3t21000020379DBD5Fd0TEMP     /usr1/oracle/rw/DATA/temp_200012048016
c3t21000020379DBD5Fd0SYSAUX   /usr1/oracle/rw/DATA/sysaux.dbf819216
```

```
c3t21000020379DBD5Fd0  ITEM    /usr1/oracle/rw/DATA/item_1000    1021968
c3t21000020379DBD5Fd0  ITM_IDX     /usr1/oracle/rw/DATA/itm_idx_2000
1021968
c3t21000020379DBD5Fd0  PRODID_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
```

Oracle file mapping (ORAMAP)

Veritas has defined and implemented two libraries: `libvxoramap_64.so` and `libvxoramap_64.sl`. These two libraries provide a mapping interface to Oracle9*i* release 2 or later. `libvxoramap_64.so` serves as a bridge between Oracle's set of storage APIs (known as "ORAMAP") and Veritas Federated 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.

With Veritas Storage Foundation for Oracle 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. This information can be used to determine the exact location of an Oracle data block on a physical device and to help identify hot spots.

Note: To use the mapping functionality, you must be using Oracle 9.2.0.3 or later.

Mapping components

The mapping components in the System Global Area (SGA) and Oracle's representation of these components are described in this section. 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:

- 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. Brief descriptions of these views are provided below. For more detailed information, refer to your Oracle documentation.

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.
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 setup

To verify that $ORACLE_HOME is set up for Oracle file mapping (ORAMAP)

1. Enter:

```
# cd $ORACLE_HOME/rdbms/filemap/bin
# ls -l
-r-xr-x--- 1 root system 900616 Apr 08 19:16 fmputl
-r-sr-xr-x 1 root system 14614 Apr 08 19:16 fmputlhp
```

2. Verify that:
   - `fmputlhp` is owned by `root` and that the `setud` bit is set.
   - The permissions for `fmputlhp` are set to `-r-sr-xr-x`.
   - The permissions for `fmputl` are set to `-r-xr-x---`.

3 If any of these items is not set as specified, make the appropriate corrections.

Enabling 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.

```
# cd $ORACLE_HOME/rdbms/filemap/etc
# cat filemap.ora
```

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`.

```
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.

Note: 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

To access dynamic performance views

1 Confirm that the Veritas mapping library for Oracle file mapping has been enabled.

```
SQL> select lib_idx idx, lib_name name, vendor_name vname, \
path_name path from v$map_library;
IDX             NAME        VNAME       PATH
-------------------------------------------------------------
```

2 `Veritas ORAMAP APIVeritas /opt/VRTSdbed/lib/libvxoramap_64.so`

After storage mapping has been enabled, Oracle datafiles can be mapped using the `DBMS_STORAGE_MAP` package.

The following example shows how to map a datafile using SQL:

```
SQL> execute
dbms_storage_map.map_file('/oradata/system01.dbf’,'DATAFILE',
TRUE);
```

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:

```
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;
NAME                    IDX     STATUS  TYPE    STR         FSIZE       NEXTS
-----------------------------------------------------------------------------
/oradata/system01.dbf  0   VALID   DATAFILE       FILE    512008      2
```

To display the contents of v$map_file_extent.

```
SQL> select * from v$map_file_extent;
FILE_MAP_IDX   EXT_NUM     EXT_ELEM_OFF   EXT_SIZE   EXT_FILE_OFF   EXT_TY  ELEM_IDX
-----------------------------------------------------------------------------------
0              0           11534336       512000     0              DATA        0
0              1           36504          8          512000         DATA        0
```

To display the contents of v$map_element:

```
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
-----------------------------------------------------------------------------------
0   /dev/vx/rdsk/ramdg/oradata MIRROR         12582912   1          MIRROR
1   vxvm:ramdg/oradata-01      STRIPE         12582912   2          STRIPE
2   /dev/vx/rdmp/EMC0_16       PARTITION      17846400   2          HOST DEVICE
3   /dev/rhdisk13              MIRROR         17846400   2          MIRROR
4   EMC000184502242:02:0c:01   DISK           1.43E+08   0          DISK
5   EMC000184502242:31:0c:01   DISK           1.43E+08   0          DISK
6   /dev/rhdisk41              MIRROR         17846400   2          MIRROR
7   /dev/vx/rdmp/EMC0_12       PARTITION      17846400   2          HOST DEVICE
8   /dev/rhdisk14              MIRROR         17846400   2          MIRROR
9   EMC000184502242:01:0d:01   DISK           1.43E+08   0          DISK
10  EMC000184502242:32:0d:01   DISK           1.43E+08   0          DISK
11  /dev/rhdisk42              MIRROR         17846400   2          MIRROR
```

To display the contents of v$map_subelement:

```
SQL> select * from v$map_subelement;

CHILD_IDX PARENT_IDX SUB_NUM SUB_SIZE ELEM_OFFSET SUB_FLAGS
---------- ---------- ------- -------- ----------- ---------
     1       0      0 12582912       0       0
     2       1      0  6291456       0       0
     7       1      1  6291456       0       0
     3       2      0 17846400       0       0
```

```
 6      2     1 17846400       0      0
 4      3     0 1.43E+08       0      0
 5      3     1 1.43E+08       0      0
 4      6     0 1.43E+08       0      0
 5      6     1 1.43E+08       0      0
 8      7     0 17846400       0      0
11      7     1 17846400       0      0
 9      8     0 1.43E+08       0      0
10      8     1 1.43E+08       0      0
 9     11     0 1.43E+08       0      0
10     11     1 1.43E+08       0      0
```

To display all the elements within the I/O stack for a specific file.

```
SQL> with fv as
2   (select file_map_idx, file_name from v$map_file
3   where file_name = '/oradata/system01.dbf')
4   select
5   fv.file_name, lpad(' ', 4 * (level - 1)) || \
    el.elem_name elem_name, el.elem_size, el.elem_type, \
    el.elem_descr
6   from
7   v$map_subelement sb, v$map_element el, fv,
8   (select unique elem_idx from v$map_file_io_stack io, fv
    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
-------------------------------------------------------------------------------------
/oradata/system01.dbf  vxvm:ramdg/oradata-01      12582912   STRIPE     STRIPE
/oradata/system01.dbf  /dev/vx/rdmp/EMC0_16       17846400   PARTITION  HOST DEVICE
/oradata/system01.dbf  /dev/rhdisk13              17846400   MIRROR     MIRROR
/oradata/system01.dbf  EMC000184502242:02:0c:01   1.43E+08   DISK       DISK
/oradata/system01.dbf  EMC000184502242:31:0c:01   1.43E+08   DISK       DISK
/oradata/system01.dbf  /dev/vx/rdmp/EMC0_12       17846400   PARTITION  HOST DEVICE
/oradata/system01.dbf  /dev/rhdisk14              17846400   MIRROR     MIRROR
/oradata/system01.dbf  EMC000184502242:01:0d:01   1.43E+08   DISK       DISK
/oradata/system01.dbf  EMC000184502242:32:0d:01   1.43E+08   DISK       DISK
```

About arrays for storage mapping and statistics

Veritas Storage Foundation for Oracle 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.

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.

On AIX, the following VAIL providers support these third-party storage arrays:

- The `vx_emc_symmetrix` provider manages EMC Symmetrix arrays.
- The `vx_ibmshark` provider manages IBM ESS (Shark) arrays.

For the most up-to-date array support information, see the appropriate hardware compatibility list (HCL) on the Technical Support website at:

http://support.veritas.com

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 Veritas Storage Foundation for Oracle RAC can discover them.

For detailed information about supported array models, see the *Veritas Array Integration Layer Array Configuration Guide*

Chapter 22

Troubleshooting SF Oracle RAC

Troubleshooting options, known problems, and their solutions:

Running scripts for engineering support analysis

Three 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 Veritas Tech Support can assist with analyzing and solving any problems.

getcomms

This script gathers information about the GAB and LLT modules. The file `/tmp/commslog.`*time_stamp*`.tar` contains the script's output.

To use getcomms

On *each* system, enter:

```
# /opt/VRTSgab/getcomms -local
```

hagetcf

This 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 hagetcf

On *each* system, enter:

```
# /opt/VRTSvcs/bin/hagetcf
```

Troubleshooting tips

To check the Oracle installation error log

Access:

```
$ORACLE_BASE/oraInventory/logs/installActions<date_time>.log
```

This file contains errors that occurred during installation. It clarifies the nature of the error and at exactly which point it occurred during the installation. If there are any installation problems, sending this file to Tech Support is required for debugging the issue.

To check the Veritas log file

Access:

```
/var/VRTSvcs/log/engine_A.log
```

This file contains all actions performed by `HAD`. Verify if there are any `CVM` or `PrivNIC` errors logged in this file, since they may prove to be critical errors.

Troubleshooting Oracle

For help resolving issues with Oracle components, check the:

- Oracle log files
- Oracle Notes
- Oracle Troubleshooting Topics

Oracle log files

To check the Oracle log file

- For Oracle 10g Release 1, access:
  `$CRS_HOME/crs/log`
- For Oracle 10g Release 2, access:
  `$CRS_HOME/log/hostname/crsd`
  where *hostname* is the string returned by the `hostname` command.

The log file in this directory contains the logs pertaining to the CRS resources such as the virtual IP, Listener, and database instances. The file indicates some configuration errors or Oracle problems, since CRS does not directly interact with any of the Veritas components.

To check for crs core dumps

Access:

`$CRS_HOME/crs/init`

Core dumps for the `crsd.bin` daemon are written here. Use this file for further debugging.

To check the Oracle css log file

- For Oracle 10g Release 1, access:
  `$CRS_HOME/css/log`
- For Oracle 10g Release 2, access:
  `$CRS_HOME/log/hostname/cssd`
  where *hostname* is the string returned by the `hostname` command.

The log files in this directory indicate actions such as reconfigurations, missed checkins, connects, and disconnects from the client CSS listener. If there are membership issues, they will show up here. If there are communication issues over the private networks, they are logged here. The `ocssd` process interacts with `vcsmm` for cluster membership.

To check for ocssd core dumps

Access:

`$CRS_HOME/css/init`

Core dumps from the `ocssd` and the `pid` for the `css` daemon whose death is treated as fatal are located here. If there are abnormal restarts for `css` the core files are found here.

Oracle Notes

259301.1: CRS and 10g Real Application Clusters

Note: Oracle Note 259301.1 is extremely important to read.

280589.1: CRS Installation Does Not Succeed if One or More Cluster Nodes Present are Not to be Configured for CRS.

265769.1: 10g RAC: Troubleshooting CRS Reboots

279793.1: How to Restore a Lost Vote Disk in 10g

146580.1: What is an ORA-600 Internal Error?

268937.1: Repairing or Restoring an Inconsistent OCR in RAC

239998.1: 10g RAC: How to Clean Up After a Failed CRS Install

Two items missing in the above guide are:

- Remove the `/etc/oracle/ocr.loc` file. This file contains the location for the Cluster registry. If this file is not removed then during the next installation the installer will not query for the OCR location and will pick it from this file.
- If there was a previous 9i Oracle installation, then remove the following file: `/var/opt/oracle/srvConfig.loc`. If this file is present the installer will pick up the Vote disk location from this file and may create the error "the Vote disk should be placed on a shared file system" even before specifying the Vote disk location.

272332.1: CRS 10g Diagnostic Collection Guide

Oracle troubleshooting topics

Topics indicate symptoms and likely procedures required for a solution.

Oracle user must be able to read /etc/llttab File

Check the permissions of the file `/etc/llttab`. 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 on 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.rc start
```

The command included in the `/etc/vcsmmtab` file enables the VCSMM driver to be started at system boot.

Missing dialog box during installation of Oracle9*i*

During installation of Oracle9*i* Release 2 using the `runInstaller` utility, if you choose the Enterprise Edition or Custom Install (with RAC option), a dialog box prompting you about the installation nodes appears.

Note: Symantec recommends using the `installsfrac` utility to install Oracle9*i* RAC. The following troubleshooting steps only apply if the `runInstaller` utility is called directly.

To check if the dialog box fails to appear

1. Exit the `runInstaller` utility.
2. On the first system, copy the Veritas CM library:
   a. Enter:
      ```
      # cd /usr/lib
      ```
   b. Enter:
      ```
      # cp libvcsmm.so $ORACLE_HOME/lib/libcmdll.so
      ```
3. Start the VCSMM driver on both the nodes by entering:
   ```
   # /etc/init.d/vcsmm.rc start
   ```
4. Restart the `runInstaller` utility.

Oracle log files show shutdown

The Oracle log files may show that a shutdown was called even when not shutdown manually.

The Oracle enterprise agent calls shutdown if monitoring of the Oracle/Netlsnr resources fails for some reason. On all cluster nodes, look at 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 `ifconfig -a` command returns the public IP address of the current node. If `ifconfig -a` 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.

CRS fails after restart

If the CRS fails to start up after boot up, check for the occurrence of the following strings in the `/var/adm/ras`file:

"Oracle CSSD failure. Rebooting for cluster integrity"

- 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.

"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.

"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`.

"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

Use the following procedure to remove Oracle CRS.

To reomve Oracle CRS

1. Run the `rootdelete.sh` script:

```
# cd /crshome/install
# ./rootdelete.sh
```

   a. 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.c* 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 ora*
```

5 Remove file from `$CRS_HOME` and Oracle Inventory:

```
# rm -r $CRS_HOME/*
```

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/rdsk/ocrvotedg/ocrvol bs=8192 \
count=18000
# dd if=/dev/zero of=/dev/vx/rdsk/ocrvotedg/votvol bs=8192 \
count=3000
```

7 Reboot the systems to make sure no CRS daemons are running.

Troubleshooting the VIP Configuration

When encountering issues with the VIP configuration, you can use the following commands and files:

- Use the `/etc/ifconfig -a` command on all nodes to check for network problems.
- Use the command: `/usr/bin/nslookup` *`virtual_host_name`* to make sure the virtual host name is registered with the DNS server.
- Verify the `/etc/hosts` file on each node.
- Check the output from the 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. If these files are present from a previous configuration, remove them. See “CRS fails after restart”.

Troubleshooting fencing

Topics indicate symptoms and likely procedures required for a solution.

SCSI reservation errors during bootup

When restarting a node of an SF Oracle RAC cluster, SCSI reservation errors may be observed such as:

```
Mar 25 13:18:28 galaxy 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. The message may be safely ignored.

vxfentsthdw fails when SCSI TEST UNIT READY command fails

A message may occur resembling:

```
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 Hitachi Data Systems 99XX arrays if bit 186 of the system mode option is not enabled.

vxfentsthdw fails when prior registration key exists on disk

If you attempt to use the `vxfentsthdw` utility to test a disk that has a registration key already set, it will fail. If you suspect a key exists on the disk you plan to test, use the `vxfenadm -g` command to display it.

To display a key on a disk

Enter:

```
# vxfenadm -g diskname
```

- If the disk is not SCSI-3 compliant, an error is returned indicating:

  ```
  Inappropriate ioctl for device.
  ```

- If you have a SCSI-3 compliant disk and no key exists, then the output resembles:

  ```
  Reading SCSI Registration Keys...
  Device Name: diskname
  Total Number Of Keys: 0
  No keys ...
  ```

  Proceed to test the disk using the `vxfentsthdw` utility. See "Testing the disks using the vxfentsthdw script" on page 107.

- If keys exist, you must remove them before you test the disk. Refer to "Removing Existing Keys From Disks" in the next section.

Removing existing keys from disks

Use this procedure to remove existing registration and keys created by another node from a disk.

To remove the registration and reservation keys from disk

1. Create a file to contain the access names of the disks:

   ```
   # vi /tmp/disklist
   ```

   For example:

   ```
   /dev/hdisk26
   ```

2. Read the existing keys:

   ```
   # vxfenadm -g all -f /tmp/disklist
   ```

   The output from this command displays the key:

   ```
   Device Name: /dev/hdisk26
   Total Number Of Keys: 1
   key[0]:
   Key Value [Numeric Format]:   65,49,45,45,45,45,45,45
   Key Value [Character Format]: A1------
   ```

3. If you know on which node the key was created, log in to that node and enter:

   ```
   # vxfenadm -x -kA1 -f /tmp/disklist
   ```

   The key is removed.

   If you do not know on which node the key was created, follow step 4 through step 6 to remove the key.

4. Register a second key "A2" temporarily with the disk:

   ```
   # vxfenadm -m -kA2 -f /tmp/disklist
   Registration completed for disk path /dev/sdh
   ```

5. Remove the first key from the disk by pre-empting it with the second key:

   ```
   # vxfenadm -p -kA2 -f /tmp/disklist -vA1
   key: A2------ prempted the key: A1------ on disk
   /dev/sdh
   ```

6. Remove the temporary key assigned in step 4.

   ```
   # vxfenadm -x -kA2 -f /tmp/disklist
   Deleted the key : [A2------] from device /dev/sdk
   ```

   No registration keys exist for the disk.

7. Verify that the keys were properly cleaned:

   ```
   # vxfenadm -g all -f /tmp/diskslist
   ```

System panic prevents potential data corruption

When a system 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.
```

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 *galaxy* and *nebula* is functioning normally when the private network links are broken. Also suppose *galaxy* is the ejected system. When *galaxy* reboots before the private network links are restored, its membership configuration does not show *nebula*; however, when it attempts to register with the coordinator disks, it discovers *nebula* is registered with them. Given this conflicting information about *nebula, galaxy* 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, *galaxy* reboots, and *nebula* fails to come back up. From the view of the cluster from *galaxy, nebula* may still have the registrations on the coordinator disks.

Case 1: nebula up, galaxy ejected (actual potential split brain)

To respond to Case 1

1. Determine if *galaxy* is up or not.
2. If it is up and running, shut it down and repair the private network links to remove the split brain condition.

3 Restart *galaxy*.

Case 2: nebula down, galaxy ejected (apparent potential split brain)

To respond to Case 2

1 Physically verify that *nebula* is down.

2 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.

3 Clear the keys on the coordinator disks as well as the data disks using the command `/opt/VRTSvcs/vxfen/bin/vxfenclearpre`. See "Clearing keys after split brain" on page 416.

4 Make any necessary repairs to *nebula* and reboot.

Clearing keys after split brain

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

To use vxfenclearpre

1. To prevent data corruption, shut down all other systems in the cluster that have access to the shared storage.
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:

```
scsi1 (0,6,0) : RESERVATION CONFLICT
SCSI disk error : host 1 channel 0 id 6 lun 0 return
code = 18
I/O error: dev 08:80, sector 2000
```

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 systems in the cluster.

Adding or removing coordinator disks

Use the following procedure to add disks to the coordinator disk group, or to remove disks from the coordinator disk group.

Note the following about the procedure:

✔ You must have an odd number (three minimum) of disks/LUNs in the coordinator disk group.

- ✔ The disk you add must support SCSI-3 persistent reservations; see "Viewing guidelines for checking SCSI-3 support" on page 89.
- ✔ You must reboot each system in the cluster before the changes made to the coordinator disk group take effect.

To remove and replace a disk in the coordinator disk group

1. Log in as root user on one of the cluster systems.
2. 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 system 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 blocks are removed.
3. To add disks to the disk group, or to remove disks from the disk group, use the VxVM disk administrator utility, `vxdiskadm`.
4. After disks are added or removed, deport the disk group:

   ```
   # vxdg deport `cat /etc/vxfendg`
   ```
5. Execute on all nodes in the cluster:

   ```
   # /etc/init.d/vxfen.rc start
   ```

Troubleshooting ODM

Topics indicate symptoms and likely procedures required for a solution.

File System configured incorrectly for ODM shuts down Oracle

Linking Oracle9*i* with the Veritas ODM libraries provides the best file system performance. See "Relinking the SF Oracle RAC libraries to Oracle - using the installer" on page 143 for instructions on creating the link. Shared file systems in RAC clusters without ODM Libraries linked to Oracle9*i* 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 command:

```
#   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 it should, check:

- `/dev/odm` mount options in `/etc/filesystems`. If the `nocluster` option is being used, it can force the `disabled` cluster support state.
- Make sure the `VRTSgms` (group messaging service) package is installed.

If `/dev/odm` is not mounted, no status can be reported.

To start ODM

1 Execute:

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

2 Execute:

```
# /etc/rc.d/rc2.d/S99odm start
```

Troubleshooting VCSIPC

Topics indicate symptoms and likely procedures required for a solution.

VCSIPC errors in Oracle trace/log files

If you see any VCSIPC errors in the Oracle trace/log files, check `/var/log/messages` for any LMX error messages. If you see messages that contain any of the following:

```
. . . out of buffers
. . . out of ports
. . . no minors available
```

Refer to "Tunable kernel driver parameters" on page 489.

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 CVM

Topics indicate symptoms and likely procedures required for a solution.

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
Please make sure that CVM and vxfen are configured and operating
correctly
```

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.

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. Refer to “Removing existing keys from disks” on page 413.

Starting 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. Refer to “Removing existing keys from disks” on page 413.

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:

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.

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 example indicates that not all LUNs are discovered by SCSI. The problem might be fixed 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.

Troubleshooting CVR

If the rlink is not up to date, use the `hares -action` command with the `resync` action token to synchronize the RVG.

The following example command is issued on any node (`galaxy`, in this case) in the primary cluster, specifying the RVGSharedPri resource, `ora_vvr_shpri`:

```
# hares -action ora_vvr_shpri resync -sys galaxy
```

Troubleshooting interconnects

Topics indicate symptoms and likely procedures required for a solution.

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 cfgmgr 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
```

Section VIII

Reference information

Reference information:

Appendix A

Sample VCS configuration files for SF Oracle RAC

- All configuration shown here assume that Oracle and CRS binaries are installed on local disks and that they are managed by the operating system. These file systems must be specified in the file /etc/filesystems.
- The “cluster” definition in all of the configurations should specify UseFence=SCSI3.
- Sample main.cf file examples are provided for:
  - Oracle 9i
  - Oracle 10g without the Oracle agent
  - Oracle 10g with the Oracle agent
  - Oracle 10g primary replication site
  - Oracle 10 secondary replication site

Oracle9i configuration

Configuration details:

- Named: 9i_main.cf
- Has two service groups: cvm and oradb1_grp
- oradb1_grp depends on cvm
- oradb1_grp has Oracle and oradata mount resource

```
include "types.cf"
include "CFSTypes.cf"
include "CVMTypes.cf"
include "OracleTypes.cf"

cluster rac_cluster101 (
    UserNames = { admin = bopHo }
    Administrators = { admin }
    UseFence = SCSI3
    )

system galaxy (
    )

system nebula (
    )

group oradb1_grp (
    SystemList = { galaxy = 0, nebula = 1 }
    AutoFailOver = 0
    Parallel = 1
    AutoStartList = { galaxy, nebula }
    )

    Oracle ora1 (
        Critical = 0
            Sid @galaxy = vrts1
            Sid @nebula = vrts2
            Owner = oracle
            Home = "/app/oracle/orahome"
            Pfile @galaxy = "/app/oracle/orahome/dbs/initvrts1.ora"
            Pfile @nebula = "/app/oracle/orahome/dbs/initvrts2.ora"
            )

    CFSMount oradata_mnt (
        Critical = 0
        MountPoint = "/oradata"
        BlockDevice = "/dev/vx/dsk/oradatadg/oradatavol"
        )
```

```
    CVMVolDg oradata_voldg (
        CVMDiskGroup = oradatadg
        CVMVolume = { oradatavol }
        CVMActivation = sw
        )

    requires group cvm online local firm
    ora1 requires oradata_mnt
    oradata_mnt requires oradata_voldg

group cvm (
    SystemList = { galaxy = 0, nebula = 1 }
    AutoFailOver = 0
    Parallel = 1
    AutoStartList = { galaxy, nebula }
    )

    NIC listener_nic (
        Device = en0
        NetworkType = ether
        NetworkHosts = { "10.10.11.101", "10.10.11.102" }
        )

    IP listener_ip
        Device = en0
        Address @galaxy = "10.10.11.1"
        Address @nebula = "10.10.11.2"
        NetMask = "255.255.240.0"
        )

    Netlsnr listener (
        Owner = oracle
        Home = "/app/oracle/orahome"
        TnsAdmin = "/app/oracle/orahome/network/admin"
        Listener @galaxy = listener_galaxy
        Listener @nebula = listener_nebula
        ListEnvFile = "/opt/VRTSvcs/bin/Netlsnr/envfile"
        MonScript = "./bin/Netlsnr/LsnrTest.pl"
        )

    CFSMount orasrvm_mnt (
        Critical = 0
        MountPoint = "/orasrvm"
        BlockDevice = "/dev/vx/dsk/orasrvmdg/orasrvvol"
        )

    CVMVolDg orasrvm_voldg (
        CVMDiskGroup = orasrvdg
        CVMVolume = {orasrvvol}
        CVMActivation = sw
    )
```

```
CFSfsckd vxfsckd (
    )

CVMCluster cvm_clus (
    CVMClustName = rac_cluster101
    CVMNodeId = { galaxy = 0, nebula = 1 }
    CVMTransport = gab
    CVMTimeout = 200
    )

CVMVxconfigd cvm_vxconfigd (
    Critical = 0
    CVMVxconfigdArgs = { syslog }
    )

listener requires listener_ip
listener_ip requires listener_nic
orasrvm_voldg requires cvm_clus
orasrvm_mnt requires orasrvm_voldg
orasrvm_mnt requires vxfsckd
vxfsckd requires cvm_clus
cvm_clus requires cvm_vxconfigd
```

Oracle 10g configurations

Oracle 10g configuration without Oracle agent

Configuration details:

- Named: `10g_simple_main.cf`
- Use for single 10g and Oracle database only
- Has only one service group: cvm
- cvm group includes PrivNIC and Application resource for CSSD

```
include "types.cf"
include "CFSTypes.cf"
include "CVMTypes.cf"
include "OracleTypes.cf"
include "PrivNIC.cf"

cluster rac_cluster101 (
    UserNames = { admin = bopHo}
    Administrators = { admin }
    UseFence = SCSI3
    )

system galaxy (
    )

system nebula (
    )

group cvm (
    SystemList = { galaxy = 0, nebula = 1 }
    AutoFailOver = 0
    Parallel = 1
    AutoStartList = { galaxy, nebula }
    )

    Application cssd (
        Critical = 0
        StartProgram = "/opt/VRTSvcs/rac/bin/cssd-online"
        StopProgram = "/opt/VRTSvcs/rac/bin/cssd-offline"
        CleanProgram = "/opt/VRTSvcs/rac/bin/cssd-clean"
        MonitorProgram = "/opt/VRTSvcs/rac/bin/cssd-monitor"
        OnlineRetryLimit = 20
        )

    CFSMount ocrvote_mnt (
        Critical = 0
        MountPoint = "/ocrvote"
```

```
    BlockDevice = "/dev/vx/dsk/ocrvotedg/ocrvotevol"
    )

CVMVolDg ocrvote_voldg (
    Critical = 0
    CVMDiskGroup = ocrvotedg
    CVMVolume = { ocrvotevol }
    CVMActivation = sw
    )

CFSMount oradata_mnt (
    Critical = 0
    MountPoint = "/oradata"
    BlockDevice = "/dev/vx/dsk/oradatadg/oradatavol"
    )

CVMVolDg oradata_voldg (
    Critical = 0
    CVMDiskGroup = oradatadg
    CVMVolume = { oradatavol }
    CVMActivation = sw
    )

CFSfsckd vxfsckd (
    )

CVMCluster cvm_clus (
    CVMClustName = rac_cluster101
    CVMNodeId = { galaxy = 0, nebula = 1 }
    CVMTransport = gab
    CVMTimeout = 200
    )

CVMVxconfigd cvm_vxconfigd (
    Critical = 0
    CVMVxconfigdArgs = { syslog }
    )

PrivNIC ora_priv (
    Critical = 0
    Device = { en1 = 0, en2 = 1}
    Address@galaxy = "192.168.12.1"
    Address@nebula = "192.168.12.2"
    NetMask = "255.255.240.0"
        )

cssd requires ocrvote_mnt
cssd requires oradata_mnt
cssd requires ora_priv
ocrvote_mnt requires ocrvote_voldg
oradata_mnt requires oradata_voldg
oradata_voldg requires cvm_clus
```

```
    ocrvote_voldg requires cvm_clus
    ocrvote_mnt requires vxfsckd
    oradata_mnt requires vxfsckd
    vxfsckd requires cvm_clus
    cvm_clus requires cvm_vxconfigd
```

Oracle 10g configuration with Oracle agent

Configuration details:

- Named: 10g_main.cf
- More general purpose, can have multiple Oracle databases
- Has two service groups: cvm and oradb1_grp
- oradb1_grp depends on cvm
- oradb1_grp has Oracle and oradata mount resource

```
include "types.cf"
include "CFSTypes.cf"
include "CVMTypes.cf"
include "OracleTypes.cf"
include "PrivNIC.cf"

cluster rac_cluster101 (
    UserNames = { admin = bopHo }
    Administrators = { admin }
    UseFence = SCSI3
    )

system galaxy (
    )

system nebula (
    )

group oradb1_grp (
    SystemList = { galaxy = 0, nebula = 1 }
    AutoFailOver = 0
    Parallel = 1
    AutoStartList = { galaxy, nebula }
    )

    Oracle ora1 (
        Critical = 0
        Sid @galaxy = vrts1
        Sid @nebula = vrts2
        Owner = oracle
        Home = "/app/oracle/orahome"
        StartUpOpt = "SRVCTLSTART"
```

```
        ShutDownOpt = "SRVCTLSTOP"
        )

    CFSMount oradata_mnt (
        Critical = 0
        MountPoint = "/oradata"
        BlockDevice = "/dev/vx/dsk/oradatadg/oradatavol"
        )

    CVMVolDg oradata_voldg (
        Critical = 0
        CVMDiskGroup = oradatadg
        CVMVolume = { oradatavol }
        CVMActivation = sw
        )

    requires group cvm online local firm
    ora1 requires oradata_mnt
    oradata_mnt requires oradata_voldg

group cvm (
    SystemList = { galaxy = 0, nebula = 1 }
    AutoFailOver = 0
    Parallel = 1
    AutoStartList = { galaxy, nebula }
    )

    Application cssd (
        Critical = 0
        StartProgram = "/opt/VRTSvcs/rac/bin/cssd-online"
        StopProgram = "/opt/VRTSvcs/rac/bin/cssd-offline"
        CleanProgram = "/opt/VRTSvcs/rac/bin/cssd-clean"
        MonitorProgram = "/opt/VRTSvcs/rac/bin/cssd-monitor"
        OnlineRetryLimit = 20
        )

    CFSMount ocrvote_mnt (
        Critical = 0
        MountPoint = "/ocrvote"
        BlockDevice = "/dev/vx/dsk/ocrvotedg/ocrvotevol"
        )

    CVMVolDg ocrvote_voldg (
        Critical = 0
        CVMDiskGroup = ocrvotedg
        CVMVolume = { ocrvotevol }
        CVMActivation = sw
        )

    CFSfsckd vxfsckd (
        )
```

```
CVMCluster cvm_clus (
    CVMClustName = rac_cluster101
    CVMNodeId = { galaxy = 0, nebula = 1 }
    CVMTransport = gab
    CVMTimeout = 200
    )

CVMVxconfigd cvm_vxconfigd (
    Critical = 0
    CVMVxconfigdArgs = { syslog }
    )

PrivNIC ora_priv (
    Critical = 0
    Device = { en1 = 0, en2 = 1}

    Address@galaxy = "192.168.12.1"
    Address@nebula = "192.168.12.2"
    NetMask = "255.255.240.0"
            )

cssd requires ocrvote_mnt
cssd requires ora_priv
ocrvote_mnt requires ocrvote_voldg
ocrvote_mnt requires vxfsckd
ocrvote_voldg requires cvm_clus
vxfsckd requires cvm_clus
cvm_clus requires cvm_vxconfigd
```

Oracle 10g configuration for CVM/VVR primary site

Configuration details:

- Named: cvmvvr_primary_main.cf
- More general purpose, can have multiple Oracle databases

```
include "types.cf"
include "CFSTypes.cf"
include "CVMTypes.cf"
include "OracleTypes.cf"
include "PrivNIC.cf"
include "VVRTypes.cf"

cluster rac_cluster101 (
    UserNames = { admin = bopHo }
    ClusterAddress = "10.10.10.101"
    Administrators = { admin }
    UseFence = SCSI3
    )

remotecluster rac_cluster102 (
    ClusterAddress = "10.11.10.102"
    )

heartbeat Icmp (
    ClusterList = { rac_cluster102 }
    Arguments @rac_cluster102 = { "10.11.10.102" }
    )

system galaxy (
    )

system nebula (
    )

group ClusterService (
    SystemList = { galaxy = 0, nebula = 1 }
    AutoStartList = { galaxy, nebula }
    OnlineRetryLimit = 3
    OnlineRetryInterval = 120
    )

    Application wac (
        StartProgram = "/opt/VRTSvcs/bin/wacstart"
        StopProgram = "/opt/VRTSvcs/bin/wacstop"
        MonitorProcesses = { "/opt/VRTSvcs/bin/wac" }
        RestartLimit = 3
        )

    IP gcoip (
```

```
        Device = en0

        Address = "10.10.10.101"
        NetMask = "255.255.240.0"
        )

    NIC csgnic (
        Device = en0
        )

    gcoip requires csgnic
    wac requires gcoip

group RVGgroup (
    SystemList = { galaxy = 0, nebula = 1 }
    Parallel = 1
    AutoStartList = { galaxy, nebula }
    )

    CVMVolDg racdata_voldg (
        CVMDiskGroup = oradatadg
        CVMActivation = sw
        )

    RVGShared racdata_rvg (
        RVG = rac1_rvg
        DiskGroup = oradatadg
        )

    requires group cvm online local firm
    racdata_rvg requires racdata_voldg

group cvm (
    SystemList = { galaxy = 0, nebula = 1 }
    AutoFailOver = 0
    Parallel = 1
    AutoStartList = { galaxy, nebula }
    )

    Application cssd (
        Critical = 0
        StartProgram = "/opt/VRTSvcs/ops/bin/cssd-online"
        StopProgram = "/opt/VRTSvcs/ops/bin/cssd-offline"
        CleanProgram = "/opt/VRTSvcs/ops/bin/cssd-clean"
        MonitorProgram = "/opt/VRTSvcs/ops/bin/cssd-monitor"
        OnlineRetryLimit = 20
        )

    CFSfsckd vxfsckd (
        )

    CVMCluster cvm_clus (
```

```
        CVMClustName = rac_cluster101
        CVMNodeId = { galaxy = 0, nebula = 1 }
        CVMTransport = gab
        CVMTimeout = 200
        )

    CFSMount ocrvote_mnt (
        Critical = 0
        MountPoint = "/ocrvote"
        BlockDevice = "/dev/vx/dsk/ocrvotedg/ocrvotevol"
        )

    CVMVolDg ocrvote_voldg (
        Critical = 0
        CVMDiskGroup = ocrvotedg
        CVMVolume = { ocrvotevol }
        CVMActivation = sw
        )

    CVMVxconfigd cvm_vxconfigd (
        Critical = 0
        CVMVxconfigdArgs = { syslog }
        )

    PrivNIC ora_priv (
        Critical = 0
        Device = { en1 = 0, en2 = 1}

        Address@galaxy = "192.168.12.1"
        Address@nebula = "192.168.12.2"
        NetMask = "255.255.240.0"
        )
    cssd requires ocrvote_mnt
    cssd requires ora_priv
    ocrvote_mnt requires ocrvote_voldg
    ocrvote_mnt requires vxfsckd
    ocrvote_voldg requires cvm_clus
    vxfsckd requires cvm_clus
    cvm_clus requires cvm_vxconfigd

group oradb1_grp (
    SystemList = { galaxy = 0, nebula = 1 }
    Parallel = 1
    ClusterList = { rac_cluster101 = 0, rac_cluster102 = 1 }
    OnlineRetryInterval = 300
    ClusterFailOverPolicy = Manual
    AutoStartList = { galaxy, nebula }
    Authority = 1
    )

    CFSMount oradata_mnt (
        MountPoint = "/oradata"
```

```
        BlockDevice = "/dev/vx/dsk/oradatadg/oradatavol"
        )

    Oracle ora1 (
        Critical = 0
        Sid @galaxy = vrts1
        Sid @nebula = vrts2
        Owner = oracle
        Home = "/app/oracle/orahome"
        StartUpOpt = SRVCTLSTART
        ShutDownOpt = SRVCTLSTOP
        )

    RVGSharedPri ora_vvr_sharedpri (
        RvgResourceName = racdata_rvg
        OnlineRetryLimit = 0
        )

    requires group RVGgroup online local firm
    ora1 requires oradata_mnt
    oradata_mnt requires ora_vvr_sharedpri

group rlogowner (
    SystemList = { galaxy = 0, nebula = 1 }
    AutoStartList = { galaxy, nebula }
    OnlineRetryLimit = 2
    )

    IP logowner_ip (
        Device = en0
        Address = "10.10.9.101"
        NetMask = "255.255.240.0"
        )

    NIC nic (
        Device = en0
        )

    RVGLogowner logowner (
        RVG = rac1_rvg
        DiskGroup = oradatadg
        )

    requires group RVGgroup online local firm
    logowner requires logowner_ip
    logowner_ip requires nic
```

Oracle 10g configuration for CVM/VVR secondary site

Configuration details:

- Named: cvmvvr_secondary_main.cf
- More general purpose, can have multiple Oracle databases

```
include "types.cf"
include "CFSTypes.cf"
include "CVMTypes.cf"
include "OracleTypes.cf"
include "PrivNIC.cf"
include "VVRTypes.cf"

cluster rac_cluster102 (
    UserNames = { admin = bopHo }
    ClusterAddress = "10.11.10.102"
    Administrators = { admin }
    UseFence = SCSI3
    )

remotecluster rac_cluster101 (
    ClusterAddress = "10.10.10.101"
    )

heartbeat Icmp (
    ClusterList = { rac_cluster101 }
    Arguments @rac_cluster101 = { "10.10.10.101" }
    )

system mercury (
    )

system jupiter (
    )

group ClusterService (
    SystemList = { mercury = 0, jupiter = 1 }
    AutoStartList = { mercury, jupiter }
    OnlineRetryLimit = 3
    OnlineRetryInterval = 120
    )

    Application wac (
        StartProgram = "/opt/VRTSvcs/bin/wacstart"
        StopProgram = "/opt/VRTSvcs/bin/wacstop"
        MonitorProcesses = { "/opt/VRTSvcs/bin/wac" }
        RestartLimit = 3
        )

    IP gcoip (
        Device = en0
        Address = "10.11.10.102"
        NetMask = "255.255.240.0"
        )
```

```
    NIC csgnic (
        Device = en0
        )

    gcoip requires csgnic
    wac requires gcoip

group RVGgroup (
    SystemList = { mercury = 0, jupiter = 1 }
    Parallel = 1
    AutoStartList = { mercury, jupiter }
    )

    CVMVolDg racdata_voldg (
        CVMDiskGroup = oradatadg
        CVMActivation = sw
        )

    RVGShared racdata_rvg (
        RVG = rac1_rvg
        DiskGroup = oradatadg
        )

    requires group cvm online local firm
    racdata_rvg requires racdata_voldg

group cvm (
    SystemList = { mercury = 0, jupiter = 1 }
    AutoFailOver = 0
    Parallel = 1
    AutoStartList = { mercury, jupiter }
    )

    Application cssd (
        Critical = 0
        StartProgram = "/opt/VRTSvcs/ops/bin/cssd-online"
        StopProgram = "/opt/VRTSvcs/ops/bin/cssd-offline"
        CleanProgram = "/opt/VRTSvcs/ops/bin/cssd-clean"
        MonitorProgram = "/opt/VRTSvcs/ops/bin/cssd-monitor"
        OnlineRetryLimit = 20
        )

    CFSfsckd vxfsckd (
        )

    CVMCluster cvm_clus (
        CVMClustName = rac_cluster102
        CVMNodeId = { mercury = 0, jupiter = 1 }
        CVMTransport = gab
        CVMTimeout = 200
        )
```

```
    CFSMount ocrvote_mnt (
        Critical = 0
        MountPoint = "/ocrvote"
        BlockDevice = "/dev/vx/dsk/ocrvotedg/ocrvotevol"
        )

    CVMVolDg ocrvote_voldg (
        Critical = 0
        CVMDiskGroup = ocrvotedg
        CVMVolume = { ocrvotevol }
        CVMActivation = sw
        )

    CVMVxconfigd cvm_vxconfigd (
        Critical = 0
        CVMVxconfigdArgs = { syslog }
        )

    PrivNIC ora_privnic (
        Critical = 0
        Device = { en1 = 0, en2 = 1}

        Address@galaxy = "192.168.12.1"
        Address@nebula = "192.168.12.2"
        NetMask = "255.255.240.0"
                )
    cssd requires ocrvote_mnt
    cssd requires ora_priv
    ocrvote_mnt requires ocrvote_voldg
    ocrvote_mnt requires vxfsckd
    ocrvote_voldg requires cvm_clus
    vxfsckd requires cvm_clus
    cvm_clus requires cvm_vxconfigd


group oradb1_grp (
    SystemList = { mercury = 0, jupiter = 1 }
    Parallel = 1
    ClusterList = { rac_cluster101 = 0, rac_cluster102 = 1 }
    OnlineRetryInterval = 300
    ClusterFailOverPolicy = Manual
    Authority = 1
    AutoStartList = { mercury, jupiter }
    )

    CFSMount oradata_mnt (
        MountPoint = "/oradata"
        BlockDevice = "/dev/vx/dsk/oradatadg/oradatavol"
        )

    Oracle ora1 (
```

```
        Critical = 0
        Sid @mercury = vrts1
        Sid @jupiter = vrts2
        Owner = oracle
        Home = "/app/oracle/orahome"
        StartUpOpt = SRVCTLSTART
        ShutDownOpt = SRVCTLSTOP
        )

    RVGSharedPri ora_vvr_sharedpri (
        RvgResourceName = racdata_rvg
        OnlineRetryLimit = 0
        )

    requires group RVGgroup online local firm
    ora1 requires oradata_mnt
    oradata_mnt requires ora_vvr_sharedpri

group rlogowner (
    SystemList = { mercury = 0, jupiter = 1 }
    AutoStartList = { mercury, jupiter }
    OnlineRetryLimit = 2
    )

    IP logowner_ip (
        Device = en0
        Address = "10.11.9.102"
        NetMask = "255.255.240.0"
        )

    NIC nic (
        Device = en0
        )

    RVGLogowner logowner (
        RVG = rac1_rvg
        DiskGroup = oradatadg
        )

    requires group RVGgroup online local firm
    logowner requires logowner_ip
    logowner_ip requires nic
```

Appendix B

Creating a starter database

Creating a database for Oracle9i

Create a database tablespaces for Oracle9i using one of two options:

- Option 1: on shared raw volumes
- Option 2: on cluster file system CFS

Creating starter database tablespaces on raw volumes (option 1)

To create a starter database on raw VxVM volumes, prepare the shared storage for the database tablespaces.

1. Log in as `root` user.
2. On the master node, create a shared disk group:

   ```
   # vxdg -s init oradatadg HDS0_20
   ```

3. Create a volume in the shared group for *each* of the required tablespaces:

   ```
   # vxassist -g oradatadg make VRT_galaxy 1000M
   # vxassist -g oradatadg make VRT_spfile1 10M
   .
   .
   ```

4. Define the access mode and permissions for the volumes storing the Oracle data. For each volume listed in `$ORACLE_HOME/raw_config`, use the `vxedit`(1M) command:

   ```
   vxedit -g disk_group set group=group user=user mode=660 volume
   ```

   For example:

   ```
   # vxedit -g oradatadg set group=dba user=oracle mode=660 \
   VRT_galaxy
   ```

In this example, VRT_galaxy is the name of one of the volumes. Repeat the command to define access mode and permissions for each volume in the oradatadg.

Running the dbca Utility for Raw VxVM Volumes

Use the dbca utility on the master node to create a general-purpose database on raw VxVM volumes. The utility is a graphical user interface and requires setting the DISPLAY environment variable.

1. Make sure an oracle account is created on all nodes.
2. Verify that rsh works among all nodes under the oracle account.
3. Make sure the file configured for srvm is accessible to the oracle user.
4. Start the Oracle Real Application Clusters Manageability daemon on each node:

   ```
   $ gsdctl start
   ```

5. From one node, log in as oracle user.
6. For Oracle9i users, create and start the Oracle listener using the NETCA utility (refer to the Oracle9i Installation Guide):

   ```
   $ netca
   ```

7. Run the dbca utility to create the database. Refer to Oracle documentation for details on the dbca utility.

Creating starter database tablespaces on CFS using dbca (option 2)

When creating a starter database on a cluster file system, create a disk group and volume for the tablespaces before running the dbca utility.

1. Create a disk group (for example, oradatadg):

   ```
   # vxdg -s init oradatadg HDS0_20
   ```

2. Create a single shared volume (for example, rac_vol1) large enough to contain a file system for all the tablespaces. The tablespaces require about 6.8 GB:

   ```
   # vxassist -g oradatadg make rac_vol1 6600M
   ```

3. Start the volume in the disk group:

   ```
   # vxvol -g oradatadg startall
   ```

4. Create a VxFS file system in this volume. From one node, type:

   ```
   # mkfs -V vxfs /dev/vx/rdsk/oradatadg/oradatavol
   ```

5. Create a mount point for the shared file system:

   ```
   # mkdir /oradata
   ```

6. From the same node, mount the file system:

```
# mount -V vxfs -o cluster /dev/vx/dsk/oradatadg/oradatavol \
/oradata
```

7 Set "oracle" as the owner of the file system and "775" as the permissions:

```
# chown oracle:dba /oradata
# chmod 755 /oradata
```

8 On the other node(s), complete step 5 through step 7.

Running the dbca Utility for Cluster File System

As `oracle` user, use the `dbca` utility on the master node to create a general purpose database on a cluster file system. This utility is a graphical user interface and requires setting the `DISPLAY` environment variable.

1 Make sure an `oracle` account is created on all nodes.

2 Verify that `rsh` works among all the nodes under the `oracle` account.

3 Examine the `/var/opt/oracle/srvConfig.loc` configuration file and note the path defined for the `srvconfig_loc` variable. For example:

```
srvconfig_loc=/db/srvm.ora
```

where `/db/srvm.ora` is the path to the SRVM configuration file.

a Use the `ls -l` *pathname* command to display the long listing for the defined path. The file is initialized if the output shows a file size greater than zero bytes.

b If the variable is not initialized (size is 0), type:

```
$ srvconfig -init
```

c If the path is not yet defined (listing does not appear), initialize the variable after using the `touch` command to create it:

```
$ touch /db/srvm.ora
$ srvconfig -init
```

4 Start the Oracle Real Application Clusters Manageability daemon on each node:

```
$ gsdctl start
```

5 From one node, log in as `oracle` user.

6 For Oracle9i, create and start the Oracle listener using `NETCA` utility (refer to the Oracle9i Installation Guide):

```
$ netca
```

7 Run the `dbca` utility. When starting the utility with a cluster file system, use the `-datafileDestination` option to specify the mount point. For example:

```
$ dbca -datafileDestination /oradata
```

Refer to the Oracle documentation for details on the `dbca` utility.

Creating a database for Oracle 10g

Create a database tablespaces for Oracle9i using one of two options:

- Option 1: on shared raw volumes
- Option 2: on cluster file system CFS

Before you begin, take note of the these prerequisites:

- CRS daemons must be running. To verify the status of CRS, type:
  ```
  # $CRS_HOME/bin/crs_stat
  ```
- Use the `ping` command to verify that all private IP addresses on each node are up.

Creating database tablespaces shared on raw volumes (option 1)

1. Log in as `root` user.
2. On the master node, create a shared disk group:
   ```
   # vxdg -s init oradatadg HDS0_20
   ```
3. Create a volume in the shared group for *each* of the required tablespaces. Refer to the Oracle documentation to determine the tablespace requirements. For example, type:
   ```
   # vxassist -g oradatadg make VRT_galaxy 1000M
   # vxassist -g oradatadg make VRT_spfile1 10M
   .
   .
   ```
4. Define the access mode and permissions for the volumes storing the Oracle data. For *each* volume listed in `$ORACLE_HOME/raw_config`, use the `vxedit`(1M) command:
   ```
   vxedit -g disk_group set group=group user=user mode=660 volume
   ```
   For example:
   ```
   # vxedit -g oradatadg set group=dba user=oracle mode=660 \
   VRT_galaxy
   ```
   In this example, `VRT_galaxy` is the name of one of the volumes. Repeat the command to define access mode and permissions for each volume in the `oradatadg`.
5. Create the database using Oracle documentation.

Creating database tablespaces shared on CFS (option 2)

If you plan to use a cluster file system to store the Oracle database, use the following procedure to create the file system.

1. Create a disk group (for example, `oradatadg`):

```
# vxdg -s init oradatadg HDS0_20
```

2 Create a single shared volume (for example, `oradatavol`) that is large enough to contain a file system for all tablespaces. Refer to the Oracle documentation for tablespace sizes). Assuming 6.8 GB are required for the tablespaces, type:

```
# vxassist -g oradatadg make oradatavol 6800M
```

3 Start the volume in the disk group:

```
# vxvol -g oradatadg startall
```

4 Create a VxFS file system in this volume. From one node, type:

```
# mkfs -V vxfs /dev/vx/rdsk/oradatadg/oradatavol
```

5 Create a mount point for the shared file system:

```
# mkdir /oradata
```

6 From the same node, mount the file system:

```
# mount -V vxfs -o cluster /dev/vx/dsk/oradatadg/oradatavol \
/oradata
```

7 Set “oracle” as the owner of the file system, and set “755” as the permissions:

```
# chown oracle:oinstall /oradata
# chmod 755 /oradata
```

8 On the other node(s), complete step 5 through step 7.

9 Refer to Oracle documentation to create the database.

Appendix C

Agent reference

This Appendix describes the entry points and the attributes :

Use this information to make necessary changes to the configuration.Refer to the *Veritas Cluster Server User's Guide* for information on how to modify the VCS configuration.

CVMCluster agent

The CVMCluster resource is configured automatically during installation. The CVMCluster agent controls system membership on the cluster port associated with VxVM in a cluster.

CVMCluster agent, entry points

The following table describes the entry points used by the CVMCluster agent

Entry Point	Description
Online	Joins a node to the CVM cluster port. Enables the Volume Manager cluster functionality by autoimporting shared disk groups.
Offline	Removes a node from the CVM cluster port.
Monitor	Monitors the node's CVM cluster membership state.

CVMCluster agent type

The following table describes the user-modifiable attributes of the CVMCluster resource type

.

Attribute	Dimension	Description
CVMClustName	string-scalar	Name of the cluster.
CVMNodeAddr	string-association	List of host names and IP addresses.
CVMNodeId	string-association	An associative list consisting of the name of the system and the system's LLT ID number.
CVMTransport	string-scalar	Specifies cluster messaging mechanism. Default = gab **Note:** Do not change this value.
PortConfigd	integer-scalar	Port number used by CVM for vxconfigd-level communication.
PortKmsgd	integer-scalar	Port number used by CVM for kernel-level communication.
CVMTimeout	integer-scalar	Timeout in seconds used for CVM cluster reconfiguration. Default = 200

CVMCluster agent type definition

The following type definition is included in the file, `CVMTypes.cf`. Note that the `CVMNodeAddr`, `PortConfigd`, and `PortKmsgd` attributes are not used in an SF Oracle RAC environment because GAB, the required cluster communication messaging mechanism, does not use them.

```
type CVMCluster (
        static int InfoTimeout = 0
        static int NumThreads = 1
        static int OnlineRetryLimit = 2
        static int OnlineTimeout = 400
        static str ArgList[] = { CVMTransport, CVMClustName,
           CVMNodeAddr, CVMNodeId, PortConfigd, PortKmsgd,
           CVMTimeout }
        NameRule = ""
        str CVMClustName
        str CVMNodeAddr{}
        str CVMNodeId{}
        str CVMTransport
        int PortConfigd
        int PortKmsgd
        int CVMTimeout
)
```

CVMCluster agent sample configuration

The following is an example definition for the CVMCluster service group. See Appendix A, "Sample VCS configuration files for SF Oracle RAC" on page 425 for a more extensive `main.cf` example that includes the CVMCluster resource.

```
CVMCluster cvm_clus (
        Critical = 0
        CVMClustName = RACCluster1
        CVMNodeId = { galaxy = 0, nebula = 1 }
        CVMTransport = gab
        CVMTimeout = 200
)
```

CVMVxconfigd Agent

The CVMVxconfigd agent is responsible for starting and monitoring the `vxconfigd` daemon. The `vxconfigd` daemon maintains disk and disk group configurations, communicates configuration changes to the kernel, and modifies configuration information stored on disks. CVMVxconfigd is always required in the CVM service group.

The CVMVxconfigd is an OnOnly agent; that is, the agent starts it when the cluster starts up and it is always restarted by VCS whenever necessary. This is specified by default in the `Operations` attribute.

It is highly recommended that the `vxconfigd` daemon be started with the `syslog` option, which enables logging of debug messages. The `syslog` option is configured for the CVMVxconfigd agent during installation.

CVMVxconfigd agent, entry points

Entry Point	Description
Online	Starts the `vxconfigd` daemon
Offline	N/A
Monitor	Monitors whether `vxconfigd` daemon is running

CVMVxconfigd agent type

Attribute	Dimension	Description
CVMVxconfigdArgs	keylist	Includes the list of arguments to be sent to the `online` entry point. It is highly recommended that the `syslog` option always be specified.

CVMVxconfigd type definition

The following type definition is included in the file, `CVMTypes.cf`.

```
type CVMVxconfigd (
        static int FaultOnMonitorTimeouts = 2
        static int RestartLimit = 5
        static str ArgList[] { CVMVxconfigdArgs }
        static str Operations = OnOnly
        keylist CVMVxconfigdArgs
)
```

Sample CVMVxconfigd agent configuration

The following is an example definition for the CVMVxconfigd resource in the CVM service group. See also Appendix A, "Sample VCS configuration files for SF Oracle RAC".

```
CVMVxconfigd cvm_vxconfigd (
        Critical = 0
        CVMVxconfigdArgs = { syslog }
)
.
.
cvm_clus requires cvm_vxconfigd
// resource dependency tree
//
// group cvm
// {
// CVMCluster cvm_clus
//     {
//     CVMVxconfigd cvm_vxconfigd
//     }
// }
```

CVMVolDg and CFSMount resources

The CVMVolDg agent represents and controls CVM disk groups and the CVM volumes within the disk groups. Because of the global nature of the CVM disk groups and the CVM volumes, they are imported only once on the CVM master node.

Configure the CVMVolDg agent for each disk group used by an Oracle service group. A disk group must be configured to only one Oracle service group.If cluster file systems are used for the database, configure the CFSMount agent for each volume in the disk group.

CVMVolDg agent entry points

The following table describes the entry points used by the CVMVolDg agent.

Entry Point	Description
Online	If the system is the CVM master and the disk group is not imported, the online entry point imports the disk group and starts all the volumes in the shared disk group. It then sets the disk group activation mode to shared-write as long as the `CVMActivation` attribute is set to `sw`. The activation mode is set on both slave and master systems.
Offline	Clears internal state.
Monitor	Monitors the specified critical volumes in the disk group. The volumes to be monitored are specified by the `CVMVolume` attribute. In an SF Oracle RAC environment, at least one volume in a disk group must be specified.
Clean	Clears internal state.

CVMVolDg agent type attribute descriptions

The following table describes the user-modifiable attributes of the CVMVolDg resource type.

Attribute	Dimension	Description
CVMDiskGroup	string-scalar	Names the disk group.
CVMVolume	string-keylist	Lists the critical volumes in the disk group. At least one volume in the disk group must be specified.
CVMActivation	string-scalar	Sets the activation mode for the disk group. Default = `sw`

CVMVolDg agent type definition

The CVMVolDg type definition is included in the `CVMTypes.cf` file, installed by the `installsfrac` utility.

```
type CVMVolDg (
        static keylist RegList = { CVMActivation }
        static str ArgList[] = { CVMDiskGroup, CVMVolume,
            CVMActivation }
        str CVMDiskGroup
        keylist CVMVolume[]
        str CVMActivation
        temp int voldg_stat
)
```

Sample CVMVolDg agent configuration

Each Oracle service group requires a CVMVolDg resource type to be defined. Refer to "Sample VCS configuration files for SF Oracle RAC" on page 425 to see CVMVolDg defined in a more extensive example.

```
CVMVolDg ora_voldg (
        CVMDiskGroup = oradatadg
        CVMVolume = { oradata1, oradata2 }
        CVMActivation = sw
)
```

CFSMount agent entry points

The CFSMount agent brings online, takes offline, and monitors a cluster file system mount point. The agent executable is `/opt/VRTSvcs/bin/CFSMount/CFSMountAgent`. The CFSMount type definition is in the file `/etc/VRTSvcs/conf/config/CFSTypes.cf`.

Entry Point	Description
Online	Mounts a block device in cluster mode.
Offline	Unmounts the file system, forcing unmount if necessary, and sets primary to secondary if necessary.
Monitor	Determines if the file system is mounted. Checks mount status using the fsclustadm command.
Clean	A null operation for a cluster file system mount.

CFSMount agent type, attribute descriptions

The table lists user-modifiable attributes of the CFSMount Agent resource type.

Attribute	Dimension	Description
MountPoint	string-scalar	Directory for the mount point.
BlockDevice	string-scalar	Block device for the mount point.
NodeList	string-keylist	List of nodes on which to mount. If NodeList is NULL, the agent uses the service group system list.

MountOpt (optional)	string-scalar	Options for the mount command. To create a valid MountOpt attribute string:
		■ Use the VxFS type-specific options only.
		■ Do not use the -o flag to specify the VxFS-specific options.
		■ Do not use the **-v** vxfs file system type option.
		■ The cluster option is not required.
		■ Specify options in comma-separated list as in these examples:
		`ro`
		`ro,cluster`
		`blkclear,mincache=closesync`

Policy (optional)	string-scalar	List of nodes to assume the primaryship of the cluster file system if the primary fails. If set to NULL or if none of the hosts specified in the list is active when the primary fails, a node is randomly selected from the set of active nodes to assume primaryship.

CFSMount agent type definition

The CFSMount agent type definition is included in the `CFSTypes.cf` file, installed by the `installsfrac` utility.

```
type CFSMount (
        static keylist RegList = { MountOpt, Policy, NodeList }
        static int FaultOnMonitorTimeouts = 1
        static int InfoTimeout = 0
        static int OnlineRetryLimit = 16
        static int OnlineWaitLimit = 0
        static str ArgList[] = { MountPoint, BlockDevice,
            MountOpt }
        NameRule = resource.MountPoint
        str MountPoint
        str MountType
        str BlockDevice
        str MountOpt
        keylist NodeList
        keylist Policy
        temp str Primary
        str SetPrimary
        str RemountRes
        str ForceOff
        )
```

Sample CFSMount agent configuration

Each Oracle service group requires a CFSMount resource type to be defined. Refer to Appendix A, "Sample VCS configuration files for SF Oracle RAC" on page 425 to see CFSMount defined in a more extensive example.

```
CFSMount ora_mount (
        MountPoint = "/oradata"
        BlockDevice = "/dev/vx/dsk/oradatadg/oradatavol1"
        Primary = nebula
)
```

PrivNIC agent

The PrivNIC resource is used to maintain a “private IP address” that is locally highly available on LLT Ethernet interfaces. Such private IP addresses are required by the CRS daemons in Oracle10g to provide communication.

The PrivNIC agent relies on LLT to monitor the interfaces. It queries LLT to count the number of visible nodes on each of the LLT interfaces.

PrivNIC agent: monitor entry point

The following table describes the `monitor` entry point used by the PrivNIC agent. Only the monitor entry point is required because the resource is persistent.

Entry Point	Description
Monitor	Queries LLT to make a list of nodes visible on every LLT network interface. It applies various filters to the list to arrive at a most desired failover decision and calculates a “winner” device on which to configure the IP address. The “winner” is compared to the currently active device where the IP is currently configured; if the active and winner device are different, the agent fails over the device.

PrivNIC agent: type attribute descriptions

The following table describes the user-modifiable attributes of the PrivNIC resource type

Required Attributes

Attribute	Dimension	Description

Device	string - association	Specifies the network interface device as shown by the "ifconfig" command and the "network-id" associated with the interface. Network-ids of the interfaces connected to the same physical network must match. The interface with the lower network-id has the higher preference for failover. At least one interface device *must* be specified. Example:
		`Device@galaxy = {en1=0, en2=1, en3=2}`
		`Device@nebula = {en1=0, en2=1, en3=2}`

Address	string-scalar	The numerical private IP address. For example: `Address = "192.11.12.13"`
NetMask	string - association	The numerical netmask for the private IP address. For example: `Address = "255.255.255.0"`

Optional Attributes

Attribute	Dimension	Description
DeviceTag	string - association	Associates an LLT device "tag" with device via the network-id. If an LLT device tag (as specified in the `/etc/llttab` file) differs from the name of the network interface as shown in "ifconfig," then DeviceTag must be specified for that interface.
GabPort	string-scalar	A single lower-case letter specifying the name of the GAB port to be used for filtering. "o" is the default. NULL disables GAB port filtering. Example: `GabPort = "b"`

Attribute	Dimension	Description
UseVirtualIP	integer-scalar	The default is 0, which specifies that the agent use the physical interface for configuring the private IP address when possible. The value 1 specifies that the agent always use the virtual interface for configuring the private IP address. The value 2 (which includes the functionality of the value 1) specifies the agent should complain if the private IP address already exists on a physical interface.
UseSystemList	integer-scalar	The value 1 specifies that the agent use the SystemList of the service group to filter the node list. Default = 0.
ExcludeNode	integer-vector	List of nodes to permanently excluded from calculation.

PrivNIC agent: type definition

The following shows the content of the `PrivNIC.cf` file:

```
type PrivNIC (
        static str ArgList[] = { Device, DeviceTag, Address,
           NetMask, UseVirtualIP, GabPort, UseSystemList,
           ExcludeNode }
        static int OfflineMonitorInterval = 60
        static int MonitorTimeout = 300
        static str Operations = None

        str Device{}
        str DeviceTag{}
        str Address = ""
        str NetMask = ""
        int UseVirtualIP = 0
        str GabPort = "o"
        int UseSystemList = 0
        int ExcludeNode[]
)
```

PrivNIC agent: sample configuration

The following is a sample configuration using the PrivNIC agent.

```
group cvm (
        SystemList = { galaxy = 0, nebula = 1 }
        AutoFailOver = 0
        Parallel = 1
        AutoStartList = { galaxy, nebula }
    )
        PrivNIC ora_priv (
        Device = { en1 = 0, en2 = 1, en3 = 5 }
```

```
    Address@galaxy = "192.11.12.13"
    Address@nebula = "192.11.12.14"
    NetMask = "255.255.255.0"
)
```

Configuring the Application agent to monitor CSSD

The `cssd` resource is optional. It monitors the Oracle 10*g* `cssd` process. The purpose of the `cssd` resource is to ensure that the dependency of `cssd` on the OCR and VOTE resources and the PrivNIC (optional) resource are satisfied. If the `cssd` resource is online and any of its dependencies are brought offline, the machine will reboot. This agent allows this behavior to be avoided since the dependencies will be enforced by VCS.

Note: VCS will not actually stop the CRS daemon. Instead, it will report an error message to the VCS engine log file if an offline is issued. Refer to the Oracle documentation to understand how to safely stop the CRS daemon.
VCS will not start the CRS daemon. It will wait for the daemon to start automatically upon system boot up. If CRS daemon is stopped, refer to the Oracle documentation to understand how to safely restart the daemon.

The `cssd` resource should use the Application agent. The name of the resource is up to the user. The following are required attributes of the `cssd` resource:

Attribute Name	Required Value
Critical	0
OnlineRetryLimit	20
StartProgram	/opt/VRTSvcs/rac/bin/cssd-online
StopProgram	/opt/VRTSvcs/rac/bin/cssd-offline
CleanProgram	/opt/VRTSvcs/rac/bin/cssd-clean
MonitorProgram	/opt/VRTSvcs/rac/bin/cssd-monitor

An example `main.cf` entry is as follows:

```
Application cssd-resource (
    Critical = 0
    StartProgram = "/opt/VRTSvcs/rac/bin/cssd-online"
    StopProgram = "/opt/VRTSvcs/rac/bin/cssd-offline"
    CleanProgram = "/opt/VRTSvcs/rac/bin/cssd-clean"
    MonitorProgram = "/opt/VRTSvcs/rac/bin/cssd-monitor"
    OnlineRetryLimit = 20
    )
```

Oracle agent functions

The Oracle agent monitors the database processes. Table C-1 lists the Oracle agent operations. The functions an agent performs are called entry points.

For more on the Veritas high availability agent for Oracle, see the *Veritas High Availability for Oracle Agent Guide*.

Table C-1 Oracle agent operations

Agent operation	Description
Online	Starts the Oracle database by using the following `svrmgrl` or `sqlplus` command: `startup force pfile=$PFile` The default Startup option is STARTUP_FORCE. You can also configure the agent to start the database using different Startup options for Oracle. See "Startup options" on page 464.
Offline	Stops the Oracle database with the specified options by using the following `svrmgrl` or `sqlplus` command: `shutdown immediate` The default Shutdown option is IMMEDIATE. You can also configure the agent to stop the database using different Shutdown options for Oracle. See "Shutdown options" on page 465.
Monitor	Verifies the status of the Oracle processes. The Oracle agent provides two levels of monitoring: basic and detail. See "Monitor options for Oracle agent" on page 465.
Clean	Forcibly stops the Oracle database by using the following `svrmgrl` or `sqlplus` command: `shutdown abort` If the process does not respond to the `shutdown` command, the agent scans the process table for processes associated with the configured instance and kills them.
Info	Provides static and dynamic information about the state of the database. See "Info entry point" on page 466.
Action	Performs predefined actions on a resource. See "Action entry point" on page 466.

Startup and shutdown options

You can specify Startup and Shutdown options for Oracle instances that are configured.

Startup options

Table C-2 lists the startup options that the agent supports.

Table C-2 Startup options

Option	Description
STARTUP_FORCE (Default)	Runs `startup force pfile='location_of_pfile'` if the pfile is configured. If the pfile is not configured, the agent runs `startup force`. It picks up the default parameter files from their default locations.
STARTUP	Runs `startup pfile='location_of_pfile'` if the pfile is configured. If the pfile is not configured, the agent picks up the default parameter files from their default locations and runs `startup`.
RESTRICTED	Starts the database in the RESTRICTED mode.
RECOVERDB	Performs a database recovery on instance startup.

CUSTOM	Uses a predefined SQL script (`start_custom_$SID.sql`) and runs custom startup options. The script must be in the /opt/VRTSagents/ha/bin/Oracle directory and must have access to the Oracle Owner OS user. If the file is not present, the agent logs an error message.
	With a custom script, the agent takes the following action:
	`sqlplus /nolog <<!`
	`connect / as sysdba;`
	`@start_custom_$SID.sql`
	`exit;`
	`!`

Shutdown options

Table C-3 lists the shutdown options that the agent supports.

Table C-3 Shutdown options

Option	Description
IMMEDIATE (Default)	Shuts down the Oracle instance by running `shutdown immediate`.
TRANSACTIONAL	Runs the `shutdown transactional` command. This option is valid only for database versions that support this option.
CUSTOM	Uses a predefined SQL script (`shut_custom_$SID.sql`) and runs custom shutdown options. The script must be in the /opt/VRTSagents/ha/bin/Oracle directory and must have access to the Oracle Owner OS user. If the file is not present, the agent shuts the agent down with the default option.

Monitor options for Oracle agent

The Oracle agent provides two levels of monitoring: basic and detail. By default, the agent does a basic monitoring.

Basic monitoring options

The basic monitoring mode has two options: Process check and Health check. Table C-4 describes the basic monitoring options.

Table C-4 Basic monitoring options

Option	Description
0 (Default)	Process check The agent scans the process table for the ora_dbw, ora_smon, ora_pmon, and ora_lgwr processes to verify that Oracle is running.
1	Health check (supported on Oracle 10*g* and later) The agent uses the Health Check APIs from Oracle to monitor the SGA and retrieve the information about the instance.

Detail monitoring

In the detail monitoring mode, the agent performs a transaction on a test table in the database to ensure that Oracle is functioning properly.

Info entry point

The Veritas high availability agent for Oracle supports the Info entry point, which provides static and dynamic information about the state of the database.

To invoke the Info entry point, type the following command:

The entry point retrieves the following static information:

- Version
- InstanceNo
- InstanceName
- DatabaseName
- HostName
- StartupTime
- Parallel
- Thread
- InstanceRole

The entry point retrieves the following dynamic information:

- InstanceStatus
- Logins
- OpenMode
- LogMode
- ShutdownPending
- DatabaseStatus
- Shared Pool Percent free
- Buffer Hits Percent
-

You can add additional attributes by adding sql statements to the file /opt/VRTSagents/ha/bin/Oracle/resinfo.sql. For example:

```
select 'static:HostName:'||host_name from v$instance;
select 'dynamic:ShutdownPending:'||shutdown_pending from
v$instance;
```

The format of the selected record must be as follows:

```
attribute_type:userkey_name:userkey_value
```

The variable *attribute_type* can take the value static and/or dynamic.

Action entry point

The Veritas high availability agent for Oracle supports the Action entry point, which enables you to perform predefined actions on a resource. Table C-5 describes the agent's predefined actions.

To perform an action on a resource, type the following command:

You can also add custom actions for the agent. For further information, refer to the *Veritas Cluster Server Agent Developer's Guide*.

Table C-5 Predefined agent actions

Action	Description
VRTS_GetInstanceName	Retrieves the name of the configured instance. You can use this option for the Oracle and the Netlsnr resources.
VRTS_GetRunningServices	Retrieves the list of processes monitored by the agent. You can use this option for the Oracle and the Netlsnr resources.

Table C-5 Predefined agent actions

Action	Description
DBRestrict	Changes the database session to enable the RESTRICTED mode.
DBUndoRestrict	Changes the database session to disable the RESTRICTED mode.
DBSuspend	Suspends a database.
DBResume	Resumes a suspended database.
DBTbspBackup	Backs up a tablespace; `actionargs` contains name of the tablespace to be backed up.

Netlsnr agent functions

The listener is a server process that listens to incoming client connection requests and manages traffic to the database. The Netlsnr agent brings the listener services online, monitors their status, and takes them offline. Table C-6 lists the Netlsnr agent operations.

Table C-6 Netlsnr agent operations

Agent operation	Description
Online	Starts the listener process by using the following command: `lsnrctl start $LISTENER`
Offline	Stops the listener process by using the following command: `lsnrctl stop $LISTENER` If the listener is configured with a password, the agent uses the password to stop the listener.
Monitor	Verifies the status of the listener process. The Netlsnr agent provides two levels of monitoring: basic and detail. ■ In the basic monitoring mode, the agent scans the process table for the `tnslsnr` process to verify the listener process is running. (Default) ■ In the detail monitoring mode, the agent uses the `lsnrctl status $LISTENER` command to verify the status of the Listener process.
Clean	Scans the process table for `tnslsnr $Listener` and kills it.

Table C-6 Netlsnr agent operations

Agent operation	Description
Action	Performs predefined actions on a resource. See “Action entry point” on page 466.

Appendix D

I/O fencing topics

This appendix includes additional topics related to I/O fencing, including descriptions of how to initialize disks, how to use options and methods of the vxfentsthdw command to test disks for SCSI-3 compliance, the vxfenadm command, and various I/O fencing behaviors to protect data in certain scenarios.

Initializing disks as VxVM disks

Install the driver and HBA card. Refer to the documentation from the vendor for instructions.

After you physically add shared disks to the nodes, you must initialize them as VxVM disks and verify that all the nodes see the same disk. Use the example procedure; see the *Veritas Volume Manager Administrator's Guide* for more information on adding and configuring disks.

To initialize disks

1. Make the new disks recognizable. On each node, enter:

   ```
   # lsdev -Cc disk
   ```

2. If the ASL for the array you are adding is not installed, obtain and install it on each node before proceeding. The ASL for the supported storage device you are adding is available from the disk array vendor or Symantec technical support.

3. Verify that the Array Support Library (ASL) for the disk array is installed on each of the nodes. Run the following command on each node and examine the output to verify the installation of ASL. The following output is a sample:

   ```
   # vxddladm listsupport all
   LIBNAME              VID
   ==============================================================
   libvxCLARiiON.so     DGC
   ```

```
libvxcscovrts.so      CSCOVRTS
libvxemc.so           EMC
```

4 Scan all disk drives and their attributes, update the VxVM device list, and reconfigure DMP with the new devices. Type:

```
# vxdisk scandisks
```

See the Veritas Volume Manager documentation for details on adding and configuring disks.

5 Determine the VxVM name by which a disk drive (or LUN) is known.

In the following example, a disk with the AIX device name /dev/rhdisk75 is identified by VxVM as EMC0_17:

```
# vxdmpadm getdmpnode nodename=hdisk75
NAME      STATE    ENCLR-TYPE   PATHS  ENBL   DSBL  ENCLR-NAME
============================================================
EMC0_17  ENABLED  EMC           1      1      0     EMC0
```

Notice that in the example command, the AIX device name for the block device was used.

You can, as an option, run the command `vxdisk list` *`vxvm_device_name`* to see additional information about the disk, including the AIX device name. For example:

```
# vxdisk list EMC0_17
```

6 To initialize the disks as VxVM disks, use one of the following methods:

- Use the interactive vxdiskadm utility to initialize the disks as VxVM disks.

  For more information see the *Veritas Volume Manager Administrator's Guide*.

- Use the `vxdisksetup` command to initialize a disk as a VxVM disk.

  `vxdisksetup -i` *`device_name`* `format=cdsdisk`

  The example specifies the CDS format:

  ```
  # vxdisksetup -i EMC0_17 format=cdsdisk
  ```

  Repeat this command for each disk you intend to use as a coordinator disk.

vxfentsthdw options and methods

The vxfentsthdw basic operation is described earlier in this guide.

See"Checking shared disks for SCSI-3 support" on page 89.

You can use the vxfentsthdw 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 disks.

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 `vxfentsthdw`(1M) manual page.

General guidelines for using vxfentsthdw

- 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 vxfentsthdw utility with the `-n` option.
  After completing the testing process, 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 `vxfenadm -i diskpath` command to verify a disk's serial number.
  "Verifying the nodes see the same disk" on page 105
- 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 `-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/rhdisk75 is ready to be configured for I/O Fencing
  on node north
  ```
- If the disk you intend to test has existing SCSI-3 registration keys, the test issues a warning before proceeding.

Table D-1 describes various options the utility provides to test storage devices.

Table D-1 vxfentsthdw options

vxfentsthdw option	Description	When to use
-n	Use /bin/rsh.	Use when `rsh` is used for communication.

Table D-1 vxfentsthdw options

vxfentsthdw option	Description	When to use
-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.
-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.	By default, the script picks up the OS paths for disks in the disk group. If you want the script to use the DMP path, 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.
-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.
-f *filename*	Utility tests system/device combinations listed in a text file. May be used with -r and -t options.	For testing several disks.
-g *disk_group*	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.

Testing the coordinator disk group using vxfentsthdw -c

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

- From the node north, the disks are /dev/rhdisk75, /dev/rhdisk76, and /dev/rhdisk77.
- From the node south, the same disks are /dev/rhdisk80, /dev/rhdisk81, and /dev/rhdisk82.

Note: To test the coordinator disk group using the vxfentsthdw utility, the utility requires that the coordinator disk group, vxfencoorddg, 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:
   north
   Enter the second node of the cluster:
   south
   ```

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/rhdisk75 have PASSED.
   The disk is now ready to be configured for I/O Fencing on node
   north as a COORDINATOR DISK.

   ALL tests on the disk /dev/rhdisk80 have PASSED.
   The disk is now ready to be configured for I/O Fencing on node
   south 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.

If you need to replace a disk in an active coordinator disk group, refer to the procedure in the troubleshooting chapter.

"Adding or removing coordinator disks" on page 416

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.
   "Initializing disks as VxVM disks" on page 469
   "Configuring coordinator disks" on page 108
3. Retest the disk group.

Using the -r option for non-destructive testing

To test disk devices containing data you want to preserve, you can use the -r option with the -m, -f, or -g options, which are described in the following sections. For example, to use the -m option and the -r option, you can run the utility by entering:

```
# /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.

Using the -m option

Review the procedure to test the shared disks.

"Testing the disks using the vxfentsthdw script" on page 107

The -m option which is the default option.

Using the -f option

Use the -f option to test disks that are listed in a text file. For example, you can create a file to test two disks shared by systems north and south that might resemble:

```
north /dev/rhdisk75 south /dev/rhdisk77
north /dev/rhdisk76 south /dev/rhdisk78
```

Where the first disk is listed in the first line and is seen by north as /dev/rhdisk75 and by south as /dev/rhdisk77. The other disk, in the second line, is seen as /dev/rhdisk76 from north and /dev/rhdisk78 from south. Typically, the list of disks could be extensive.

Suppose you created the file named disks_blue. To test the disks, you would enter:

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

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

You can redirect the test results to a text file. Precede the command with “yes” to acknowledge that the testing destroys any data on the disks to be tested.

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.

For example:

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

Using the -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 of vxdg command when importing the disk group.

The utility reports the test results one disk at a time. You can redirect the test results to a text file for review.

```
# /opt/VRTSvcs/vxfen/bin/vxfentsthdw -g red_disks_dg >
redtest.txt
```

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

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.

How I/O fencing works in different event scenarios

Table D-2 describes how I/O fencing works to prevent data corruption in different failure event scenarios. For each event, corrective operator actions are indicated.

Table D-2 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 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 D-2 I/O fencing scenarios

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 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 D-2 I/O fencing scenarios

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. "System panic prevents potential data corruption" on page 413.

Table D-2 I/O fencing scenarios

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. "System panic prevents potential data corruption" on page 413

Table D-2 I/O fencing scenarios

Event	Node A: What happens?	Node B: What happens?	Operator action
The disk array containing two of the three coordinator disks is powered off.	Node A continues to operate as long as no nodes leave the cluster.	Node B continues to operate as long as no nodes leave the cluster.	
Node B leaves the cluster and the disk array is still powered off.	Node A races for a majority of coordinator disks. Node A fails because only one of three coordinator disks is available. Node A removes itself from 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 the vxfenadm utility

Administrators can use the `vxfenadm` command to troubleshoot and test fencing configurations. The command's options for use by administrators are:

-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

Registration key formatting

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.

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/rhdisk74 by entering:

```
# vxfenadm -g /dev/rhdisk74
Reading SCSI Registration Keys...
Device Name: /dev/rhdisk74
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 plus 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.”

Appendix E

Configuring the Symantec License Inventory Agent

This appendix includes the following topics:

The Symantec License Inventory Manager installation disc is available separately. For information on how to order the full product, see "How to order the Symantec License Inventory Manager license and media kit" on page 487. The installation media provides online documentation with details on all topics discussed in this appendix.

Read the following Technical Support TechNote for the latest information on updates, patches, and software issues regarding this product:

http://support.veritas.com/docs/282183

You can also download the *Symantec License Inventory Agent 4.1 Release Notes*, from this website.

About the Symantec License Inventory Manager

The Symantec License Inventory Manager (license inventory manager) is an enterprise asset management tracking tool that inventories Symantec Information Availability products in your network and consolidates critical information on the deployment of these products to facilitate license management and compliance tracking. Using the information provided by the license inventory manager, you can:

- Determine all the Symantec software products and licenses being used in your enterprise
- Achieve easier license self-compliance management
- Know your Enterprise License Agreement deployment status
- Reduce administrative overhead for managing license compliance
- Renew support and maintenance based on the licenses you have deployed
- Gain more control over your Symantec software usage
- Manage department chargebacks based on actual software usage
- Use more flexible licensing and pricing models
- Exploit detailed deployment data to perform return on investment analyses for purchased software

The license inventory manager is a three-tiered system that consists of a server tier, access point tier, and an agent tier. The server tier is the Symantec License Inventory Server, which consolidates and stores information that it gathers from the agents and access points.

The optional access point tier includes Symantec License Inventory Access Points and serves as a consolidation layer between the agents and server.

The agent tier includes Symantec License Inventory Agents, which are deployed on individual hosts in a network. Each agent gathers product information on the supported Symantec products that are installed on the agent's host, then sends the information to an access point or the server.

When the Symantec License Inventory Agent is installed

The Symantec product installer installs or upgrades the agent on the host with the Symantec product. The agent is installed in the following directory:

`/opt/SYMClma`

The agent is installed with a default configuration that minimizes its impact on a running system. The minimum configuration prevents remote communication with the agent to keep its data and interfaces secure.

When the server and access points are installed

The server and access points are not installed automatically. If you want to use the Symantec License Inventory Manager, you must manually install the server and, optionally, the access points. After you install the server and access points, the agents can gather information and you can create inventory reports.

You can install the server and access points from the Symantec License Inventory Manager installation disc.

What you can do with the agent after it is installed

If you are already participating in a Symantec sales program that requires the use of the agent, or if you want to order and deploy the Symantec License Inventory Manager, you can use the agent to track Symantec products on the systems on which it was installed. To use the agent, however, you must manually configure it to enable remote communication between the agent and its server or access point.

Complete instructions for reconfiguring the agent are provided in the *Symantec License Inventory Manager 4.1 Release Notes*. You can download this document from the following website:

http://support.veritas.com/docs/282183

How to remove the agent

If you do not want to use the Symantec License Inventory Manager, you can remove the agent using the operating system package removal commands to remove the agent packages, which include SYMClma and VRTSsmf.

The server and access point also use the VRTSsmf package. If the server or access point is installed on this host with the agent, you can remove the SYMClma package, but not the VRTSsmf package. If neither the server nor the access point is installed on this host, you can remove both the SYMClma and VRTSsmf packages.

If you remove both packages, remove the SYMClma package first.

Table E-1 lists the commands required to remove these packages on the supported platforms.

Table E-1 Package removal commands required to remove the agent

Platform	Package removal command
AIX	`installp -u VRTSlma` `installp -u VRTSsmf`
HP-UX	`swremove SYMClma` `swremove VRTSsmf`
Linux	`rpm evv SYMClma` `rpm evv VRTSsmf`
Solaris	`pkgrm VRTSlma` `pkgrm VRTSsmf`

Later, you can reinstall the agent with the Symantec License Inventory Manager installation disc. This disc is available in the Symantec License Inventory Manager kit.

How to order the Symantec License Inventory Manager license and media kit

To order a Symantec License Inventory Manager license and media kit, contact your Symantec sales representative.

The installation media provides online documentation for the Symantec License Inventory Manager. You can contact your sales representative to order printed copies of the documentation. The documents you can order include:

- *Symantec License Inventory Manager Installation and Configuration Guide*
- *Symantec License Inventory Manager Administrator's Guide*
- *Symantec License Inventory Manager User's Guide*

Appendix F

Tunable kernel driver parameters

The tunable parameters described in this appendix are not intended to be used for performance enhancement. Several of the parameters pre-allocate memory for critical data structures, and a change in their values could increase memory use or degrade performance.

Note: Do not change the tunable kernel parameters described in this document without assistance from Symantec support personnel.

About LMX Tunable Parameters

The following table describes the LMX driver tunable parameters:

LMX Parameter	Description	Default Value	Maximum Value
lmx_minor_max	Specifies the maximum number of contexts/minors system-wide. Each Oracle process typically has two LMX contexts. "Contexts" and "minors" are used interchangeably in the documentation.	8192	65535
lmx_port_max	Specifies the number of communication endpoints for transferring messages from the sender to the receiver in a uni-directional manner.	8192	65535

LMX Parameter	Description	Default Value	Maximum Value
lmx_buffer_max	Specifies the number of addressable regions in memory to which LMX data can be copied.	8192	65535

Example: Configuring LMX Parameters

If you see the message “no minors available” on one node, you can increase the value for the `contexts` kernel parameter. For example, you can configure the system for `contexts=16384`, which allows a total of 8192 Oracle processes. The error message may contain the term “minors,” but you must use the term “contexts” when changing the value of the parameter.

To change the LMX tunable parameter `contexts`, use the `chdev` command, unload the driver, and reload the driver. Change the parameter on each cluster system. For example, to change the `contexts` parameter to 16384, do the following:

1. List the current value of the tunable LMX parameters:

```
# lsattr -El vcsmm
contexts   8192  N/A  True
ports      4096  N/A  True
buffers    4096  N/A  True
```

   The current value for `contexts` is 8192 (the default).

2. Enter:

```
# chdev -I vcsmm -P -a contexts=16384
```

3. Unload and reload the driver after making the change:

```
# /etc/methods/vcsmmext -stop
# /etc/methods/vcsmmext -start
```

4. Perform the commands on each cluster system to change the parameter.

About VXFEN Tunable Parameters

On each node, edit the file /kernel/drv/vxfen.conf to change the value of the vxfen driver tunable global parameter, vxfen_max_delay and vxfen_min_delay. You must restart the system to put change into effect.

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
vxfen_max_delay	Specifies the maximum number of seconds that the smaller sub-cluster waits before racing with larger clusters for control of the coordinator disks. This value must be greater than the vxfen_max_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 clusters for control of the coordinator disks. 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 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.

Example: Configuring the VXFEN Parameters

To change the VXFEN tunable parameter, use the chdev command, unload the driver, and reload the driver. Change the parameter on each cluster system. The example procedure changes the value of the vxfen_min_delay parameter.

To change the VXFEN tunable parameter

1. Check the status of the driver and start it if necessary:

   ```
   # /etc/methods/vxfenext -status
   ```

   To start the driver:

   ```
   # /etc/init.d/vxfen.rc start
   ```

2. List the current value of the tunable parameter:

   ```
   # lsattr -El vxfen
   vxfen_min_delay=0
   ```

   The current value is 1 (the default).

3. Enter:

   ```
   # chdev -l vxfen -P -a vxfen_min_delay=30
   ```

4. Ensure that SF Oracle RAC is shut down. Unload and reload the driver after changing the value of the tunable:

   ```
   # /etc/methods/vxfenext -stop
   # /etc/methods/vxfenext -start
   ```

5. Perform the commands on each cluster system to change the parameter.

Appendix G

Error messages

The error messages listed in this appendix are grouped by the software module in which the error occurs.

LMX Error Messages, Critical

The following table lists LMX kernel module error messages. These messages 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.

Table G-1 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*

Message ID	LMX Message
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 Error Messages, Non-Critical

The following table contains LMX error messages that may be displayed during runtime. Refer to "Running Scripts for Engineering Support Analysis" on page 125 for information on how to gather information about your systems and configuration that Symantec support personnel can use to assist you.

Table G-2 If you encounter errors while running your Oracle application due to the display of these messages, you may use the lmxconfig command to turn off their display. Noncritical Error Messages

Message ID	LMX Message
06001	lmxreqlink duplicate kreq= 0x*address*, req= 0x*address*
06002	lmxreqlink duplicate ureq= 0x*address* kr1= 0x*address*, kr2= 0x*address* req type = number
06003	lmxrequnlink not found kreq= 0x*address* from= number
06004	lmxrequnlink_l not found kreq= 0x*address* from= number
06005	kreq was not found
06101	lmxpollreq not in doneq CONN kreq= 0x*address*
06201	lmxnewcontext lltinit fail err= number

Message ID	LMX Message
06202	lmxnewcontext lltregister 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*
06801	lmxconfig insufficient privilege, uid= *number*
06901	lmxlltnodestat: LLT getnodeinfo failed err= *number*

VxVM Errors Related to 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 has not been configured. Follow the instructions in the chapter on installing and configuring SF Oracle RAC to set up coordinator disks and start I/O fencing. Then clear the faulted resources and online the service groups.
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

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 printed.
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 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 vxfen driver, if there is a fencing race going on in the cluster. The vxfen driver can be configured by retrying after sometime (after the cluster completes the fencing).

VXFEN Driver Informational Message

```
date and time VXFEN:00021:Starting to eject leaving nodes(s)
from data disks.

date and time VXFEN:00022:Completed ejection of leaving node(s)
from data disks.
```

These messages are for information only. They show how long it takes the data disks to be fenced for nodes that have left the cluster.

Informational Messages When Node is Ejected

```
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.

Glossary

Agent

A process that starts, stops, and monitors all configured resources of a type, and reports their status to VCS.

Active/Active Configuration

A failover configuration where each system runs a service group. If either fails, the other one takes over and runs both service groups. Also known as a symmetric configuration.

Active/Passive Configuration

A failover configuration consisting of one service group on a primary system, and one dedicated backup system. Also known as an asymmetric configuration.

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.

Cluster Manager (Java Console)

A Java-based graphical user interface to manage VCS clusters. It provides complete administration capabilities for a cluster, and can run on any system inside or outside the cluster, on any operating system that supports Java.

Cluster Manager (Web Console)

A Web-based graphical user interface for monitoring and administering the cluster.

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 Heartbeats (GABDISK)

A way to improve cluster resiliency, GABDISK enables a heartbeat to be placed on a physical disk shared by all systems in the cluster.

Failover

A failover occurs when a service group faults and is migrated to another system.

GAB

Group Atomic Broadcast (GAB) is a communication mechanism of the VCS engine that manages cluster membership, monitors heartbeat communication, and distributes information throughout the cluster.

Global Service Group

A VCS service group which spans across two or more clusters. The `ClusterList` attribute for this group contains the list of clusters over which the group spans.

hashadow Process

A process that monitors and, when required, restarts HAD.

High Availability Daemon (HAD)

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.

Jeopardy

A node is in *jeopardy w*hen 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.

LLT

Low Latency Transport (LLT) is a communication mechanism of the VCS engine that provides kernel-to-kernel communications and monitors network communications.

main.cf

The file in which the cluster configuration is stored.

Monitor Program

The Monitor Program informs the application agent whether the application process is online or offline, and properly returning service requests.

Network Partition

If all network connections between any two groups of systems fail simultaneously, a *network partition* occurs. When this happens, systems on both sides of the partition can restart applications from the other side resulting in duplicate services, or "split-brain." A split brain occurs when two independent systems configured in a cluster assume they have exclusive access to a given resource (usually a file system or volume). The most serious problem caused by a network partition is that it affects the data on shared disks. See "Jeopardy" and "Seeding".

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.

N-to-1

An N-to-1 configuration is based on the concept that multiple, simultaneous server failures are unlikely; therefore, a single backup server can protect multiple active servers. When a server fails, its applications move to the backup server. For example, in a 4-to-1 configuration, one server can protect four servers, which reduces redundancy cost at the server level from 100 percent to 25 percent.

N-to-N

N-to-N refers to multiple service groups running on multiple servers, with each service group capable of being failed over to different servers in the cluster. For example, consider a four-node cluster with each node supporting three critical database instances. If any

node fails, each instance is started on a different node, ensuring no single node becomes overloaded.

N-to-M

N-to-M (or Any-to-Any) refers to multiple service groups running on multiple servers, with each service group capable of being failed over to different servers in the same cluster, and also to different servers in a linked cluster. For example, consider a four-node cluster with each node supporting three critical database instances and a linked two-node back-up cluster. If all nodes in the four-node cluster fail, each instance is started on a node in the linked back-up cluster.

Replication

Replication is the synchronization of data between systems where shared storage is not feasible. The systems that are copied may be in local backup clusters or remote failover sites. The major advantage of replication, when compared to traditional backup methods, is that current data is continuously available.

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 or network interface card, a software component such as Oracle8*i* 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.

Seeding

Seeding is used to protect a cluster from a preexisting network partition. By default, when a system comes up, it is not seeded. Systems can be seeded automatically or manually. Only systems that have been seeded can run VCS. Systems are seeded automatically only when: an unseeded system communicates with a seeded system or all systems in the cluster are unseeded and able to communicate with each other. See “Network Partition”.

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.

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.

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

The types.cf file describes standard resource types to the VCS engine; specifically, the data required to control a specific resource.

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.

Index

Symbols

A

B

C

D

E

U

V