Storage Foundation for Oracle® RAC 7.4 Administrator's Guide - Linux

VERITAS™

Last updated: 2018-05-30

Legal Notice

Veritas Technologies LLC
500 E Middlefield Road
Mountain View, CA 94043

http://www.veritas.com

Technical Support

Technical Support maintains support centers globally. All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policies. For information about our support offerings and how to contact Technical Support, visit our website:

https://www.veritas.com/support

You can manage your Veritas account information at the following URL:

https://my.veritas.com

If you have questions regarding an existing support agreement, please email the support agreement administration team for your region as follows:

Worldwide (except Japan)	CustomerCare@veritas.com
Japan	CustomerCare_Japan@veritas.com

Documentation

Make sure that you have the current version of the documentation. Each document displays the date of the last update on page 2. The latest documentation is available on the Veritas website:

https://sort.veritas.com/documents

Documentation feedback

Your feedback is important to us. Suggest improvements or report errors or omissions to the documentation. Include the document title, document version, chapter title, and section title of the text on which you are reporting. Send feedback to:

doc.feedback@veritas.com

You can also see documentation information or ask a question on the Veritas community site:

http://www.veritas.com/community/

Veritas Services and Operations Readiness Tools (SORT)

Veritas Services and Operations Readiness Tools (SORT) is a website that provides information and tools to automate and simplify certain time-consuming administrative tasks. Depending on the product, SORT helps you prepare for installations and upgrades, identify risks in your datacenters, and improve operational efficiency. To see what services and tools SORT provides for your product, see the data sheet:

https://sort.veritas.com/data/support/SORT_Data_Sheet.pdf

Contents

Section 1

SF Oracle RAC concepts and administration

Chapter 1

Overview of Storage Foundation for Oracle RAC

This chapter includes the following topics:

About Storage Foundation for Oracle RAC

Storage Foundation™ for Oracle® RAC (SF Oracle RAC) leverages proprietary storage management and high availability technologies to enable robust, manageable, and scalable deployment of Oracle RAC on UNIX platforms. The solution uses Veritas Cluster File System technology that provides the dual advantage of easy file system management as well as the use of familiar operating system tools and utilities in managing databases.

The solution comprises the Cluster Server (VCS), Veritas Cluster Volume Manager (CVM), Veritas Oracle Real Application Cluster Support (VRTSdbac), Veritas Oracle Disk Manager (VRTSodm), Veritas Cluster File System (CFS), and Storage

Foundation, which includes the base Veritas Volume Manager (VxVM) and Veritas File System (VxFS).

Benefits of SF Oracle RAC

SF Oracle RAC provides the following benefits:

- Support for file system-based management. SF Oracle RAC provides a generic clustered file system technology for storing and managing Oracle data files as well as other application data.
- Support for different storage configurations:
  Shared storage
  Flexible Storage Sharing (FSS): Sharing of Direct Attached Storage (DAS) and internal disks over network
- Faster performance and reduced costs per I/O per second (IOPS) using SmartIO. SmartIO supports read caching for the VxFS file systems that are mounted on VxVM volumes, in several caching modes and configurations. SmartIO also supports block-level read caching for applications running on VxVM volumes.
- For Oracle 11gR2 and 12cR1, use Cluster File System and Cluster Volume Manager for placement of Oracle Cluster Registry (OCR) and voting disks. These technologies provide robust shared block interfaces for placement of OCR and voting disks. In the absence of SF Oracle RAC, separate LUNs need to be configured for OCR and voting disks.
- For Oracle 12cR2, you must store the OCR and voting files on Oracle ASM disk groups. You must create the Oracle ASM disks group on the raw CVM volumes during Oracle Grid installation.
- Support for a standardized approach toward application and database management. Administrators can apply their expertise of technologies toward administering SF Oracle RAC.
- Increased availability and performance using Dynamic Multi-Pathing (DMP). DMP provides wide storage array support for protection from failures and performance bottlenecks in the Host Bus Adapters (HBA), Storage Area Network (SAN) switches, and storage arrays.
- Easy administration and monitoring of multiple SF Oracle RAC clusters using Veritas InfoScale Operations Manager.
- VCS OEM plug-in provides a way to monitor SF Oracle RAC resources from the OEM console.
  For more information, see the *Veritas InfoScale Storage and Availability Management for Oracle Databases* guide.
- Improved file system access times using Oracle Disk Manager (ODM).

- Ability to configure Oracle Automatic Storage Management (ASM) disk groups over CVM volumes to take advantage of Dynamic Multi-Pathing (DMP).
- Enhanced scalability and availability with access to multiple Oracle RAC instances per database in a cluster.
- Support for backup and recovery solutions using volume-level and file system-level snapshot technologies, Storage Checkpoints, and Database Storage Checkpoints.
  For more information, see the *Veritas InfoScale Storage and Availability Management for Oracle Databases* guide.
- Support for space optimization using periodic deduplication in a file system to eliminate duplicate data without any continuous cost.
  For more information, see the Storage Foundation Administrator's documentation.
- Ability to fail over applications with minimum downtime using Cluster Server (VCS) and Veritas Cluster File System (CFS).
- Prevention of data corruption in split-brain scenarios with robust SCSI-3 Persistent Group Reservation (PGR) based I/O fencing or Coordination Point Server-based I/O fencing. The preferred fencing feature also enables you to specify how the fencing driver determines the surviving subcluster.
- Support for sharing application data, in addition to Oracle database files, across nodes.
- Support for policy-managed databases in Oracle RAC 11g Release 2 and later versions.
- Support for container and pluggable databases in Oracle RAC 12c and later versions.
- Fast disaster recovery with minimal downtime and interruption to users. Users can transition from a local high availability site to a wide-area disaster recovery environment with primary and secondary sites. If a site fails, clients that are attached to the failed site can reconnect to a surviving site and resume access to the shared database.
- Verification of disaster recovery configuration using fire drill technology without affecting production systems.
- Support for a wide range of hardware replication technologies as well as block-level replication using VVR.
- Support for campus clusters with the following capabilities:
  - Consistent detach with Site Awareness
  - Site aware reads with VxVM mirroring

- Monitoring of Oracle resources
- Protection against split-brain scenarios

How SF Oracle RAC works (high-level perspective)

Oracle Real Application Clusters (RAC) is a parallel database environment that takes advantage of the processing power of multiple computers. Oracle stores data logically in the form of tablespaces and physically in the form of data files. 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 that interact with each other and with the operating system to manage memory structure and do general housekeeping.

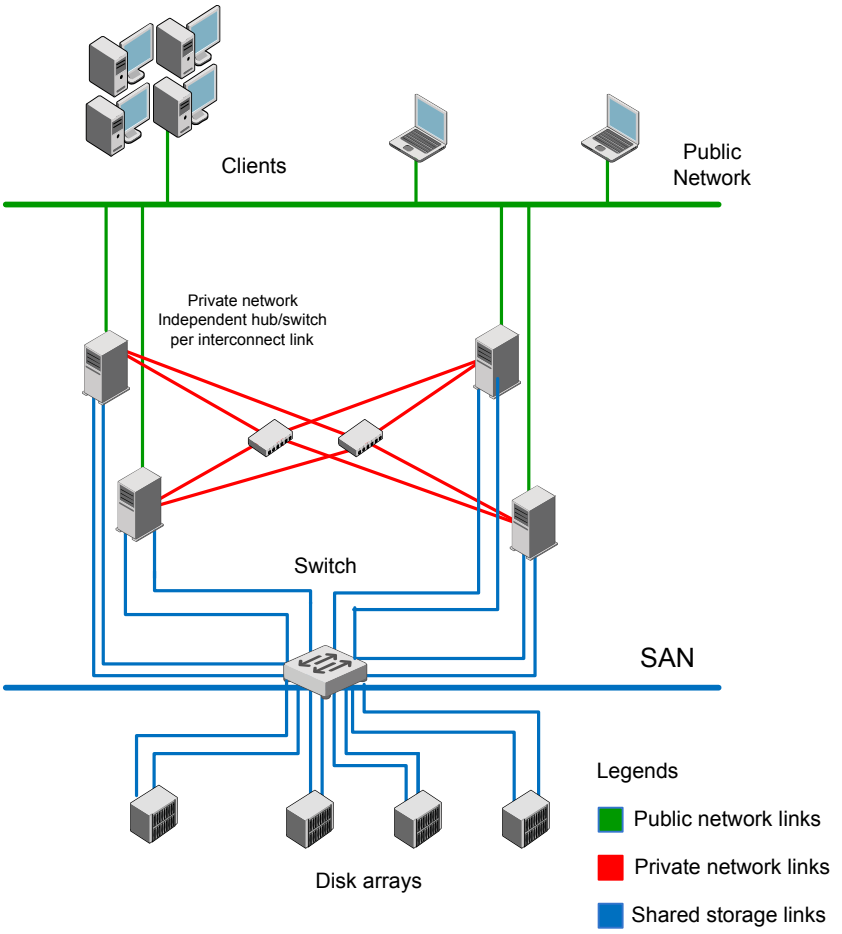
SF Oracle RAC provides the necessary infrastructure for running Oracle RAC and coordinates access to the shared data for each node to provide consistency and integrity. Each node adds its processing power to the cluster as a whole and can increase overall throughput or performance.

At a conceptual level, SF Oracle RAC is a cluster that manages applications (Oracle instances), networking, and storage components using resources contained in service groups. SF Oracle RAC clusters have the following properties:

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

Figure 1-1 displays the basic layout and individual components required for a SF Oracle RAC installation.

Figure 1-1 SF Oracle RAC basic layout and components



The basic layout has the following characteristics:

- Multiple client applications that access nodes in the cluster over a public network.
- Nodes that are connected by at least two private network links (also called cluster interconnects) that are connected to two different switches using 100BaseT or gigabit Ethernet controllers on each system.
  If the private links are on a single switch, isolate them using VLAN.
- Nodes that are connected to iSCSI or Fibre Channel shared storage devices over SAN.
  All shared storage must support SCSI-3 PR.

- Nodes must be connected with private network links using similar network devices and matching port numbers.

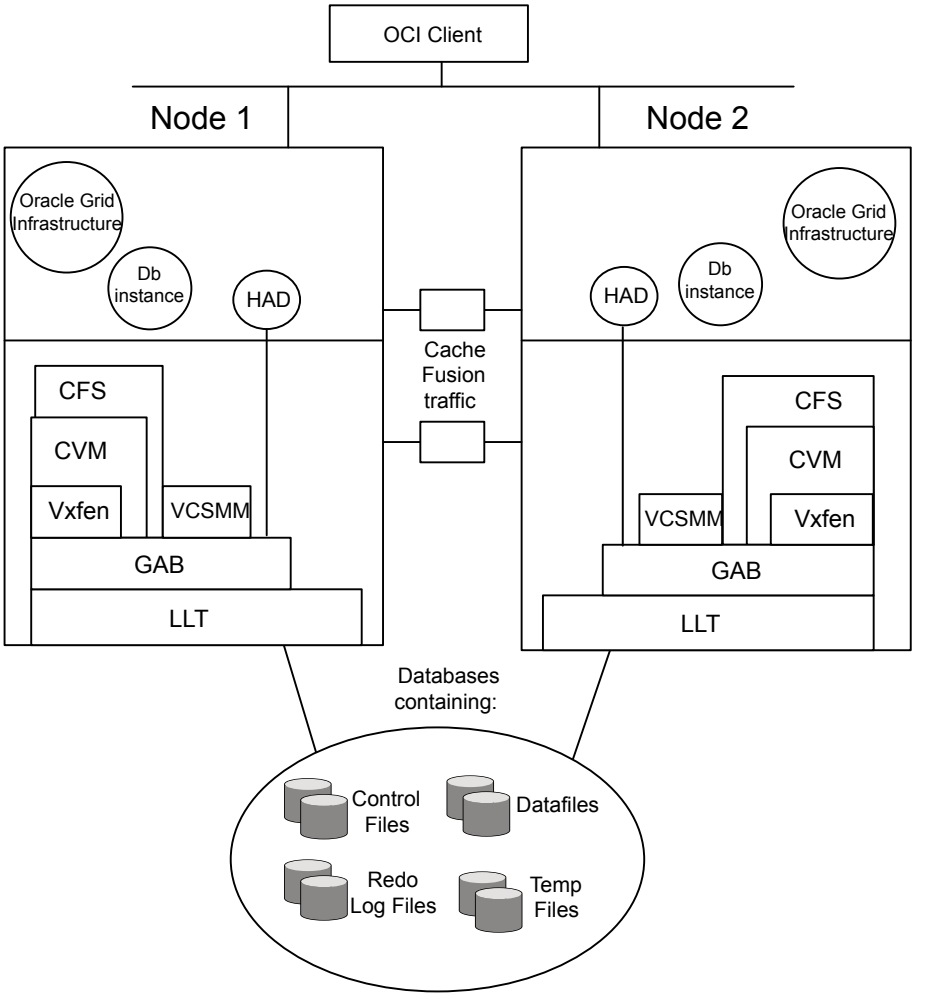
  For example, if you use eth1 on one end of a link, the other end must also use eth1.
- The Oracle Cluster Registry, vote disks, and data files configured on the shared storage that is available to each node. The shared storage can be a cluster file system or ASM disk groups created using raw VxVM volumes.
- Three or an odd number of standard disks or LUNs (recommended number is three) used as coordinator disks or as coordination point (CP) servers for I/O fencing.
- VCS manages the resources that are required by Oracle RAC. The resources must run in parallel on each node.

SF Oracle RAC includes the following technologies that are engineered to improve performance, availability, and manageability of Oracle RAC environments:

- Cluster File System (CFS) and Cluster Volume Manager (CVM) technologies to manage multi-instance database access to shared storage.
- An Oracle Disk Manager (ODM) library to maximize Oracle disk I/O performance.
- Interfaces to Oracle Grid Infrastructure and RAC for managing cluster membership.

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

Figure 1-2 SF Oracle RAC architecture



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

Component products and processes of SF Oracle RAC

Table 1-1 lists the component products of SF Oracle RAC.

Table 1-1 SF Oracle RAC component products

Component product	Description
Cluster Volume Manager (CVM)	Enables simultaneous access to shared volumes based on technology from Veritas Volume Manager (VxVM). See "Cluster Volume Manager (CVM)" on page 26.
Cluster File System (CFS)	Enables simultaneous access to shared file systems based on technology from Veritas File System (VxFS). See "Cluster File System (CFS)" on page 33.
Cluster Server (VCS)	Manages Oracle RAC databases and infrastructure components. See "Cluster Server (VCS)" on page 36.
Veritas I/O fencing	Protects the data on shared disks when nodes in a cluster detect a change in the cluster membership that indicates a split-brain condition. See "About I/O fencing" on page 39.
Oracle RAC	Component of the Oracle database product that allows a database to be installed on multiple servers. See "Oracle RAC components" on page 40.
Oracle Disk Manager (Database Accelerator)	Provides the interface with the Oracle Disk Manager (ODM) API. See "Oracle Disk Manager" on page 43.
RAC Extensions	Manages cluster membership between cluster nodes. See "RAC extensions" on page 44.

Communication infrastructure

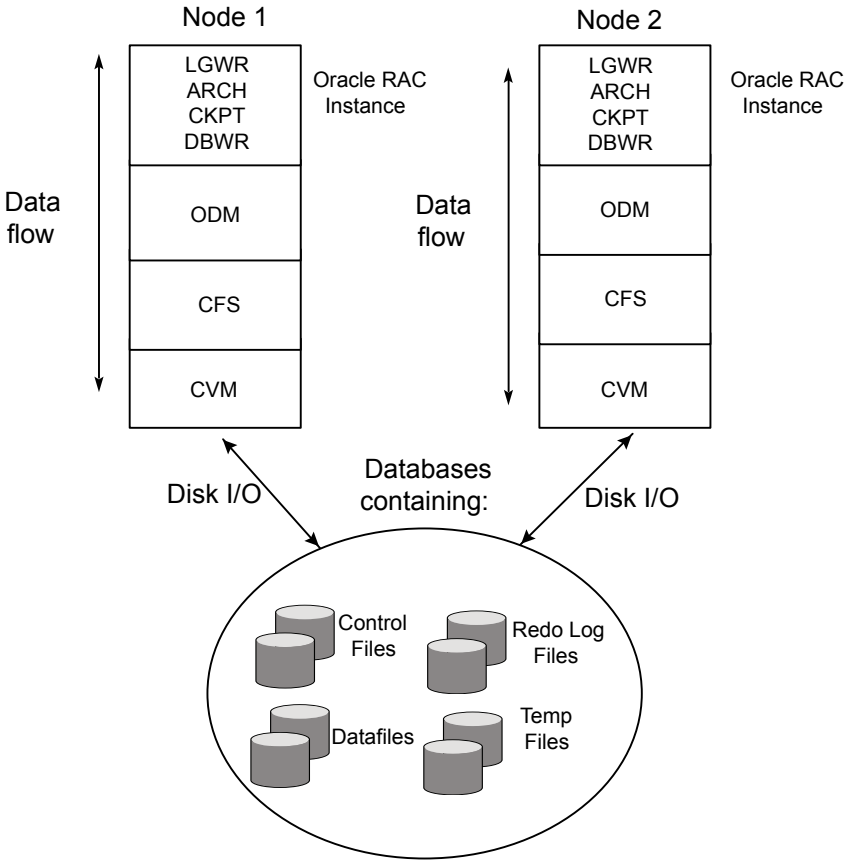
To understand the communication infrastructure, review the data flow and communication requirements.

Data flow

The CVM, CFS, ODM, and Oracle RAC elements reflect the overall data flow, or data stack, from an instance running on a server to the shared storage. The various Oracle processes composing an instance -- such as DBWR (Database Writer), LGWR (Log Writer process), CKPT (Storage Checkpoint process), and ARCH (Archive process (optional)) -- read and write data to the storage through the I/O stack. Oracle communicates through the ODM interface to CFS, which in turn accesses the storage through the CVM.

Figure 1-3 represents the overall data flow.

Figure 1-3 Data stack

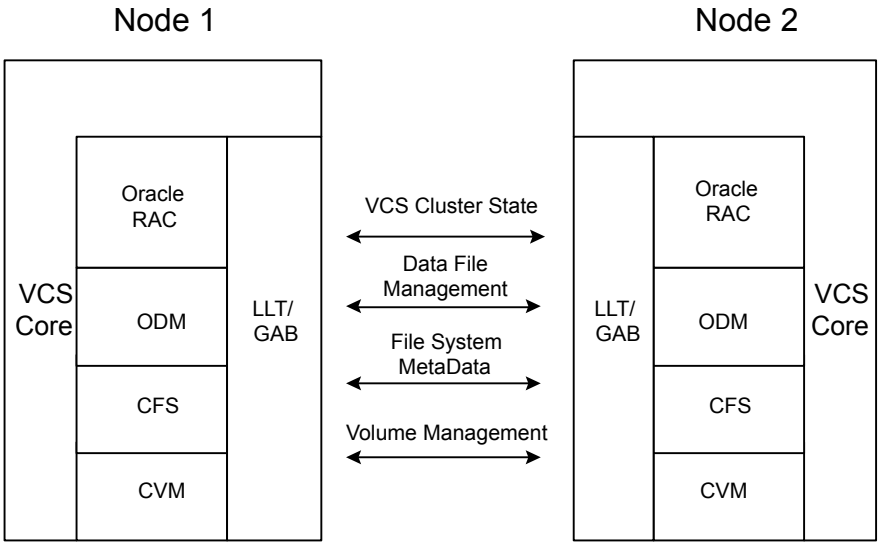


Communication requirements

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

Figure 1-4 represents the communication stack.

Figure 1-4 Communication stack



Cluster interconnect communication channel

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

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

About Low Latency Transport (LLT)

The Low Latency Transport protocol is used for all cluster communications as a high-performance, low-latency replacement for the IP stack.

LLT has the following two major functions:

- Traffic distribution
  LLT provides the communications backbone for GAB. LLT distributes (load balances) inter-system communication across all configured network links. This distribution ensures all cluster communications are evenly distributed across all network links for performance and fault resilience. If a link fails, traffic is redirected to the remaining links. A maximum of eight network links are supported.

- Heartbeat

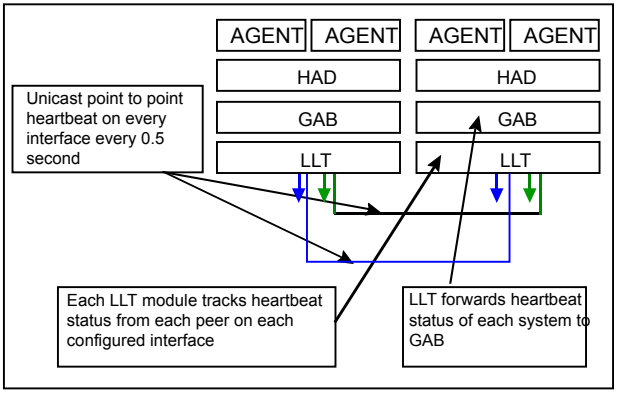
  LLT is responsible for sending and receiving heartbeat traffic over each configured network link. The heartbeat traffic is point to point unicast. LLT uses ethernet broadcast to learn the address of the nodes in the cluster. All other cluster communications, including all status and configuration traffic is point to point unicast. The heartbeat is used by the Group Membership Services to determine cluster membership.

The heartbeat signal is defined as follows:

- LLT on each system in the cluster sends heartbeat packets out on all configured LLT interfaces every half second.
- LLT on each system tracks the heartbeat status from each peer on each configured LLT interface.
- LLT on each system forwards the heartbeat status of each system in the cluster to the local Group Membership Services function of GAB.
- GAB receives the status of heartbeat from all cluster systems from LLT and makes membership determination based on this information.

Figure 1-5 shows heartbeat in the cluster.

Figure 1-5 Heartbeat in the cluster



LLT can be configured to designate specific cluster interconnect links as either high priority or low priority. High priority links are used for cluster communications to GAB as well as heartbeat signals. Low priority links, during normal operation, are used for heartbeat and link state maintenance only, and the frequency of heartbeats is reduced to 50% of normal to reduce network overhead.

If there is a failure of all configured high priority links, LLT will switch all cluster communications traffic to the first available low priority link. Communication traffic will revert back to the high priority links as soon as they become available.

While not required, best practice recommends to configure at least one low priority link, and to configure two high priority links on dedicated cluster interconnects to provide redundancy in the communications path. Low priority links are typically configured on the public or administrative network.

If you use different media speed for the private NICs, Veritas recommends that you configure the NICs with lesser speed as low-priority links to enhance LLT performance. With this setting, LLT does active-passive load balancing across the private links. At the time of configuration and failover, LLT automatically chooses the link with high-priority as the active link and uses the low-priority links only when a high-priority link fails.

LLT sends packets on all the configured links in weighted round-robin manner. LLT uses the linkburst parameter which represents the number of back-to-back packets that LLT sends on a link before the next link is chosen. In addition to the default weighted round-robin based load balancing, LLT also provides destination-based load balancing. LLT implements destination-based load balancing where the LLT link is chosen based on the destination node id and the port. With destination-based load balancing, LLT sends all the packets of a particular destination on a link. However, a potential problem with the destination-based load balancing approach is that LLT may not fully utilize the available links if the ports have dissimilar traffic. Veritas recommends destination-based load balancing when the setup has more than two cluster nodes and more active LLT ports. You must manually configure destination-based load balancing for your cluster to set up the port to LLT link mapping.

See "Configuring destination-based load balancing for LLT" on page 87.

LLT on startup sends broadcast packets with LLT node id and cluster id information onto the LAN to discover any node in the network that has same node id and cluster id pair. Each node in the network replies to this broadcast message with its cluster id, node id, and node name.

LLT on the original node does not start and gives appropriate error in the following cases:

- LLT on any other node in the same network is running with the same node id and cluster id pair that it owns.
- LLT on the original node receives response from a node that does not have a node name entry in the `/etc/llthosts` file.

How LLT supports RDMA capability for faster interconnects between applications

LLT and GAB support fast interconnect between applications using RDMA technology over InfiniBand and Ethernet media (RoCE). To leverage the RDMA capabilities of the hardware and also support the existing LLT functionalities, LLT

maintains two channels (RDMA and non-RDMA) for each of the configured RDMA links. Both RDMA and non-RDMA channels are capable of transferring data between the nodes and LLT provides separate APIs to their clients, such as, CFS, CVM, to use these channels. The RDMA channel provides faster data transfer by leveraging the RDMA capabilities of the hardware. The RDMA channel is mainly used for data-transfer when the client is capable to use this channel. The non-RDMA channel is created over the UDP layer and LLT uses this channel mainly for sending and receiving heartbeats. Based on the health of the non-RDMA channel, GAB decides cluster membership for the cluster. The connection management of the RDMA channel is separate from the non-RDMA channel, but the connect and disconnect operations for the RDMA channel are triggered based on the status of the non-RDMA channel

If the non-RDMA channel is up but due to some issues in RDMA layer the RDMA channel is down, in such cases the data-transfer happens over the non-RDMA channel with a lesser performance until the RDMA channel is fixed. The system logs displays the message when the RDMA channel is up or down.

LLT uses the Open Fabrics Enterprise Distribution (OFED) layer and the drivers installed by the operating system to communicate with the hardware. LLT over RDMA allows applications running on one node to directly access the memory of an application running on another node that are connected over an RDMA-enabled network. In contrast, on nodes connected over a non-RDMA network, applications cannot directly read or write to an application running on another node. LLT clients such as, CFS and CVM, have to create intermediate copies of data before completing the read or write operation on the application, which increases the latency period and affects performance in some cases.

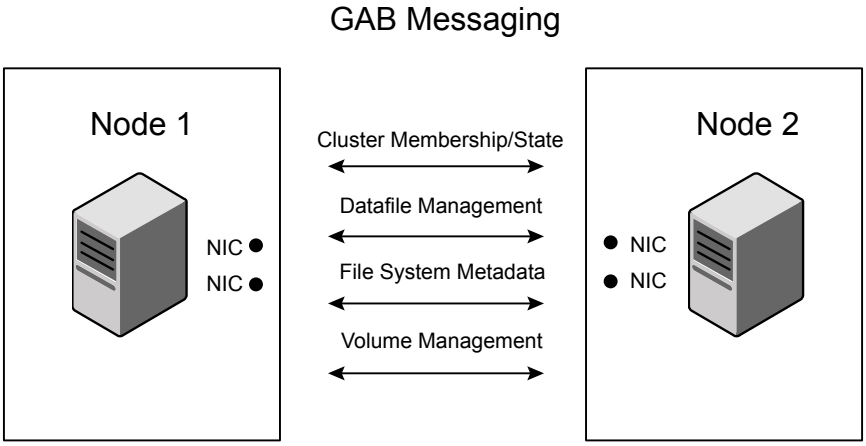
LLT over an RDMA network enables applications to read or write to applications on another node over the network without the need to create intermediate copies. This leads to low latency, higher throughput, and minimized CPU host usage thus improving application performance. Cluster volume manager and Cluster File Systems, which are clients of LLT and GAB, can use LLT over RDMA capability for specific use cases.

Group Membership Services/Atomic Broadcast

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

Figure 1-6 shows the cluster communication using GAB messaging.

Figure 1-6 Cluster communication



Review the following information on cluster membership and cluster communication:

- Cluster membership
  At a high level, all nodes configured by the installer can operate as a cluster; these nodes form a cluster membership. In SF Oracle RAC, a cluster membership specifically refers to all systems configured with the same cluster ID communicating by way of a redundant cluster interconnect.
  All nodes in a distributed system, such as SF Oracle RAC, must remain constantly alert to the nodes currently participating in the cluster. Nodes can leave or join the cluster at any time because of shutting down, starting up, rebooting, powering off, or faulting processes. SF Oracle RAC uses its cluster membership capability to dynamically track the overall cluster topology.
  SF Oracle RAC uses LLT heartbeats to determine cluster membership:
  - When systems no longer receive heartbeat messages from a peer for a predetermined interval, a protocol excludes the peer from the current membership.
  - GAB informs processes on the remaining nodes that the cluster membership has changed; this action initiates recovery actions specific to each module. For example, CVM must initiate volume recovery and CFS must perform a fast parallel file system check.
  - When systems start receiving heartbeats from a peer outside of the current membership, a protocol enables the peer to join the membership.
- Cluster communications
  GAB provides reliable cluster communication between SF Oracle RAC modules. GAB provides guaranteed delivery of point-to-point messages and broadcast

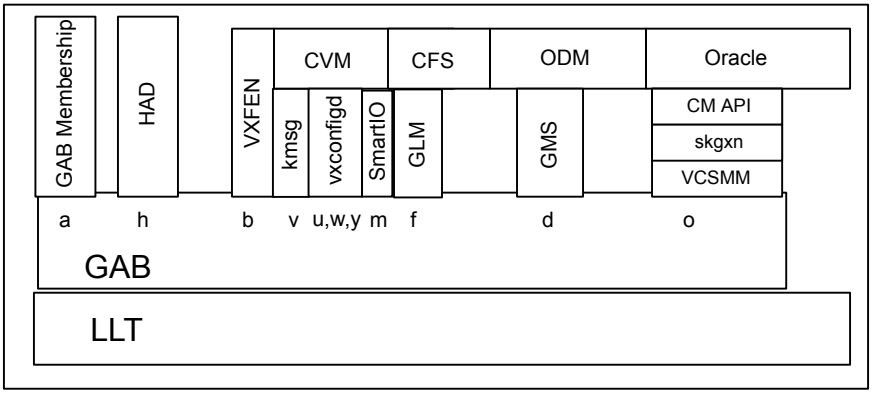
messages to all nodes. Point-to-point messaging involves sending and acknowledging the message. Atomic-broadcast messaging ensures all systems within the cluster receive all messages. If a failure occurs while transmitting a broadcast message, GAB ensures all systems have the same information after recovery.

Low-level communication: port relationship between GAB and processes

All components in SF Oracle RAC use GAB for communication. Each process that wants 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 unregisters.

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

Figure 1-7 Low-level communication



Cluster Volume Manager (CVM)

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

CVM supports performance-enhancing capabilities, such as striping, mirroring, and mirror break-off (snapshot) for off-host backup. You can use standard VxVM

commands from one node in the cluster to manage all storage. All other nodes immediately recognize any changes in disk group and volume configuration with no user interaction.

For detailed information, see the *Storage Foundation Cluster File System High Availability Administrator's Guide*.

CVM architecture

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

Just as with VxVM, the Volume Manager configuration daemon, `vxconfigd`, maintains the configuration of logical volumes. This daemon handles changes to the volumes by updating the operating system at the kernel level. For example, if a mirror of a volume fails, the mirror detaches from the volume and `vxconfigd` determines the proper course of action, updates the new volume layout, and informs the kernel of a new volume layout. CVM extends this behavior across multiple nodes and propagates volume changes to the master `vxconfigd`.

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

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

See Figure 1-7 on page 26.

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.

By default, 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.

Set the `storage_connectivity` tunable to asymmetric to enable a cluster node to join even if the node does not have access to all of the shared storage. Similarly, a node can import a shared disk group even if there is a local failure to the storage.

For detailed information, see the *Storage Foundation Cluster File System High Availability Administrator's Guide*.

CVM communication

CVM communication involves various GAB ports for different types of communication. For an illustration of these ports:

See Figure 1-7 on page 26.

CVM communication involves the following GAB ports:

- Port w
  Most CVM communication uses port w for vxconfigd communications. During any change in volume configuration, such as volume creation, plex attachment or detachment, and volume resizing, vxconfigd on the master node uses port w to share this information with slave nodes.
  When all slaves use port w to acknowledge the new configuration as the next active configuration, the master updates this record to the disk headers in the VxVM private region for the disk group as the next configuration.
- Port m
  CVM uses port m for SmartIO VxVM cache coherency using Group Lock Manager (GLM).
- 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.
- Port u
  CVM uses the group atomic broadcast (GAB) port u to ship the commands from the slave node to the master node.
- Port y
  CVM uses port y for kernel-to-kernel communication required while shipping I/Os from nodes that might have lost local access to storage to other nodes in the cluster.

CVM recovery

When a node leaves a cluster, the new membership is delivered by GAB, to CVM on existing cluster nodes. The fencing driver (VXFEN) ensures that split-brain scenarios are taken care of before CVM is notified. CVM then initiates recovery of

mirrors of shared volumes that might have been in an inconsistent state following the exit of the node.

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

Configuration differences with VxVM

CVM configuration differs from VxVM configuration in the following areas:

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

About Flexible Storage Sharing

Flexible Storage Sharing (FSS) enables network sharing of local storage, cluster wide. The local storage can be in the form of Direct Attached Storage (DAS) or internal disk drives. Network shared storage is enabled by using a network interconnect between the nodes of a cluster.

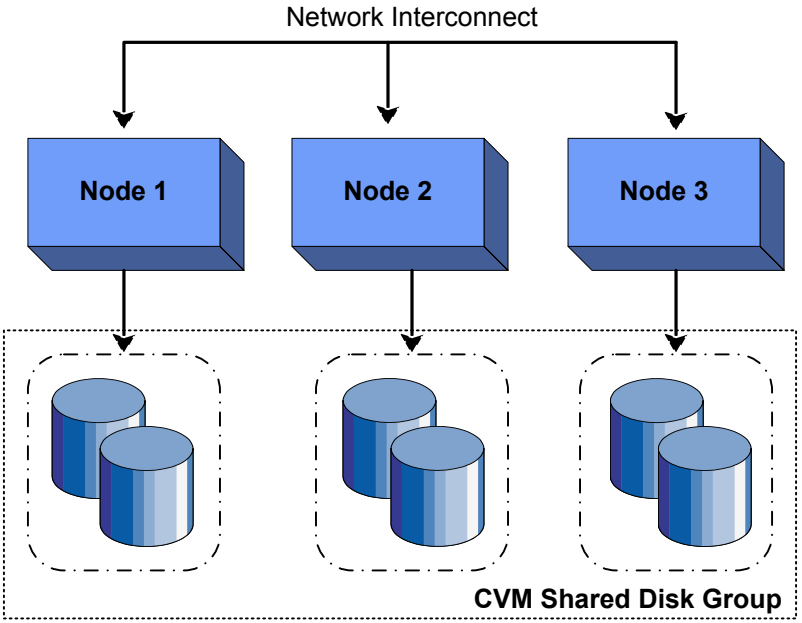
FSS allows network shared storage to co-exist with physically shared storage, and logical volumes can be created using both types of storage creating a common storage namespace. Logical volumes using network shared storage provide data redundancy, high availability, and disaster recovery capabilities, without requiring physically shared storage, transparently to file systems and applications.

FSS can be used with SmartIO technology for remote caching to service nodes that may not have local SSDs.

FSS is supported on clusters containing up to 64 nodes with CVM protocol versions 140 and above. For more details, refer to the *Veritas InfoScale Release Notes*.

Figure 1-8 shows a Flexible Storage Sharing environment.

Figure 1-8 Flexible Storage Sharing Environment



Flexible Storage Sharing use cases

The following list includes several use cases for which you would want to use the FSS feature:

Use of local storage in current use cases	The FSS feature supports all current use cases of the Storage Foundation for Oracle RAC (SF Oracle RAC) stack without requiring SAN-based storage.

Off-host processing	Data Migration: ■ From shared (SAN) storage to network shared storage ■ From network shared storage to SAN storage ■ From storage connected to one node (DAS)/cluster to the storage connected to a different node (DAS)/cluster, that do not share the storage Back-up/Snapshots: An additional node can take a back-up by joining the cluster and reading from volumes/snapshots that are hosted on the DAS/shared storage, which is connected to one or more nodes of the cluster, but not the host taking the back-up.
DAS SSD benefits leveraged with existing SF Oracle RAC features	■ Mirroring across DAS SSDs connected to individual nodes of the cluster. DAS SSDs provides better performance than SAN storage (including SSDs). FSS provides a way to share these SSDs across cluster. ■ Keeping one mirror on the SSD and another on the SAN storage provides faster read access due to the SSDs, and also provide high availability of data due to the SAN storage. ■ There are several best practices for using SSDs with Storage Foundation. All the use-cases are possible with SAN attached SSDs in clustered environment. With FSS, DAS SSDs can also be used for similar purposes.
FSS with SmartIO for file system caching	If the nodes in the cluster have internal SSDs as well as HDDs, the HDDs can be shared over the network using FSS. You can use SmartIO to set up a read/write-back cache using the SSDs. The read cache can service volumes created using the network-shared HDDs.

FSS with SmartIO for remote caching	FSS works with SmartIO to provide caching services for nodes that do not have local SSD devices. In this scenario, Flexible Storage Sharing (FSS) exports SSDs from nodes that have a local SSD. FSS then creates a pool of the exported SSDs in the cluster. From this shared pool, a cache area is created for each node in the cluster. Each cache area is accessible only to that particular node for which it is created. The cache area can be of type, VxVM or VxFS. The cluster must be a CVM cluster. The volume layout of the cache area on remote SSDs follows the simple stripe layout, not the default FSS allocation policy of mirroring across host. If the caching operation degrades performance on a particular volume, then caching is disabled for that particular volume. The volumes that are used to create cache areas must be created on disk groups with disk group version 200 or later. However, data volumes that are created on disk groups with disk group version 190 or later can access the cache area created on FSS exported devices. **Note:** CFS write-back caching is not supported for cache areas created on remote SSDs. For more information, see the document *Veritas InfoScale SmartIO for Solid State Drives Solutions Guide*.
Campus cluster configuration	Campus clusters can be set up without the need for Fibre Channel (FC) SAN connectivity between sites.
FSS in cloud environments	The Flexible Shared Storage (FSS) Technology allows you to overcome the limitations of 'Share-Nothing' storage in cloud environments. FSS enables you to create shared-nothing clusters by sharing cloud block storage over the network.

See “Administering Flexible Storage Sharing” on page 147.

Limitations of Flexible Storage Sharing

Note the following limitations for using Flexible Storage Sharing (FSS):

- FSS is only supported on clusters of up to 64 nodes.
- Disk initialization operations should be performed only on nodes with local connectivity to the disk.
- FSS does not support the use of boot disks, opaque disks, and non-VxVM disks for network sharing.
- Hot-relocation is disabled on FSS disk groups.
- The VxVM cloned disks operations are not supported with FSS disk groups.
- FSS does not support non-SCSI3 disks connected to multiple hosts.
- Dynamic LUN Expansion (DLE) is not supported.
- FSS only supports instant data change object (DCO), created using the `vxsnap` operation or by specifying "logtype=dco dcoversion=20" attributes during volume creation.
- By default creating a mirror between SSD and HDD is not supported through `vxassist`, as the underlying mediatypes are different. To workaround this issue, you can create a volume with one mediatype, for instance the HDD, which is the default mediatype, and then later add a mirror on the SSD.
  For example:

  ```
  # vxassist -g diskgroup make volume size init=none

  # vxassist -g diskgroup mirror volume mediatype:ssd

  # vxvol -g diskgroup init active volume
  ```

  See “Administering mirrored volumes using vxassist” on page 152.

Cluster File System (CFS)

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

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

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

For detailed information, see the Storage Foundation Cluster File System High Availability Administrator's documentation.

CFS architecture

SF Oracle RAC uses CFS to manage a file system in a large database environment. Since CFS is an extension of VxFS, it operates in a similar fashion and caches metadata and data in memory (typically called buffer cache or vnode cache). CFS uses a distributed locking mechanism called Global Lock Manager (GLM) to ensure all nodes have a consistent view of the file system. GLM provides metadata and cache coherency across multiple nodes by coordinating access to file system metadata, such as inodes and free lists. The role of GLM is set on a per-file system basis to enable load balancing.

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

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

CFS communication

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

CFS file system benefits

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

- Increased manageability, including easy creation and expansion of files
  In the absence of CFS, 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.

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 any node.

CFS recovery

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

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

Comparing raw volumes and CFS for data files

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

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

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 user.

Cluster Server (VCS)

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

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

VCS architecture

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

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

SF Oracle RAC provides specific agents for VCS to manage CVM, CFS, and Oracle components like Oracle Grid Infrastructure and database (including instances).

VCS communication

VCS 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 GAB port h. HAD also receives information from other cluster members to update its own view of the cluster.

About the IMF notification module

The notification module of Intelligent Monitoring Framework (IMF) is the Asynchronous Monitoring Framework (AMF).

AMF is a kernel driver which hooks into system calls and other kernel interfaces of the operating system to get notifications on various events such as:

- When a process starts or stops.
- When a block device gets mounted or unmounted from a mount point.

AMF also interacts with the Intelligent Monitoring Framework Daemon (IMFD) to get disk group related notifications. AMF relays these notifications to various VCS Agents that are enabled for intelligent monitoring.

See "About resource monitoring" on page 37.

About resource monitoring

VCS agents poll the resources periodically based on the monitor interval (in seconds) value that is defined in the MonitorInterval or in the OfflineMonitorInterval resource type attributes. After each monitor interval, VCS invokes the monitor agent function for that resource. For example, for process offline monitoring, the process agent's monitor agent function corresponding to each process resource scans the process table in each monitor interval to check whether the process has come online. For process online monitoring, the monitor agent function queries the operating system for the status of the process id that it is monitoring. In case of the mount agent, the monitor agent function corresponding to each mount resource checks if the block device is mounted on the mount point or not. In order to determine this, the monitor function does operations such as mount table scans or runs `statfs` equivalents.

With intelligent monitoring framework (IMF), VCS supports intelligent resource monitoring in addition to poll-based monitoring. IMF is an extension to the VCS agent framework. You can enable or disable the intelligent monitoring functionality of the VCS agents that are IMF-aware. For a list of IMF-aware agents, see the *Cluster Server Bundled Agents Reference Guide*.

See "How intelligent resource monitoring works" on page 38.

See "Enabling and disabling intelligent resource monitoring for agents manually" on page 87.

Poll-based monitoring can consume a fairly large percentage of system resources such as CPU and memory on systems with a huge number of resources. This not only affects the performance of running applications, but also places a limit on how many resources an agent can monitor efficiently.

However, with IMF-based monitoring you can either eliminate poll-based monitoring completely or reduce its frequency. For example, for process offline and online monitoring, you can completely avoid the need for poll-based monitoring with IMF-based monitoring enabled for processes. Similarly for vxfs mounts, you can eliminate the poll-based monitoring with IMF monitoring enabled. Such reduction in monitor footprint will make more system resources available for other applications to consume.

Note: Intelligent Monitoring Framework for mounts is supported only for the VxFS, CFS, and NFS mount types.

With IMF-enabled agents, VCS will be able to effectively monitor larger number of resources.

Thus, intelligent monitoring has the following benefits over poll-based monitoring:

- Provides faster notification of resource state changes
- Reduces VCS system utilization due to reduced monitor function footprint
- Enables VCS to effectively monitor a large number of resources

Consider enabling IMF for an agent in the following cases:

- You have a large number of process resources or mount resources under VCS control.
- You have any of the agents that are IMF-aware.

For information about IMF-aware agents, see the following documentation:

- See the *Cluster Server Bundled Agents Reference Guide* for details on whether your bundled agent is IMF-aware.
- See the *Cluster Server Agent for Oracle Installation and Configuration Guide* for IMF-aware agents for Oracle.
- See the *Storage Foundation Cluster File System High Availability Installation Guide* for IMF-aware agents in CFS environments.

How intelligent resource monitoring works

When an IMF-aware agent starts up, the agent initializes with the IMF notification module. After the resource moves to a steady state, the agent registers the details that are required to monitor the resource with the IMF notification module. For example, the process agent registers the PIDs of the processes with the IMF notification module. The agent's imf_getnotification function waits for any resource state changes. When the IMF notification module notifies the imf_getnotification function about a resource state change, the agent framework runs the monitor agent function to ascertain the state of that resource. The agent notifies the state change to VCS which takes appropriate action.

A resource moves into a steady state when any two consecutive monitor agent functions report the state as ONLINE or as OFFLINE. The following are a few examples of how steady state is reached.

- When a resource is brought online, a monitor agent function is scheduled after the online agent function is complete. Assume that this monitor agent function reports the state as ONLINE. The next monitor agent function runs after a time

interval specified by the MonitorInterval attribute. If this monitor agent function too reports the state as ONLINE, a steady state is achieved because two consecutive monitor agent functions reported the resource state as ONLINE. After the second monitor agent function reports the state as ONLINE, the registration command for IMF is scheduled. The resource is registered with the IMF notification module and the resource comes under IMF control.The default value of MonitorInterval is 60 seconds.

A similar sequence of events applies for taking a resource offline.

- Assume that IMF is disabled for an agent type and you enable IMF for the agent type when the resource is ONLINE. The next monitor agent function occurs after a time interval specified by MonitorInterval. If this monitor agent function again reports the state as ONLINE, a steady state is achieved because two consecutive monitor agent functions reported the resource state as ONLINE.

  A similar sequence of events applies if the resource is OFFLINE initially and the next monitor agent function also reports the state as OFFLINE after you enable IMF for the agent type.

See "About the IMF notification module" on page 36.

Cluster configuration files

VCS uses two configuration files in a default configuration:

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

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

About I/O fencing

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

The fencing operation determines the following:

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

This decision prevents possible data corruption. The installer installs the I/O fencing driver, part of VRTSvxfen RPM, when you install Veritas InfoScale Enterprise. To protect data on shared disks, you must configure I/O fencing after you install Veritas InfoScale Enterprise and configure SF Oracle RAC.

With disk-based or server-based I/O fencing, coordination points can be either coordinator disks or coordination point servers (CP servers) or both. You can configure disk-based or server-based I/O fencing:

Disk-based I/O fencing	I/O fencing that uses coordinator disks is referred to as disk-based I/O fencing. Disk-based I/O fencing ensures data integrity in a single cluster.
Server-based I/O fencing	I/O fencing that uses at least one CP server system is referred to as server-based I/O fencing. Server-based fencing can include only CP servers, or a mix of CP servers and coordinator disks. Server-based I/O fencing ensures data integrity in clusters.

For detailed information, see the *Cluster Server Administrator's Guide*.

Review the following notes:

- The CP server provides an alternative arbitration mechanism without having to depend on SCSI-3 compliant coordinator disks. Data disk fencing in CVM will still require SCSI-3 I/O fencing.
- SF Oracle RAC requires at least 3 CP servers to function as coordination points in the cluster.

See the *Storage Foundation for Oracle RAC Configuration and Upgrade Guide*.

Oracle RAC components

This section provides a brief description of Oracle Clusterware/Grid Infrastructure, the Oracle Cluster Registry, application resources, and the voting disk.

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

Oracle Clusterware/Grid Infrastructure

Oracle Clusterware/Grid Infrastructure manages Oracle cluster-related functions including membership, group services, global resource management, and databases. Oracle Clusterware/Grid Infrastructure is required for every Oracle RAC instance.

Oracle Clusterware/Grid Infrastructure requires the following major components:

- A cluster interconnect that allows for cluster communications
- A private virtual IP address for cluster communications over the interconnect
- A public virtual IP address for client connections
- For Oracle 11g Release 2 and later versions, a public IP address as a Single Client Access Name (SCAN) address on the Domain Name Server (DNS) for round robin resolution to three IP addresses (recommended) or at least one IP address
- Shared storage accessible by each node

Co-existence with VCS

Oracle Clusterware/Grid Infrastructure supports co-existence with vendor clusterwares such as Cluster Server. When you install Oracle Clusterware/Grid Infrastructure on an SF Oracle RAC cluster, Oracle Clusterware/Grid Infrastructure detects the presence of VCS by checking the presence of the Veritas membership module (VCSMM) library. It obtains the list of nodes in the cluster from the VCSMM library at the time of installation.

When a node fails to respond across the interconnect, Oracle Clusterware/Grid Infrastructure waits before evicting another node from the cluster. This wait-time is defined by the CSS miss-count value. Oracle Clusterware/Grid Infrastructure sets the CSS miss-count parameter to a larger value (600 seconds) in the presence of VCS. This value is much higher than the LLT peer inactivity timeout interval. Thus, in the event of a network split-brain, the two clusterwares, VCS and Oracle Clusterware/Grid Infrastructure, do not interfere with each other's decisions on which nodes remain in the cluster. I/O fencing is allowed to decide on the surviving nodes first, followed by Oracle Clusterware/Grid Infrastructure.

VCS uses LLT for communicating between cluster nodes over private interconnects while Oracle Clusterware/Grid Infrastructure uses private IP addresses configured over the private interconnects to communicate between the cluster nodes. To coordinate membership changes between VCS and Oracle Clusterware/Grid Infrastructure, it is important to configure the Oracle Clusterware/Grid Infrastructure private IP address over the network interfaces used by LLT. VCS uses the CSSD agent to start, stop, and monitor Oracle Clusterware/Grid Infrastructure. The CSSD agent ensures that the OCR, the voting disk, and the private IP address resources required by Oracle Clusterware/Grid Infrastructure are brought online by VCS before

Oracle Clusterware/Grid Infrastructure starts. This prevents the premature startup of Oracle Clusterware/Grid Infrastructure, which causes cluster failures.

Oracle Cluster Registry

The Oracle Cluster Registry (OCR) contains cluster and database configuration and state information for Oracle RAC and Oracle Clusterware/Grid Infrastructure.

The information maintained in the OCR includes:

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

For Oracle 11gR2 and 12cR1, the OCR data exists on ASM or a cluster file system that is accessible to each node. However, for Oracle 12cR2 the OCR and voting files are stored on Oracle ASM disk groups, as Oracle 12cR2 does not support storage of cluster files directly on CFS.

Use CVM mirrored volumes to protect OCR data from failures.Oracle Clusterware/Grid Infrastructure faults nodes if OCR is not accessible because of corruption or disk failure. Oracle automatically backs up OCR data. You can also export the OCR contents before making configuration changes in Oracle Clusterware/Grid Infrastructure. This way, if you encounter configuration problems and are unable to restart Oracle Clusterware/Grid Infrastructure, you can restore the original contents.

Consult the Oracle documentation for instructions on exporting and restoring OCR contents.

Application resources

Oracle Clusterware/Grid Infrastructure application resources are similar to VCS resources. Each component controlled by Oracle Clusterware/Grid Infrastructure is defined by an application resource, including databases, instances, services, and node applications.

Unlike VCS, Oracle Clusterware/Grid Infrastructure uses separate resources for components that run in parallel on multiple nodes.

Resource profiles

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

Oracle Clusterware/Grid Infrastructure node applications

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

Voting disk

The voting disk is a heartbeat mechanism used by Oracle Clusterware/Grid Infrastructure to maintain cluster node membership.

In Oracle RAC 11g Release 2 and later versions, voting disk data exists on ASM or on a cluster file system that is accessible to each node.

The Oracle Clusterware Cluster Synchronization Service daemon (ocssd) provides cluster node membership and group membership information to RAC instances. On each node, ocssd processes write a heartbeat to the voting disk every second. If a node is unable to access the voting disk, Oracle Clusterware/Grid Infrastructure determines the cluster is in a split-brain condition and panics the node in a way that only one sub-cluster remains.

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 the following performance benefits provided by Oracle Disk Manager:

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

Databases using file systems typically incur additional overhead:

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

ODM communication

ODM uses port d to communicate with ODM instances on other nodes in the cluster.

RAC extensions

Oracle RAC relies on support services provided by VCS such as Veritas Cluster Server Membership Manager (VCSMM) to protect data integrity.

Veritas Cluster Server membership manager

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

Oracle and cache fusion traffic

Private IP addresses are required by Oracle for cache fusion traffic.

You must use UDP IPC for the database cache fusion traffic.

Periodic health evaluation of SF Oracle RAC clusters

SF Oracle RAC provides a health check utility that evaluates the components and configuration in an SF Oracle RAC cluster. The utility when invoked gathers real-time operational information on cluster components and displays the report on your system console. You must run the utility on each node in the cluster.

The utility evaluates the health of the following components:

- Low Latency Transport (LLT)
- LLT Multiplexer (LMX)
- VCSMM
- I/O fencing
- Oracle Clusterware/Grid Infrastructure
- PrivNIC and MultiPrivNIC

The health check utility is installed at `/opt/VRTSvcs/rac/healthcheck/healthcheck` during the installation of SF Oracle RAC.

The utility determines the health of the components by gathering information from configuration files or by reading the threshold values set in the health check configuration file `/opt/VRTSvcs/rac/healthcheck/healthcheck.cf`

The utility displays a warning message when the operational state of the component violates the configured settings for the component or when the health score approaches or exceeds the threshold value. You can modify the health check configuration file to set the threshold values to the desired level.

The health checks for the I/O fencing, PrivNIC, MultiPrivNIC, and Oracle Clusterware components do not use threshold settings.

Note: You must set the `ORACLE_HOME` and `CRS_HOME` parameters in the configuration file as appropriate for your setup.

Table 1-2 provides guidelines on changing the threshold values.

Table 1-2 Setting threshold values

Requirement	Setting the threshold
To detect warnings early	Reduce the corresponding threshold value.
To suppress warnings	Increase the corresponding threshold value. **Warning:** Using very high threshold values can prevent the health check utility from forecasting potential problems in the cluster. Exercise caution with high values.

Note: You can schedule periodic health evaluation of your clusters, by scheduling the utility to run as a cron job.

See "Scheduling periodic health checks for your SF Oracle RAC cluster" on page 70.

For detailed information on the list of health checks performed for each component, see the appendix *List of SF Oracle RAC health checks*.

About Virtual Business Services

Virtual Business Services provide continuous high availability and reduce frequency and duration of service disruptions for multi-tier business applications running on heterogeneous operating systems and virtualization technologies. A Virtual Business Service represents the multi-tier application as a single consolidated entity and builds on the high availability and disaster recovery provided for the individual tiers by products such as Cluster Server and ApplicationHA. Additionally, a Virtual Business Service can also represent all the assets used by the service such as arrays, hosts, and file systems, though they are not migrated between server tiers. A Virtual Business Service provides a single consolidated entity that represents a multi-tier business service in its entirety. Application components that are managed by Cluster Server or ApplicationHA can be actively managed through a Virtual Business Service.

You can configure and manage Virtual Business Services created in Veritas InfoScale Operations Manager by using Veritas InfoScale Operations Manager Virtual Business Services Availability Add-on. Besides providing all the functionality that was earlier available through Business Entity Operations Add-on, VBS Availability Add-on provides the additional ability to configure fault dependencies between the components of the multi-tier application.

Note: All the Application Entities that were created using Veritas Operations Manager Virtual Business Service Operations Add-on versions 3.1 and 4.0 are available as Virtual Business Services after you deploy the VBS Availability Add-on in Veritas InfoScale Operations Manager . Veritas InfoScale Operations Manager is a prerequisite for running Virtual Business Services.

Features of Virtual Business Services

The following VBS operations are supported:

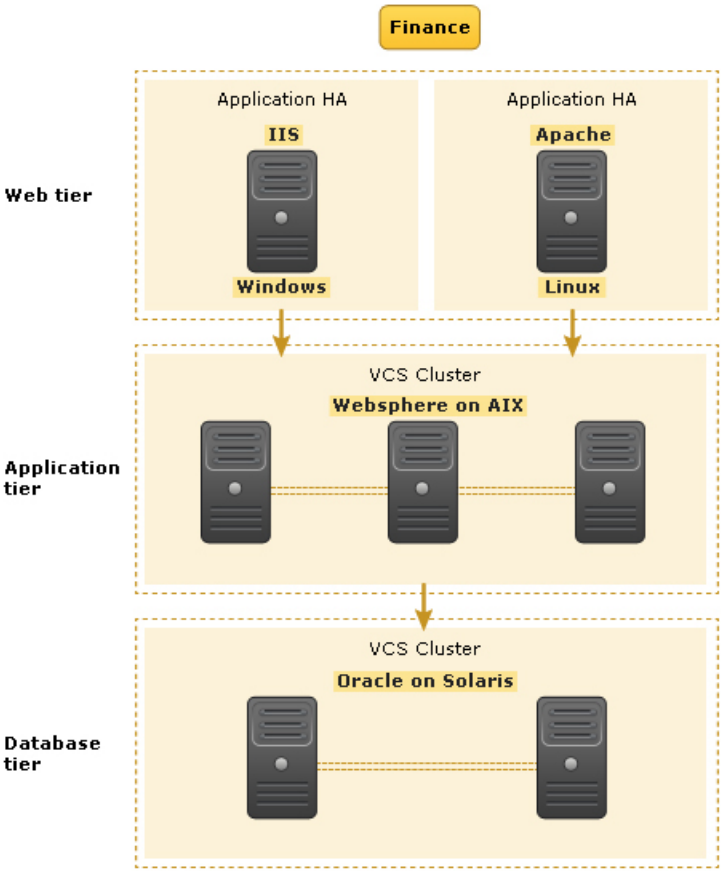
- Start Virtual Business Services from the Veritas InfoScale Operations Manager console: When a virtual business service starts, its associated service groups are brought online.
- Stop Virtual Business Services from the Veritas InfoScale Operations Manager console: When a virtual business service stops, its associated service groups are taken offline.
- Applications that are under the control of ApplicationHAcan be part of a virtual business service. ApplicationHAenables starting, stopping, and monitoring of an application within a virtual machine. If applications are hosted on VMware virtual machines, you can configure the virtual machines to automatically start or stop when you start or stop the virtual business service, provided the vCenter for those virtual machines has been configured in Veritas InfoScale Operations Manager.
- Define dependencies between service groups within a virtual business service: The dependencies define the order in which service groups are brought online and taken offline. Setting the correct order of service group dependency is critical to achieve business continuity and high availability. You can define the dependency types to control how a tier reacts to high availability events in the underlying tier. The configured reaction could be execution of a predefined policy or a custom script.
- Manage the virtual business service from Veritas InfoScale Operations Manager or from the clusters participating in the virtual business service.
- Recover the entire virtual business service to a remote site when a disaster occurs.

Sample virtual business service configuration

This section provides a sample virtual business service configuration comprising a multi-tier application. Figure 1-9 shows a Finance application that is dependent on components that run on three different operating systems and on three different clusters.

- Databases such as Oracle running on Solaris operating systems form the database tier.
- Middleware applications such as WebSphere running on AIX operating systems form the middle tier.
- Web applications such as Apache and IIS running on Windows and Linux virtual machines form the Web tier.
  Each tier can have its own high availability mechanism. For example, you can use Cluster Server for the databases and middleware applications for the Web servers.

Figure 1-9 Sample virtual business service configuration



Each time you start the Finance business application, typically you need to bring the components online in the following order – Oracle database, WebSphere, Apache and IIS. In addition, you must bring the virtual machines online before you start the Web tier. To stop the Finance application, you must take the components offline in the reverse order. From the business perspective, the Finance service is unavailable if any of the tiers becomes unavailable.

When you configure the Finance application as a virtual business service, you can specify that the Oracle database must start first, followed by WebSphere and the Web servers. The reverse order automatically applies when you stop the virtual business service. When you start or stop the virtual business service, the components of the service are started or stopped in the defined order.

For more information about Virtual Business Services, refer to the *Virtual Business Service–Availability User's Guide*.

About Veritas InfoScale Operations Manager

Veritas InfoScale Operations Manager provides a centralized management console for Veritas InfoScale products. You can use Veritas InfoScale Operations Manager to monitor, visualize, and manage storage resources and generate reports.

Veritas recommends using Veritas InfoScale Operations Manager to manage Storage Foundation and Cluster Server environments.

You can download Veritas InfoScale Operations Manager from https://sort.veritas.com/.

Refer to the Veritas InfoScale Operations Manager documentation for installation, upgrade, and configuration instructions.

The Veritas Enterprise Administrator (VEA) console is no longer packaged with Veritas InfoScale products. If you want to continue using VEA, a software version is available for download from https://www.veritas.com/product/storage-management/infoscale-operations-manager. Storage Foundation Management Server is deprecated.

If you want to manage a single cluster using Cluster Manager (Java Console), a version is available for download from https://www.veritas.com/product/storage-management/infoscale-operations-manager. You cannot manage the new features of this release using the Java Console. Cluster Server Management Console is deprecated.

Chapter 2

Administering SF Oracle RAC and its components

This chapter includes the following topics:

Administering SF Oracle RAC

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

- Setting the environment variables
  See "Setting the environment variables for SF Oracle RAC" on page 52.
- Starting or stopping SF Oracle RAC on each node
  See "Starting or stopping SF Oracle RAC on each node" on page 52.
- Applying Oracle patches on SF Oracle RAC nodes
  See "Applying Oracle patches on SF Oracle RAC nodes" on page 60.

- Installing Veritas Volume Manager or Veritas File System patches or ODM patches on SF Oracle RAC nodes
  See "Installing Veritas Volume Manager, Veritas File System, or ODM patches on SF Oracle RAC nodes" on page 64.
- Applying operating system updates on SF Oracle RAC nodes
  See "Applying operating system updates on SF Oracle RAC nodes" on page 64.
- Adding storage to an SF Oracle RAC cluster
  See "Adding storage to an SF Oracle RAC cluster" on page 65.
- Recovering from storage failure
  See "Recovering from storage failure" on page 67.
- Backing up and restoring the Oracle database using Veritas NetBackup
  See "Backing up and restoring Oracle database using Veritas NetBackup" on page 67.
- Enhancing the performance of SF Oracle RAC clusters
  See "Enhancing the performance of SF Oracle RAC clusters" on page 68.
- Administering SmartIO
  See "Administering SmartIO" on page 69.
- Creating snapshots for offhost processing
  See "Creating snapshots for offhost processing" on page 69.
- Managing database storage efficiently using SmartTier
  See "Managing database storage efficiently using SmartTier" on page 70.
- Optimizing database storage using Thin Provisioning and SmartMove
  See "Optimizing database storage using Thin Provisioning and SmartMove" on page 70.
- Scheduling periodic health checks for your SF Oracle RAC cluster
  See "Scheduling periodic health checks for your SF Oracle RAC cluster" on page 70.
- Using environment variables to start and stop VCSMM modules
  See "Using environment variables to start and stop VCSMM modules" on page 71.
- Verifying the nodes in an SF Oracle RAC cluster
  See "Verifying the nodes in an SF Oracle RAC cluster" on page 72.

Note: The commands used for the Red Hat Enterprise Linux (RHEL) operating system in this document also apply to supported RHEL-compatible distributions.

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

See "About troubleshooting SF Oracle RAC" on page 166.

Setting the environment variables for SF Oracle RAC

Set the MANPATH variable in the .profile file (or other appropriate shell setup file for your system) to enable viewing of manual pages.

Based on the shell you use, type one of the following:

For sh, ksh, or bash

```
# export MANPATH=$MANPATH:\
/opt/VRTS/man
```

For csh

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

For csh

```
# setenv MANPATH $MANPATH:/opt/VRTS/man
```

Some terminal programs may display garbage characters when you view the man pages. Set the environment variable LC_ALL=C to resolve this issue.

Set the PATH environment variable in the .profile file (or other appropriate shell setup file for your system) on each system to include installation and other commands.

Based on the shell you use, type one of the following:

For sh, ksh, or bash

```
# PATH=/usr/sbin:/sbin:/usr/bin:\
/opt/VRTS/bin\
$PATH; export PATH
```

Starting or stopping SF Oracle RAC on each node

You can start or stop SF Oracle RAC on each node in the cluster using the SF Oracle RAC installer or manually.

To start SF Oracle RAC	Using installer: See "Starting SF Oracle RAC using the script-based installer" on page 53. Manual: See "Starting SF Oracle RAC manually on each node" on page 53.
To stop SF Oracle RAC	Using installer: See "Stopping SF Oracle RAC using the script-based installer" on page 56. Manual: See "Stopping SF Oracle RAC manually on each node" on page 56.

Starting SF Oracle RAC using the script-based installer

Run the installer with the `-start` option to start SF Oracle RAC on each node.

Note: Start SF Oracle RAC on all nodes in the cluster. Specifying only some of the nodes in the cluster may cause some of the components that depend on GAB seeding to fail.

To start SF Oracle RAC using the installer

1. Log into one of the nodes in the cluster as the root user.
2. Start SF Oracle RAC:

```
# /opt/VRTS/install/installer -start sys1 sys2
```

Starting SF Oracle RAC manually on each node

Perform the steps in the following procedures to start SF Oracle RAC manually on each node.

To start SF Oracle RAC manually on each node

1. Log into each node as the root user.
2. Start LLT:

   For RHEL 7, SLES 12, and supported RHEL distributions:

   ```
   # systemctl start llt
   ```

   For earlier versions of RHEL, SLES and supported RHEL distributions:

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

3. Start GAB:

   For RHEL 7, SLES 12, and supported RHEL distributions:

   ```
   # systemctl start gab
   ```

   For earlier versions of RHEL, SLES and supported RHEL distributions:

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

4. Start fencing:

   For RHEL 7, SLES 12, and supported RHEL distributions:

   ```
   # systemctl start vxfen
   ```

   For earlier versions of RHEL, SLES and supported RHEL distributions:

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

5. Start GLM:

   For RHEL 7, SLES 12, and supported RHEL distributions:

   ```
   # systemctl start vxglm
   ```

   For earlier versions of RHEL, SLES and supported RHEL distributions:

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

6 Start GMS:

For RHEL 7, SLES 12, and supported RHEL distributions:

```
# systemctl start vxgms
```

For earlier versions of RHEL, SLES and supported RHEL distributions:

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

7 Start VCSMM:

For RHEL 7, SLES 12, and supported RHEL distributions:

```
# systemctl start vcsmm
```

For earlier versions of RHEL, SLES and supported RHEL distributions:

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

8 Start ODM:

For RHEL 7, SLES 12, and supported RHEL distributions:

```
# systemctl start vxodm
```

For earlier versions of RHEL, SLES and supported RHEL distributions:

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

9 Start VCS, CVM, and CFS:

```
# hastart
```

10 Verify that all GAB ports are up and running:

```
# gabconfig -a
GAB Port Memberships
===============================================================
Port a gen   564004 membership 01
Port b gen   564008 membership 01
Port d gen   564009 membership 01
Port f gen   564024 membership 01
Port h gen   56401a membership 01
Port m gen   56401c membership 01
Port o gen   564007 membership 01
Port u gen   564021 membership 01
Port v gen   56401d membership 01
Port w gen   56401f membership 01
Port y gen   56401c membership 01
```

Stopping SF Oracle RAC using the script-based installer

Run the installer with the `-stop` option to stop SF Oracle RAC on each node.

To stop SF Oracle RAC using the installer

1 Log into one of the nodes in the cluster as the root user.

2 Stop VCS:

```
# hastop -all
```

3 Stop SF Oracle RAC:

```
# /opt/VRTS/install/installer -stop sys1 sys2
```

Stopping SF Oracle RAC manually on each node

Perform the steps in the following procedures to stop SF Oracle RAC manually on each node.

To stop SF Oracle RAC manually on each node

1. Stop the Oracle database.

   If the Oracle RAC instance is managed by VCS, log in as the root user and take the service group offline:

   ```
   # hagrp -offline oracle_group -sys node_name
   ```

   If the Oracle database instance is not managed by VCS, log in as the Oracle user on one of the nodes and shut down the instance:

   For Oracle RAC 12c:

   ```
   $ srvctl stop instance -db db_name \
   -node node_name
   ```

   For Oracle RAC 11.2.0.2 and later versions of 11g Release 2:

   ```
   $ srvctl stop instance -d db_name \
   -n node_name
   ```

2. Stop all applications that are not configured under VCS but dependent on Oracle RAC or resources controlled by VCS. Use native application commands to stop the application.
3. Unmount the CFS file systems that are not managed by VCS.
   - Make sure that no processes are running which make use of mounted shared file system. To verify that no processes use the VxFS or CFS mount point:

     ```
     # mount | grep vxfs | grep cluster
     ```

     ```
     # fuser -cu /mount_point
     ```

   - Unmount the CFS file system:

     ```
     # umount /mount_point
     ```

4. Take the VCS service groups offline:

   ```
   # hagrp -offline group_name -sys node_name
   ```

   Verify that the VCS service groups are offline:

   ```
   # hagrp -state group_name
   ```

5. Unmount the VxFS file systems that are not managed by VCS.

- Make sure that no processes are running which make use of mounted shared file system. To verify that no processes use the VxFS or CFS mount point:

  ```
  # mount | grep vxfs

  # fuser -cu /mount_point
  ```

- Unmount the VxFS file system:

  ```
  # umount /mount_point
  ```

6 Verify that no VxVM volumes (other than VxVM boot volumes) remain open. Stop any open volumes that are not managed by VCS.

7 Stop VCS, CVM and CFS:

```
# hastop -local
```

Verify that the ports 'f', 'm', 'u', 'v', 'w', 'y', and 'h' are closed:

```
# gabconfig -a
GAB Port Memberships
===============================================================
Port a gen   761f03 membership 01
Port b gen   761f08 membership 01
Port d gen   761f02 membership 01
Port o gen   761f01 membership 01
```

8 Stop ODM:

For RHEL 7, SLES 12, and supported RHEL distributions:

```
# systemctl stop vxodm
```

For earlier versions of RHEL, SLES and supported RHEL distributions:

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

9 Stop GMS:

For RHEL 7, SLES 12, and supported RHEL distributions:

```
# systemctl stop vxgms
```

For earlier versions of RHEL, SLES and supported RHEL distributions:

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

10 Stop GLM:

For RHEL 7, SLES 12, and supported RHEL distributions:

```
# systemctl stop vxglm
```

For earlier versions of RHEL, SLES and supported RHEL distributions:

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

11 Stop VCSMM:

For RHEL 7, SLES 12, and supported RHEL distributions:

```
# systemctl stop vcsmm
```

For earlier versions of RHEL, SLES and supported RHEL distributions:

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

12 Stop fencing:

For RHEL 7, SLES 12, and supported RHEL distributions:

```
# systemctl stop vxfen
```

For earlier versions of RHEL, SLES and supported RHEL distributions:

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

13 Stop GAB:

For RHEL 7, SLES 12, and supported RHEL distributions:

```
# systemctl stop gab
```

For earlier versions of RHEL, SLES and supported RHEL distributions:

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

14 Stop LLT:

For RHEL 7, SLES 12, and supported RHEL distributions:

```
# systemctl stop llt
```

For earlier versions of RHEL, SLES and supported RHEL distributions:

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

Applying Oracle patches on SF Oracle RAC nodes

Before installing any Oracle RAC patch or patchset software:

- Review the latest information on supported Oracle RAC patches and patchsets: https://www.veritas.com/support/en_US/article.DOC4039
- You must have installed the base version of the Oracle RAC software.

To apply Oracle patches

1. Freeze the service group that contains the Oracle resource and the cssd resource:

   ```
   # hagrp -freeze grp_name
   ```

2. Install the patches or patchsets required for your Oracle RAC installation. See the Oracle documentation that accompanies the patch or patchset.

   If the patch is for Oracle database software, after applying the patch, before starting the database or database instance, relink the SF Oracle RAC libraries with Oracle libraries.

3. Unfreeze the service group that contains the Oracle resource and the cssd resource:

   ```
   # hagrp -unfreeze grp_name
   ```

Migrating Pluggable Databases (PDB) between Container Databases (CDB)

You can use the `hapdbmigrate` utility to perform a planned end-to-end migration of pluggable databases across containers of the same database version. The destination container database may be on the same node or on another node. The PDB must have its own independent storage.

If the PDB has parent groups, they will be unlinked and frozen by the utility during the migration. After successful migration, they will be relinked and unfrozen by the utility.

The migration is supported on database version 12.1.0.2.

The configuration must meet the following requirements:

- The PDB and CDB databases must be on Veritas Cluster File System (CFS) or Veritas File system (VxFS).
- There must be no parent resources for the PDB resource.

- The PDB must be plugged in to the source CDB. The PDB resources may or may not be offline.
- The source and destination CDB resources for the migration must be different.
- The version of the destination CDB must be the same as the source CDB.
- The PDB to be migrated is mounted on an independent file system (separate mount points for CDB and PDB datafiles).
- The CDB and PDB resources must be configured in the same service group.

The utility performs the following actions during the migration:

- Unlinks and freezes the parent groups, if any, depending on the source CDB group where the PDB resource to be migrated is configured.
- Takes PDB resources offline.
- Unplugs the PDB from the CDB and creates an XML file, `<pdb_res_name>_<dest_cdb_res_name>.xml` in the XML directory provided by the user.
- Drops the PDB from the CDB keeping the datafiles.
- Takes offline all the PDB child resources. Unlinks the PDB resource from the source CDB resource and deletes the PDB resource. Unlinks and deletes all PDB child resources.
- Recreates the PDB resource and all its children with original dependencies in the destination CDB service group.
- Brings online the PDB child resources on all the nodes where the destination CDB service group is online
- Plugs the PDB in the destination CDB.
- Brings the PDB resource online after successful plugging. Unfreezes and links the parent group to the source CDB group.

The `hapdbmigrate` utility performs certain pre-requisite checks before the migration to verify that the cluster is ready for PDB migration. If the utility encounters any issues, you will need to manually fix the issues.

The utility is present in the `$VCSHOME/bin` directory.

The log files of the migration are located at `$VCSLOG/log/hapdbmigrate.log`. The logs are rotated after the file exceeds 5 MB and is saved in .gz format `hapdbmigrate.log[1..7].gz`.

You can find sample configuration files for Oracle RAC at `/etc/VRTSvcs/conf/sample_rac/sfrac18_main.cf`.

Note: Ensure that only one instance of the `hapdbmigrate` utility is running at a time.

To migrate Pluggable Databases (PDB) between Container Databases (CDB)

1. Back up the VCS configuration file `/etc/VRTSvcs/conf/config/main.cf`:

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

2. Verify that the high availability daemon (had) is running on all the nodes in the cluster.

   ```
   # hastatus -summary
   ```

3. Verify that there are no resources in faulted or unknown state.

   ```
   # hares -state|grep FAULTED
   # hares -state|grep UNKNOWN
   ```

4. Verify that the `PDBName` attribute is present for the PDB resource with the correct value in the `main.cf` configuration file.

5. Verify that the source and destination CDB resources are online.

   ```
   # hares -state resname
   ```

6. On the destination CDB, verify the following:

   - The destination CDB is not in `suspended` mode.
     Any instance of the destination CDB is not in mounted state.
     See the Oracle documentation for more information.
   - If any instance of the destination CDB is in `restricted` state, ensure that the PDB resource you want to migrate has the `StartUpOpt` attribute set to `restricted`.

     ```
     # haconf -makerw
     # hares -modify pdb1 StartUpOpt \
     RESTRICTED
     # haconf -dump -makero
     ```

7 Verify that existing dependencies do not conflict with the migration process.

The PDB child resources must not be dependent on the CDB resource or any of its child resources.

The PDB child resources must not have any parent, which is not a part of the PDB child hierarchy.

8 Verify that the XML data directory has read and write permissions for the "oracle" user. The XML data directory must be located either on PDB mounts or at a location accessible to both source and destination CDBs.

9 Run the `hapdbmigrate` utility as the root user:

Note: If there are parent groups dependent on the source CDB group, specify the `-ignoreparentgrp` option.

```
# $VCS_HOME/bin/hapdbmigrate -pdbres pdb_resname -cdbres cdb_resname \
-XMLdirectory xml_directory [-ignoreparentgrp] [-prechecks]
  -pdbres: Name of the PDB resource, which needs to be migrated
 -cdbres: Name of the CDB resource, where the PDB needs to migrate
 -XMLdirectory: XML directory location for the unplugged PDB
-ignoreparentgrp: Utility proceeds even
if the PDB group has parent groups
-prechecks: Performs prechecks and validation
-help|h: Prints usage
```

The migration log file is located at `$VCSLOG/log/hapdbmigrate.log`.

10 Verify that the PDB resource is online on the destination CDB.

```
# hares -state pdb_resname
```

11 Relink the parent service group of the source CDB group manually to the destination CDB group, if it depends on the migrated PDB.

```
# haconf -makerw
# hagrp -dep parent_sg
#Parent           Child             Relationship
parent_sg       source_CDB         online local firm
# hagrp -offline parent_sg -any
# hagrp -unlink parent_sg source_CDB
# hagrp -link parent_sg dest_CDB online local firm
# hagrp -dep parent_sg
#Parent           Child             Relationship
parent_sg       dest_CDB           online local firm
# haconf -dump -makero
```

Installing Veritas Volume Manager, Veritas File System, or ODM patches on SF Oracle RAC nodes

Perform the steps on each node in the cluster to install Veritas Volume Manager, Veritas File System, or ODM patches.

To install Veritas Volume Manager, Veritas File System, or ODM patches on SF Oracle RAC nodes

1 Log in as the root user.

2 Stop SF Oracle RAC on each node.

See "Stopping SF Oracle RAC using the script-based installer" on page 56.

3 Install the VxVM, VxFS, or ODM patch as described in the corresponding patch documentation.

4 If there are applications that are not managed by VCS, start the applications manually using native application commands.

Applying operating system updates on SF Oracle RAC nodes

If you need to apply updates to the base version of the operating system, perform the steps in this section on each node of the cluster, one node at a time.

Ensure that you check the following website for the operating system updates that are supported by Veritas:

https://sort.veritas.com/

To apply operating system updates

1. Log in to the node as the root user and change to `/opt/VRTS/install` directory:

   ```
   # cd /opt/VRTS/install
   ```

2. Take the VCS service groups offline:

   ```
   # hagrp -offline grp_name -sys node_name
   ```

3. Stop SF Oracle RAC:

   ```
   # ./installer -stop
   ```

4. Upgrade the operating system.

   See the operating system documentation.

5. If the node is not rebooted after the operating system upgrade, reboot the node:

   ```
   # shutdown -r now
   ```

6. Repeat all the steps on each node in the cluster.

Adding storage to an SF Oracle RAC cluster

You can add storage to an SF Oracle RAC cluster in the following ways:

Add a disk to a disk group	Use the `vxdg` command to add a disk to a disk group. See the `vxdg` (1M) manual page for information on various options. See "To add storage to an SF Oracle RAC cluster by adding a disk to a disk group" on page 66.

Extend the volume space on a disk group	Use the `vxresize` command to change the length of a volume containing a file system. It automatically locates available disk space on the specified volume and frees up unused space to the disk group for later use. See the `vxresize` (1M) manual page for information on various options. See "To add storage to an SF Oracle RAC cluster by extending the volume space on a disk group" on page 67.

Note: The steps in the following procedures are applicable to non-FSS configurations only.

To add storage to an SF Oracle RAC cluster by adding a disk to a disk group

- Add a disk to the disk group:

```
# vxdg -g dg_name adddisk disk_name
```

To add storage to an SF Oracle RAC cluster by extending the volume space on a disk group

1 Determine the length by which you can increase an existing volume.

```
# vxassist [-g diskgroup] maxsize
```

For example, to determine the maximum size the volume `oradatavol` in the disk group `oradatadg` can grow, given its attributes and free storage available:

```
# vxassist -g oradatadg maxsize
```

2 Extend the volume, as required. You can extend an existing volume to a certain length by specifying the new size of the volume (the new size must include the additional space you plan to add). You can also extend a volume by a certain length by specifying the additional amount of space you want to add to the existing volume.

To extend a volume to a certain length — For example, to extend the volume `oradata_vol` of size 10 GB in the disk group `oradata_dg` to 30 GB:

```
# vxresize -g oradata_dg \
oradata_vol 30g
```

To extend a volume by a certain length — For example, to extend the volume `oradata_vol` of size 10 GB in the disk group `oradata_dg` by 10 GB:

```
# vxresize -g oradata_dg \
oradata_vol +10g
```

Recovering from storage failure

Veritas Volume Manager (VxVM) protects systems from disk and other hardware failures and helps you to recover from such events. Recovery procedures help you prevent loss of data or system access due to disk and other hardware failures.

For information on various failure and recovery scenarios, see the *Veritas InfoScale Troubleshooting Guide*.

Backing up and restoring Oracle database using Veritas NetBackup

Veritas NetBackup integrates the database backup and recovery capabilities of the Oracle Recovery Manager (RMAN) with the backup and recovery management capabilities of NetBackup.

NetBackup for Oracle supports the following types of backups:

- Full backup
- Incremental backup
- Multilevel incremental backup
- Differential incremental backup
- Cumulative incremental backup

While Oracle RMAN performs backup, restore, and recovery of physical Oracle database objects (data files, tablespaces, control files, and archived redo logs), the NetBackup for Oracle XML export and XML import utilities provide backup and restore of logical database objects (tables, users, and rows).

See the *Veritas NetBackup for Oracle Administrator's Guide*.

Enhancing the performance of SF Oracle RAC clusters

The main components of clustering that impact the performance of an SF Oracle RAC cluster are:

- Kernel components, specifically LLT and GAB
- VCS engine (had)
- VCS agents

Each VCS agent process has two components—the agent framework and the agent functions. The agent framework provides common functionality, such as communication with the HAD, multithreading for multiple resources, scheduling threads, and invoking functions. Agent functions implement functionality that is particular to an agent.

For various options provided by the clustering components to monitor and enhance performance, see the chapter "VCS performance considerations" in the *Cluster Server Administrator's Guide*.

Veritas Volume Manager can improve system performance by optimizing the layout of data storage on the available hardware.

For more information on tuning Veritas Volume Manager for better performance, see the chapter "Performance monitoring and tuning" in the *Veritas Storage Foundation Administrator's Guide*.

Volume Replicator Advisor (VRAdvisor) is a planning tool that helps you determine an optimum Volume Replicator (VVR) configuration.

For installing VRAdvisor and evaluating various parameters using the data collection and data analysis process, see the *Veritas InfoScale Replication Administrator's Guide*.

Mounting a snapshot file system for backups increases the load on the system as it involves high resource consumption to perform copy-on-writes and to read data blocks from the snapshot. In such situations, cluster snapshots can be used to do off-host backups. Off-host backups reduce the load of a backup application from the primary server. Overhead from remote snapshots is small when compared to overall snapshot overhead. Therefore, running a backup application by mounting a snapshot from a relatively less loaded node is beneficial to overall cluster performance.

Administering SmartIO

The SmartIO feature of Storage Foundation and High Availability Solutions (SFHA Solutions) enables data efficiency on your solid-state drives (SSD) through I/O caching.

SmartIO uses a cache area on target devices. The cache area is the storage space that SmartIO uses to store the cached data and the metadata about the cached data. The type of the cache area determines whether it supports VxFS caching or VxVM caching. To start using SmartIO, you can create a cache area with a single command, while the application is online. When the application issues an I/O request, SmartIO checks to see if the I/O can be serviced from the cache. As applications access data from the underlying volumes or file systems, certain data is moved to the cache based on the internal heuristics. Subsequent I/Os are processed from the cache.

SmartIO supports read caching for the VxFS file systems that are mounted on VxVM volumes, in several caching modes and configurations. SmartIO also supports block-level read caching for applications running on VxVM volumes.

Note: SmartIO writeback caching is currently not supported in SF Oracle RAC environments.

For more information on using and administering SmartIO, see the *Veritas InfoScale SmartIO for Solid State Drives Solutions Guide*.

Creating snapshots for offhost processing

You can capture a point-in-time copy of actively changing data at a given instant. You can then perform system backup, upgrade, and other maintenance tasks on the point-in-time copies while providing continuous availability of your critical data.

If required, you can offload processing of the point-in-time copies onto another host to avoid contention for system resources on your production server.

See the *Veritas InfoScale Solutions Guide*.

Managing database storage efficiently using SmartTier

SmartTier matches data storage with data usage requirements. After data matching, the data can be relocated based upon data usage and other requirements determined by the storage or database administrator (DBA). The technology enables the database administrator to manage data so that less frequently used data can be moved to slower, less expensive disks. This also permits the frequently accessed data to be stored on faster disks for quicker retrieval.

See the *Veritas InfoScale Storage and Availability Management for Oracle Databases* guide.

Optimizing database storage using Thin Provisioning and SmartMove

Thin Provisioning is a storage array feature that optimizes storage use by automating storage provisioning. With Storage Foundation's SmartMove feature, Veritas File System (VxFS) lets Veritas Volume Manager (VxVM) know which blocks have data. VxVM, which is the copy engine for migration, copies only the used blocks and avoids the empty spaces, thus optimizing thin storage utilization.

See the *Veritas InfoScale Solutions Guide*.

Scheduling periodic health checks for your SF Oracle RAC cluster

You can manually invoke the health check utility at any time or schedule the utility to run as a cron job.

If the health check completes without any warnings, the following message is displayed:

```
Success: The SF Oracle RAC components are working fine on this node.
```

If you manually invoke the health check, run the utility on each node in the cluster. The results of the check are displayed on the console.

To run health checks on a cluster

1. Log in as the root user on each node in the cluster.
2. Using `vi` or any text editor, modify the health check parameters as required for your installation setup.

   Note: You must set the ORACLE_HOME and CRS_HOME parameters in the configuration file as appropriate for your setup.

   ```
   # cd /opt/VRTSvcs/rac/healthcheck/
   # vi healthcheck.cf
   ```

3. Run the health check utility:

   ```
   # ./healthcheck
   ```

If you want to schedule periodic health checks for your cluster, create a cron job that runs on each node in the cluster. Redirect the health check report to a file.

Using environment variables to start and stop VCSMM modules

The start and stop environment variables for VCSMM define the default behavior to start the modules during system restart or stop the modules during system shutdown.

Note: The startup and shutdown of VCSMM is inter-dependent on other stack modules such as LLT, GAB, fencing, and ODM. For a clean startup or shutdown of SF Oracle RAC, you must either enable or disable startup and shutdown for all these modules.

Table 2-1 lists the environment variables to start and stop VCSMM.

Table 2-1 Environment variables to start and stop VCSMM

Variable	Description
VCSMM_START	Startup mode for the VCSMM driver. By default, the VCSMM driver is enabled to start up after a system reboot. This environment variable is defined in the following file: `/etc/sysconfig/vcsmm` Default: 1

Table 2-1 Environment variables to start and stop VCSMM *(continued)*

Variable	Description
VCSMM_STOP	Shutdown mode for the VCSMM driver. By default, the VCSMM driver is enabled to stop during a system shutdown. This environment variable is defined in the following file: `/etc/sysconfig/vcsmm` Default: 1

Verifying the nodes in an SF Oracle RAC cluster

Table 2-2 lists the various options that you can use to periodically verify the nodes in your cluster.

Table 2-2 Options for verifying the nodes in a cluster

Type of check	Description
Veritas Services and Operations Readiness Tools (SORT)	Use the Veritas Services and Operations Readiness Tools (SORT) to evaluate your systems before and after any installation, configuration, upgrade, patch updates, or other routine administrative activities. The utility performs a number of compatibility and operational checks on the cluster that enable you to diagnose and troubleshoot issues in the cluster. The utility is periodically updated with new features and enhancements. For more information and to download the utility, visit http://sort.veritas.com.
SF Oracle RAC health checks	SF Oracle RAC provides a health check utility that examines the functional health of the components in an SF Oracle RAC cluster. The utility when invoked gathers real-time operational information on cluster components and displays the report on your system console. You must run the utility on each node in the cluster. To schedule periodic health checks: See "Periodic health evaluation of SF Oracle RAC clusters" on page 45.

Administering VCS

This section provides instructions for the following VCS administration tasks:

- Viewing available Veritas devices and drivers
  See "Viewing available Veritas device drivers" on page 75.

- Starting and stopping VCS
  See "Starting and stopping VCS" on page 77.
- Environment variables to start and stop VCS modules
  See "Environment variables to start and stop VCS modules" on page 77.
- Adding and removing LLT links
  See "Adding and removing LLT links" on page 81.
- Displaying the cluster details and LLT version for LLT links
  See "Displaying the cluster details and LLT version for LLT links" on page 86.
- Configuring aggregated interfaces under LLT
  See "Configuring aggregated interfaces under LLT" on page 84.
- Configuring destination-based load balancing for LLT
  See "Configuring destination-based load balancing for LLT" on page 87.
- Enabling and disabling intelligent resource monitoring
  See "Enabling and disabling intelligent resource monitoring for agents manually" on page 87.
- Administering the AMF kernel driver
  See "Administering the AMF kernel driver" on page 90.

If you encounter issues while administering VCS, see the troubleshooting section in the *Cluster Server Administrator's Guide* for assistance.

About managing VCS modules

The initialization daemon of the operating system manages the start, stop, restart, and status of the AMF, GAB, LLT, VCS, and VxFEN modules. The service files and the script files for these modules are deployed at the appropriate location at the time of installation. Table 2-3 lists these files and their locations.

`systemd` is an initialization system that controls how services are started, stopped, or otherwise managed on RHEL 7 and SLES12 or later systems. The LSB files for the VCS stack have been replaced with the corresponding `systemd` unit service files. In the older RHEL distributions, the init.d daemon managed the related services. In newer RHEL distributions, `systemd` manages them as unit service files.

The following VCS unit service files and startup scripts help provide `systemd` support for those services in Linux:

Table 2-3 Unit service files and script files for `systemd` support

Unit service file	Corresponding script file (*SourcePath*)
/usr/lib/systemd/system/amf.service	/opt/VRTSamf/bin/amf
/usr/lib/systemd/system/gab.service	/opt/VRTSgab/gab
/usr/lib/systemd/system/llt.service	/opt/VRTSllt/llt
/usr/lib/systemd/system/vcs.service	/opt/VRTSvcs/bin/vcs
/usr/lib/systemd/system/vcsmm.service	/opt/VRTSvcs/rac/bin/vcsmm
/usr/lib/systemd/system/vxfen.service	/opt/VRTSvcs/vxfen/bin/vxfen

To start, stop, restart, or view the status of one of these services, use the following command:

```
systemctl [start | stop | restart | status] unitServiceFile
```

Note: The status option of the `systemctl` command displays only the status of the unit service file, like whether it is active, inactive, or failed.

To view the actual status information about the module or to view the HAD status, use:

```
serviceSourceScript status
```

For example:

```
/opt/VRTSvcs/bin/vcs status
```

To view the source path of any service, you can use the `systemctl` command as follows:

```
# systemctl show unitServiceFile -p SourcePath
```

For example:

```
# systemctl show vcs -p SourcePath
```

The `systemctl` command displays the source path as follows:

```
SourcePath=/opt/VRTSvcs/bin/vcs
```

Wherever `systemd` support is available, all the process in the VCS stack start in system.slice instead of user.slice.

Note: You can start or stop the `CmdServer` service independently of the VCS service by using the `systemctl` commands like you do for the other VCS modules.

Viewing available Veritas device drivers

To view the available Veritas devices:

```
# cat /proc/devices
Character devices:
  1 mem
  4 /dev/vc/0
  4 tty
  4 ttyS
  5 /dev/tty
  5 /dev/console
  5 /dev/ptmx
  6 lp
  7 vcs
 10 misc
 13 input
 21 sg
 29 fb
128 ptm
136 pts
162 raw
180 usb
189 usb_device
199 VxVM
200 VxSPEC
202 cpu/msr
203 cpu/cpuid
246 vxgms
247 amf
248 vcsmm
249 vxglm
250 vxfen
251 gab
252 llt
253 uio
254 pcmcia

Block devices:
  1 ramdisk
```

```
  3 ide0
  8 sd
  9 md
 65 sd
 66 sd
 67 sd
 68 sd
 69 sd
 70 sd
 71 sd
128 sd
129 sd
130 sd
131 sd
132 sd
133 sd
134 sd
135 sd
199 VxVM
201 VxDMP
253 device-mapper
254 mdp
```

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

For example:

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

```
# lsmod |grep "^gab"
gab      246420  11 vxfen,vcsmm
```

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

```
# lsmod |grep "^vx"
vxfen                  260888  1
vxodm                  159820  1
vxglm                  267400  2
vxgms                  285692  2
vxspec                   4360  4
vxio                  3090616  3 vxspec
vxdmp                  341064  92 vxspec,vxio
vxportal                 7820  2
vxfs                  2326872  11 vxportal,fdd
```

Starting and stopping VCS

This section describes how to start and stop the VCS.

To start VCS

- On each node, start VCS:

```
# hastart
```

To stop VCS

- On each node, stop VCS:

```
# hastop -local
```

You can also use the command `hastop -all` to stop the VCS cluster on all the nodes in cluster at the same time; however, make sure that you wait for port 'h' to close before restarting VCS.

Environment variables to start and stop VCS modules

The start and stop environment variables for AMF, LLT, GAB, VxFEN, and VCS engine define the default VCS behavior to start these modules during system restart or stop these modules during system shutdown.

Note: The startup and shutdown of AMF, LLT, GAB, VxFEN, and VCS engine are inter-dependent. For a clean startup or shutdown of SF Oracle RAC, you must either enable or disable the startup and shutdown modes for all these modules.

Table 2-4 Start and stop environment variables for VCS

Environment variable	Definition and default value
AMF_START	Startup mode for the AMF driver. By default, the AMF driver is enabled to start up after a system reboot. This environment variable is defined in the following file: `/etc/sysconfig/amf` Default: 1

Table 2-4 Start and stop environment variables for VCS *(continued)*

Environment variable	Definition and default value
AMF_STOP	Shutdown mode for the AMF driver. By default, the AMF driver is enabled to stop during a system shutdown. This environment variable is defined in the following file: `/etc/sysconfig/amf` Default: 1
LLT_START	Startup mode for LLT. By default, LLT is enabled to start up after a system reboot. This environment variable is defined in the following file: `/etc/sysconfig/llt` Default: 1
LLT_STOP	Shutdown mode for LLT. By default, LLT is enabled to stop during a system shutdown. This environment variable is defined in the following file: `/etc/sysconfig/llt` Default: 1
GAB_START	Startup mode for GAB. By default, GAB is enabled to start up after a system reboot. This environment variable is defined in the following file: `/etc/sysconfig/gab` Default: 1
GAB_STOP	Shutdown mode for GAB. By default, GAB is enabled to stop during a system shutdown. This environment variable is defined in the following file: `/etc/sysconfig/gab` Default: 1
VXFEN_START	Startup mode for VxFEN. By default, VxFEN is enabled to start up after a system reboot. This environment variable is defined in the following file: `/etc/sysconfig/vxfen` Default: 1

Table 2-4 Start and stop environment variables for VCS *(continued)*

Environment variable	Definition and default value
VXFEN_STOP	Shutdown mode for VxFEN. By default, VxFEN is enabled to stop during a system shutdown. This environment variable is defined in the following file: `/etc/sysconfig/vxfen` Default: 1
VCS_START	Startup mode for VCS engine. By default, VCS engine is enabled to start up after a system reboot. This environment variable is defined in the following file: `/etc/sysconfig/vcs` Default: 1
VCS_STOP	Shutdown mode for VCS engine. By default, VCS engine is enabled to stop during a system shutdown. This environment variable is defined in the following file: `/etc/sysconfig/vcs` Default: 1

Table 2-4 Start and stop environment variables for VCS *(continued)*

Environment variable	Definition and default value
VCS_STOP_TIMEOUT	Time-out value in seconds for the stop operation of the VCS service. VCS uses this value during a system shutdown or restart operation to determine how long to wait for its stop operation to complete. After this duration has elapsed, VCS forcefully stops itself. If this value is set to 0 seconds, the stop operation does not time out. If an issue occurs when the resources are taken offline, HAD continues to be in the LEAVING state, thereby blocking the system shutdown or restart operation. Administrative intervention might be required to address such situations. Set this value to a positive integer to eliminate the need for manual intervention in case an operation is hung. After the duration specified that is in this variable has elapsed, VCS stops itself forcefully (`hastop -local -force`) and releases control of the application that is configured for HA. The operating system can then take the necessary action on the application components and continue with the shutdown or the restart operation. **Note:** If this value is set to anything other than a positive integer, VCS uses the default value (0, which indicates no time-out) instead. In `systemd` environments, the TimeoutStopSec attribute of the VCS service is set to 0 (infinity) to prevent the stop operation from timing out. This environment variable is defined in the `/etc/sysconfig/vcs` file. **Note:** Veritas recommends that you do not change the TimeoutStopSec attribute of the VCS service. If you want to configure a time-out value for the stop operation, use the VCS_STOP_TIMEOUT variable in the `/etc/sysconfig/vcs` file. **Warning:** Specifying a time-out value other than the default may have some adverse effects on the applications managed by VCS. For example, during a shutdown or a restart operation on a cluster node: Scenario 1, which may result in some unexpected behavior: If the value of VCS_STOP_TIMEOUT is too less, the VCS service stop operation times out before it can stop all the resources. This time-out may occur even when there is no issue with the cluster. Such an event may cause application-level issues in the cluster, because the application processes are no longer under the control of VCS. Scenario 2, which may result in some unexpected behavior: If a VCS agent fails to stop an application that it monitors, administrative intervention might be required. The VCS service stop operation times out and the necessary administrative intervention is not carried out. Default value: 0 seconds (indicates no time-out)

Adding and removing LLT links

In an SF Oracle RAC cluster, Oracle Clusterware heartbeat link must be configured as an LLT link. If Oracle Clusterware and LLT use different links for their communication, then the membership change between VCS and Oracle Clusterware is not coordinated correctly. For example, if only the Oracle Clusterware links are down, Oracle Clusterware kills one set of nodes after the expiry of the css-misscount interval and initiates the Oracle Clusterware and database recovery, even before CVM and CFS detect the node failures. This uncoordinated recovery may cause data corruption.

If you need additional capacity for Oracle communication on your private interconnects, you can add LLT links. The network IDs of the interfaces connected to the same physical network must match. The interfaces specified in the PrivNIC or MultiPrivNIC configuration must be exactly the same in name and total number as those which have been used for LLT configuration.

You can use the `lltconfig` command to add or remove LLT links when LLT is running.

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

See the `lltconfig`(1M) manual page for more details.

Note: When you add or remove LLT links, you need not shut down GAB or the high availability daemon, `had`. Your changes take effect immediately, but are lost on the next restart. For changes to persist, you must also update the `/etc/lltab` file.

To add LLT links

- Depending on the LLT link type, run the following command to add an LLT link:
  - For ether link type:

    ```
    # lltconfig -t devtag -d device
    [-b ether ] [-s SAP] [-m mtu]  [-I] [-Q]
    ```

  - For UDP link type:

    ```
    # lltconfig -t devtag -d device
    -b udp [-s port] [-m mtu]
    -I IPaddr -B bcast
    ```

  - For UDP6 link type:

```
# lltconfig -t devtag -d device
-b udp6 [-s port] [-m mtu]
-I IPaddr [-B mcast]
```

- For RDMA link type:

```
# lltconfig -t devtag -d device
 -b rdma -s port [-m mtu]
 -I IPaddr -B bcast
```

Where:

devtag	Tag to identify the link
device	Device name of the interface. For link type ether, you can specify the device name as an interface name. For example, eth0. Preferably, specify the device name as eth-macaddress. For example, eth- xx:xx:xx:xx:xx:xx. For link types udp and udp6, the device is the udp and udp6 device name respectively. For link type rdma, the device name is udp.
bcast	Broadcast address for the link type udp and rdma
mcast	Multicast address for the link type udp6
IPaddr	IP address for link types udp, udp6 and rdma
SAP	SAP to bind on the network links for link type ether
port	Port for link types udp, udp6 and rdma
mtu	Maximum transmission unit to send packets on network links

For example:

- For ether link type:

```
# lltconfig -t eth4 -d eth4 -s 0xcafe -m 1500
```

- For UDP link type:

```
# lltconfig -t link1 -d udp -b udp -s 50010
-I 192.168.1.1 -B 192.168.1.255
```

- For UDP6 link type:

```
# lltconfig -t link1 -d udp6 -b udp6 -s 50010 -I 2000::1
```

- For RDMA link:

```
# lltconfig -t link1 -d udp -b rdma -s 50010
 -I 192.168.1.1 -B 192.168.1.255
```

Note: If you want the addition of LLT links to be persistent after reboot, then you must edit the `/etc/lltab` with LLT entries.

To remove an LLT link

1 If the link you want to remove is configured as a PrivNIC or MultiPrivNIC resource, you need to modify the resource configuration before removing the link.

If you have configured the link under PrivNIC or MultiPrivNIC as a failover target in the case of link failure, modify the PrivNIC or MultiPrivNIC configuration as follows:

```
# haconf -makerw
# hares -modify resource_name Device link_name \
device_id [-sys hostname]
# haconf -dump -makero
```

For example, if the links eth1, eth2, and eth3 were configured as PrivNIC resources, and you want to remove eth3:

```
# haconf -makerw
# hares -modify ora_priv Device eth1
   0  eth2  1
```

where eth1 and eth2 are the links that you want to retain in your cluster.

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

2 Run the following command to disable a network link that is configured under LLT.

```
# lltconfig -t devtag -L disable
```

3 Wait for the 16 seconds (LLT peerinact time).

4 Run the following command to remove the link.

```
# lltconfig -u devtag
```

Configuring aggregated interfaces under LLT

If you want to configure LLT to use aggregated interfaces after installing and configuring VCS, you can use one of the following approaches:

- Edit the /etc/llttab file
  This approach requires you to stop LLT. The aggregated interface configuration is persistent across reboots.
- Run the `lltconfig` command
  This approach lets you configure aggregated interfaces on the fly. However, the changes are not persistent across reboots.

To configure aggregated interfaces under LLT by editing the /etc/llttab file

1. If LLT is running, stop LLT after you stop the other dependent modules.

   For RHEL 7, SLES 12, and supported RHEL distributions:

   ```
   # systemctl stop llt
   ```

   For earlier versions of RHEL, SLES, and supported RHEL distributions:

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

   See "Starting or stopping SF Oracle RAC on each node" on page 52.

2. Add the following entry to the /etc/llttab file to configure an aggregated interface.

   ```
   link tag device_name systemid_range link_type sap mtu_size
   ```

tag	Tag to identify the link
device_name	Device name of the aggregated interface.
systemid_range	Range of systems for which the command is valid. If the link command is valid for all systems, specify a dash (-).
link_type	The link type must be ether.
sap	SAP to bind on the network links. Default is 0xcafe.
mtu_size	Maximum transmission unit to send packets on network links

3. Restart LLT for the changes to take effect. Restart the other dependent modules that you stopped in step 1.

   For RHEL 7, SLES 12, and supported RHEL distributions:

   ```
   # systemctl start llt
   ```

   For earlier versions of RHEL, SLES, and supported RHEL distributions:

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

   See "Starting or stopping SF Oracle RAC on each node" on page 52.

To configure aggregated interfaces under LLT using the lltconfig command

- When LLT is running, use the following command to configure an aggregated interface:

  ```
  lltconfig -t devtag -d device
  [-b linktype ] [-s SAP] [-m mtu]
  ```

devtag	Tag to identify the link
device	Device name of the aggregated interface.
link_type	The link type must be ether.
sap	SAP to bind on the network links. Default is 0xcafe.
mtu_size	Maximum transmission unit to send packets on network links

  See the `lltconfig`(1M) manual page for more details.

  You need not reboot after you make this change. However, to make these changes persistent across reboot, you must update the /etc/lltab file.

  See "To configure aggregated interfaces under LLT by editing the /etc/lltab file" on page 85.

Displaying the cluster details and LLT version for LLT links

You can use the `lltdump` command to display the LLT version for a specific LLT link. You can also display the cluster ID and node ID details.

See the `lltdump`(1M) manual page for more details.

To display the cluster details and LLT version for LLT links

- Run the following command to display the details:

  ```
  # /opt/VRTSllt/lltdump -D -f link
  ```

  For example, if eth2 is connected to sys1, then the command displays a list of all cluster IDs and node IDs present on the network link eth2.

  ```
  # /opt/VRTSllt/lltdump -D -f eth2

  lltdump : Configuration:

  device : eth2

  sap : 0xcafe
  promisc sap : 0
  promisc mac : 0
  cidsnoop : 1
  === Listening for LLT packets ===
  cid nid vmaj vmin
  3456 1 5 0
  3456 3 5 0
  83 0 4 0
  27 1 3 7
  3456 2 5 0
  ```

Configuring destination-based load balancing for LLT

Destination-based load balancing for Low Latency Transport (LLT) is turned off by default. Veritas recommends destination-based load balancing when the cluster setup has more than two nodes and more active LLT ports.

See "About Low Latency Transport (LLT)" on page 21.

To configure destination-based load balancing for LLT

- Run the following command to configure destination-based load balancing:

  ```
  lltconfig -F linkburst:0
  ```

Enabling and disabling intelligent resource monitoring for agents manually

Review the following procedures to enable or disable intelligent resource monitoring manually. The intelligent resource monitoring feature is enabled by default. The

IMF resource type attribute determines whether an IMF-aware agent must perform intelligent resource monitoring.

See "About resource monitoring" on page 37.

To enable intelligent resource monitoring

1. Make the VCS configuration writable.

   ```
   # haconf -makerw
   ```

2. Run the following command to enable intelligent resource monitoring.

   - To enable intelligent monitoring of offline resources:

     ```
     # hatype -modify resource_type IMF -update Mode 1
     ```

   - To enable intelligent monitoring of online resources:

     ```
     # hatype -modify resource_type IMF -update Mode 2
     ```

   - To enable intelligent monitoring of both online and offline resources:

     ```
     # hatype -modify resource_type IMF -update Mode 3
     ```

3. If required, change the values of the MonitorFreq key and the RegisterRetryLimit key of the IMF attribute.

   See the *Cluster Server Bundled Agents Reference Guide* for agent-specific recommendations to set these attributes.

   Review the agent-specific recommendations in the attribute definition tables to set these attribute key values.

4. Save the VCS configuration.

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

5 Make sure that the AMF kernel driver is configured on all nodes in the cluster.

For RHEL 7, SLES 12, and supported RHEL distributions:

```
/opt/VRTSamf/bin/amf status
```

For earlier versions of RHEL, SLES, and supported RHEL distributions:

```
/etc/init.d/amf status
```

If the AMF kernel driver is configured, the output resembles:

```
AMF: Module loaded and configured
```

Configure the AMF driver if the command output returns that the AMF driver is not loaded or not configured.

See "Administering the AMF kernel driver" on page 90.

6 Restart the agent. Run the following commands on each node.

```
# haagent -stop agent_name -force -sys sys_name
# haagent -start agent_name -sys sys_name
```

To disable intelligent resource monitoring

1 Make the VCS configuration writable.

```
# haconf -makerw
```

2 To disable intelligent resource monitoring for all the resources of a certain type, run the following command:

```
# hatype -modify resource_type IMF -update Mode 0
```

3 To disable intelligent resource monitoring for a specific resource, run the following command:

```
# hares -override resource_name IMF
# hares -modify resource_name IMF -update Mode 0
```

4 Save the VCS configuration.

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

Note: VCS provides haimfconfig script to enable or disable the IMF functionality for agents. You can use the script with VCS in running or stopped state. Use the script to enable or disable IMF for the IMF-aware bundled agents, enterprise agents, and custom agents.

Administering the AMF kernel driver

Review the following procedures to start, stop, or unload the AMF kernel driver.

See "About the IMF notification module" on page 36.

See "Environment variables to start and stop VCS modules" on page 77.

To start the AMF kernel driver

1 Set the value of the AMF_START variable to 1 in the following file, if the value is not already 1:

```
# /etc/sysconfig/amf
```

2 Start the AMF kernel driver. Run the following command:

For RHEL 7, SLES 12, and supported RHEL distributions:

```
# systemctl start amf
```

For earlier versions of RHEL, SLES, and supported RHEL distributions:

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

To stop the AMF kernel driver

1 Set the value of the AMF_STOP variable to 1 in the following file, if the value is not already 1:

```
# /etc/sysconfig/amf
```

2 Stop the AMF kernel driver. Run the following command:

For RHEL 7, SLES 12, and supported RHEL distributions:

```
# systemctl stop amf
```

For earlier versions of RHEL, SLES, and supported RHEL distributions:

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

To unload the AMF kernel driver

1 If agent downtime is not a concern, use the following steps to unload the AMF kernel driver:

- Stop the agents that are registered with the AMF kernel driver.
  The `amfstat` command output lists the agents that are registered with AMF under the Registered Reapers section.
  See the `amfstat` manual page.
- Stop the AMF kernel driver.
  See "To stop the AMF kernel driver" on page 90.
- Start the agents.

2 If you want minimum downtime of the agents, use the following steps to unload the AMF kernel driver:

- Run the following command to disable the AMF driver even if agents are still registered with it.

```
# amfconfig -Uof
```

- Stop the AMF kernel driver.
  See "To stop the AMF kernel driver" on page 90.

Administering I/O fencing

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

- About administering I/O fencing
  See "About administering I/O fencing" on page 92.
- About vxfentsthdw utility
  See "About the vxfentsthdw utility" on page 92.
- About vxfenadm utility
  See "About the vxfenadm utility" on page 100.
- About vxfenclearpre utility
  See "About the vxfenclearpre utility" on page 105.
- About vxfenswap utility
  See "About the vxfenswap utility" on page 109.

Note: I/O fencing no longer supports the `raw` disk policy for fencing. Only `dmp` disk policy is supported.

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

See “Troubleshooting I/O fencing” on page 187.

See “About administering I/O fencing” on page 92.

About administering I/O fencing

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

vxfentsthdw	Tests SCSI-3 functionality of the disks for I/O fencing See “About the vxfentsthdw utility” on page 92.
vxfenconfig	Configures and unconfigures I/O fencing Lists the coordination points used by the vxfen driver.
vxfenadm	Displays information on I/O fencing operations and manages SCSI-3 disk registrations and reservations for I/O fencing See “About the vxfenadm utility” on page 100.
vxfenclearpre	Removes SCSI-3 registrations and reservations from disks See “About the vxfenclearpre utility” on page 105.
vxfenswap	Replaces coordination points without stopping I/O fencing See “About the vxfenswap utility” on page 109.
vxfendisk	Generates the list of paths of disks in the disk group. This utility requires that Veritas Volume Manager (VxVM) is installed and configured.

The I/O fencing commands reside in the /opt/VRTS/bin folder. Make sure you added this folder path to the PATH environment variable.

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

About the vxfentsthdw utility

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 disk groups.

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

The utility, which you can run from one system in the cluster, tests the storage used for data by setting and verifying SCSI-3 registrations on the disk or disks you specify, setting and verifying persistent reservations on the disks, writing data to the disks and reading it, and removing the registrations from the disks.

Refer also to the `vxfentsthdw`(1M) manual page.

General guidelines for using the vxfentsthdw utility

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

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

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

- The two nodes must have SSH (default) or rsh communication. If you use rsh, launch the `vxfentsthdw` utility with the `-n` option.
  After completing the testing process, you can remove permissions for communication and restore public network connections.
- To ensure both systems are connected to the same disk during the testing, you can use the `vxfenadm -i` *`diskpath`* command to verify a disk's serial number.
  See "Verifying that 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.
- The utility indicates a disk can be used for I/O fencing with a message resembling:

  ```
  The disk /dev/sdx is ready to be configured for
  I/O Fencing on node sys1
  ```

  If the utility does not show a message stating a disk is ready, verification has failed.
- The `-o` option overrides disk size-related errors and the utility proceeds with other tests, however, the disk may not setup correctly as the size may be smaller than the supported size. The supported disk size for data disks is 256 MB and for coordinator disks is 128 MB.
- If the disk you intend to test has existing SCSI-3 registration keys, the test issues a warning before proceeding.

About the vxfentsthdw command options

Table 2-5 describes the methods that the utility provides to test storage devices.

Table 2-5 vxfentsthdw options

vxfentsthdw option	Description	When to use
-n	Utility uses rsh for communication.	Use when rsh is used for communication.
-r	Non-destructive testing. Testing of the disks for SCSI-3 persistent reservations occurs in a non-destructive way; that is, there is only testing for reads, not writes. Can be used with `-m`, `-f`, or `-g` options.	Use during non-destructive testing. See "Performing non-destructive testing on the disks using the -r option" on page 97.
-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 Dynamic Multi-Pathing (DMP) devices. Can be used with `-c` or `-g` options.	By default, the `vxfentsthdw` script picks up the DMP paths for disks in the disk group. If you want the script to use the raw paths for disks in the disk group, use the `-w` option.
-w	Use raw devices. Can be used with `-c` or `-g` options.	With the `-w` option, the `vxfentsthdw` script picks the operating system paths for disks in the disk group. By default, the script uses the `-d` option to pick up the DMP paths for disks in the disk group.
-c	Utility tests the coordinator disk group prompting for systems and devices, and reporting success or failure.	For testing disks in coordinator disk group. See "Testing the coordinator disk group using the -c option of vxfentsthdw" on page 95.

Table 2-5 vxfentsthdw options *(continued)*

vxfentsthdw option	Description	When to use
-m	Utility runs manually, in interactive mode, prompting for systems and devices, and reporting success or failure. Can be used with `-r` and `-t` options. `-m` is the default option.	For testing a few disks or for sampling disks in larger arrays. See "Testing the shared disks using the vxfentsthdw -m option" on page 97.
-f *filename*	Utility tests system and device combinations listed in a text file. Can be used with `-r` and `-t` options.	For testing several disks. See "Testing the shared disks listed in a file using the vxfentsthdw -f option" on page 99.
-g *disk_group*	Utility tests all disk devices in a specified disk group. Can be used with `-r` and `-t` options.	For testing many disks and arrays of disks. Disk groups may be temporarily created for testing purposes and destroyed (ungrouped) after testing. See "Testing all the disks in a disk group using the vxfentsthdw -g option" on page 99.
-o	Utility overrrides disk size-related errors.	For testing SCSI-3 Reservation compliance of disks, but, overrides disk size-related errors.

Testing the coordinator disk group using the -c option of vxfentsthdw

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.

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

- From the node sys1, the disks are seen as `/dev/sdg`, `/dev/sdh`, and `/dev/sdi`.
- From the node sys2, the same disks are seen as `/dev/sdx`, `/dev/sdy`, and `/dev/sdz`.

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:

```
# vxfentsthdw -c vxfencoorddg
```

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

```
Enter the first node of the cluster: sys1
Enter the second node of the cluster: sys2
```

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/sdg have PASSED.
The disk is now ready to be configured for I/O Fencing on node
sys1 as a COORDINATOR DISK.

ALL tests on the disk /dev/sdx have PASSED.
The disk is now ready to be configured for I/O Fencing on node
sys2 as a COORDINATOR DISK.
```

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

Removing and replacing a failed disk

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

To remove and replace a failed disk

1 Use the vxdiskadm utility to remove the failed disk from the disk group.

Refer to the *Storage Foundation Administrator's Guide*.

2 Add a new disk to the node, initialize it, and add it to the coordinator disk group.

See the *Storage Foundation for Oracle RAC Configuration and Upgrade Guide* for instructions to initialize disks for I/O fencing and to set up coordinator disk groups.

If necessary, start the disk group.

See the *Storage Foundation Administrator's Guide* for instructions to start the disk group.

3 Retest the disk group.

See "Testing the coordinator disk group using the -c option of vxfentsthdw" on page 95.

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

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

To perform non-destructive testing on disks

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

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

  ```
  # vxfentsthdw -rm
  ```

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

Testing the shared disks using the vxfentsthdw -m option

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

This procedure uses the /dev/sdx disk in the steps.

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

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

```
The disk /dev/sdx is ready to be configured for
I/O Fencing on node sys1
```

Note: For A/P arrays, run the `vxfentsthdw` command only on active enabled paths.

To test disks using the vxfentsthdw script

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

   ```
   # vxfentsthdw [-n]
   ```

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

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

   Do you still want to continue : [y/n] (default: n) y
   Enter the first node of the cluster: sys1
   Enter the second node of the cluster: sys2
   ```

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

   ```
   Enter the disk name to be checked for SCSI-3 PGR on node
   sys1 in the format:
           for dmp: /dev/vx/rdmp/sdx
           for raw: /dev/sdx
   Make sure it's the same disk as seen by nodes sys1 and sys2
   /dev/sdr

   Enter the disk name to be checked for SCSI-3 PGR on node
   sys2 in the format:
           for dmp: /dev/vx/rdmp/sdx
           for raw: /dev/sdx
   Make sure it's the same disk as seen by nodes sys1 and sys2
   /dev/sdr
   ```

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

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

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

```
ALL tests on the disk /dev/sdx have PASSED
The disk is now ready to be configured for I/O Fencing on node
sys1
...
Removing test keys and temporary files, if any ...
.
.
```

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

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

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

To test the shared disks listed in a file

1 Create a text file `disks_test` to test two disks shared by systems `sys1` and `sys2` that might resemble:

```
sys1 /dev/sdz sys2 /dev/sdy
sys1 /dev/sdu sys2 /dev/sdw
```

where the first disk is listed in the first line and is seen by `sys1` as `/dev/sdz` and by `sys2` as `/dev/sdy`. The other disk, in the second line, is seen as `/dev/sdu` from `sys1` and `/dev/sdw` from `sys2`. Typically, the list of disks could be extensive.

2 To test the disks, enter the following command:

```
# vxfentsthdw -f disks_test
```

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

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

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

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

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

To test all the disks in a disk group

1. Create a disk group for the disks that you want to test.
2. Enter the following command to test the disk group test_disks_dg:

   ```
   # vxfentsthdw -g test_disks_dg
   ```

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

Testing a disk with existing keys

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

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

         ******** WARNING!!!!!!!! ********

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

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

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

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

About the vxfenadm utility

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

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

-s	read the keys on a disk and display the keys in numeric, character, and node format **Note:** The `-g` and `-G` options are deprecated. Use the `-s` option.
-i	read SCSI inquiry information from device
-m	register with disks
-n	make a reservation with disks
-p	remove registrations made by other systems
-r	read reservations
-x	remove registrations

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

About the I/O fencing registration key format

The keys that the vxfen driver registers on the data disks and the coordinator disks consist of eight bytes. The key format is different for the coordinator disks and data disks.

The key format of the coordinator disks is as follows:

Byte	0	1	2	3	4	5	6	7
Value	V	F	cID 0x	cID 0x	cID 0x	cID 0x	nID 0x	nID 0x

where:

- VF is the unique identifier that carves out a namespace for the keys (consumes two bytes)
- cID 0x is the LLT cluster ID in hexadecimal (consumes four bytes)
- nID 0x is the LLT node ID in hexadecimal (consumes two bytes)

The vxfen driver uses this key format in both sybase mode of I/O fencing.

The key format of the data disks that are configured as failover disk groups under VCS is as follows:

Byte	0	1	2	3	4	5	6	7
Value	A+nID	V	C	S				

where nID is the LLT node ID

For example: If the node ID is 1, then the first byte has the value as B ('A' + 1 = B).

The key format of the data disks configured as parallel disk groups under Cluster Volume Manager (CVM) is as follows:

Byte	0	1	2	3	4	5	6	7
Value	A+nID	P	G	R	DGcount	DGcount	DGcount	DGcount

where DGcount is the count of disk groups in the configuration (consumes four bytes).

By default, CVM uses a unique fencing key for each disk group. However, some arrays have a restriction on the total number of unique keys that can be registered. In such cases, you can use the `same_key_for_alldgs` tunable parameter to change the default behavior. The default value of the parameter is `off`. If your configuration hits the storage array limit on total number of unique keys, you can change the value to `on` using the `vxdefault` command as follows:

```
# vxdefault set same_key_for_alldgs on
# vxdefault list
KEYWORD               CURRENT-VALUE   DEFAULT-VALUE
...
same_key_for_alldgs   on              off
...
```

If the tunable is changed to `on`, all subsequent keys that the CVM generates on disk group imports or creates have '0000' as their last four bytes (DGcount is 0). You must deport and re-import all the disk groups that are already imported for the changed value of the `same_key_for_alldgs` tunable to take effect.

Displaying the I/O fencing registration keys

You can display the keys that are currently assigned to the disks using the `vxfenadm` command.

The variables such as *disk_7*, *disk_8*, and *disk_9* in the following procedure represent the disk names in your setup.

To display the I/O fencing registration keys

1 To display the key for the disks, run the following command:

```
# vxfenadm -s disk_name
```

For example:

- To display the key for the coordinator disk `/dev/sdx` from the system with node ID 1, enter the following command:

```
# vxfenadm -s /dev/sdx
key[1]:
  [Numeric Format]:  86,70,68,69,69,68,48,48
  [Character Format]: VFDEED00
* [Node Format]: Cluster ID: 57069 Node ID: 0 Node Name: sys1
```

  The `-s` option of `vxfenadm` displays all eight bytes of a key value in three formats. In the numeric format,

  - The first two bytes, represent the identifier VF, contains the ASCII value 86, 70.
  - The next four bytes contain the ASCII value of the cluster ID 57069 encoded in hex (0xDEED) which are 68, 69, 69, 68.
  - The remaining bytes contain the ASCII value of the node ID 0 (0x00) which are 48, 48. Node ID 1 would be 01 and node ID 10 would be 0A.

  An asterisk before the Node Format indicates that the `vxfenadm` command is run from the node of a cluster where LLT is configured and is running.

- To display the keys on a CVM parallel disk group:

```
# vxfenadm -s /dev/vx/rdmp/disk_7

Reading SCSI Registration Keys...

Device Name: /dev/vx/rdmp/disk_7
Total Number Of Keys: 1
key[0]:
  [Numeric Format]:  66,80,71,82,48,48,48,49
  [Character Format]: BPGR0001
  [Node Format]: Cluster ID: unknown  Node ID: 1 Node Name: sys2
```

- To display the keys on a Cluster Server (VCS) failover disk group:

```
# vxfenadm -s /dev/vx/rdmp/disk_8

Reading SCSI Registration Keys...

Device Name: /dev/vx/rdmp/disk_8
Total Number Of Keys: 1
key[0]:
  [Numeric Format]:  65,86,67,83,0,0,0,0
```

```
[Character Format]: AVCS
[Node Format]: Cluster ID: unknown  Node ID: 0 Node Name: sys1
```

2 To display the keys that are registered in all the disks specified in a disk file:

```
# vxfenadm -s all -f disk_filename
```

For example:

To display all the keys on coordinator disks:

```
# vxfenadm -s all -f /etc/vxfentab

Device Name: /dev/vx/rdmp/disk_9
Total Number Of Keys: 2
key[0]:
[Numeric Format]:  86,70,70,68,57,52,48,49
[Character Format]: VFFD9401
* [Node Format]: Cluster ID: 64916 Node ID: 1   Node Name: sys2
key[1]:
[Numeric Format]:  86,70,70,68,57,52,48,48
[Character Format]: VFFD9400
* [Node Format]: Cluster ID: 64916 Node ID: 0   Node Name: sys1
```

You can verify the cluster ID using the `lltstat -C` command, and the node ID using the `lltstat -N` command. For example:

```
# lltstat -C
57069
```

If the disk has keys that do not belong to a specific cluster, then the `vxfenadm` command cannot look up the node name for the node ID, and hence prints the node name as unknown. For example:

```
Device Name: /dev/vx/rdmp/disk_7
Total Number Of Keys: 1
key[0]:
  [Numeric Format]:  86,70,45,45,45,45,48,49
  [Character Format]: VF----01
  [Node Format]: Cluster ID: unknown  Node ID: 1  Node Name: sys2
```

For disks with arbitrary format of keys, the `vxfenadm` command prints all the fields as unknown. For example:

```
  [Numeric Format]:  65,66,67,68,49,50,51,45
  [Character Format]: ABCD123-
```

```
[Node Format]: Cluster ID: unknown Node ID: unknown
       Node Name: unknown
```

Verifying that the nodes see the same disk

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

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

To verify that the nodes see the same disks

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

```
# vxfenadm -i /dev/sdr

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

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

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

```
# vxfenadm -i /dev/sdt

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

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

About the vxfenclearpre utility

You can use the vxfenclearpre utility to remove SCSI-3 registrations and reservations on the disks as well as coordination point servers.

See "Removing preexisting keys" on page 106.

This utility now supports server-based fencing. You can use the vxfenclearpre utility to clear registrations from coordination point servers (CP servers) for the current cluster. The local node from where you run the utility must have the UUID of the current cluster at the /etc/vx/.uuids directory in the clusuuid file. If the UUID file for that cluster is not available on the local node, the utility does not clear registrations from the coordination point servers.

Note: You can use the utility to remove the registration keys and the registrations (reservations) from the set of coordinator disks for any cluster you specify in the command, but you can only clear registrations of your current cluster from the CP servers. Also, you may experience delays while clearing registrations on the coordination point servers because the utility tries to establish a network connection with the IP addresses used by the coordination point servers. The delay may occur because of a network issue or if the IP address is not reachable or is incorrect.

For any issues you encounter with the vxfenclearpre utility, you can you can refer to the log file at, `/var/VRTSvcs/log/vxfen/vxfen.log` file.

See "Issues during fencing startup on SF Oracle RAC cluster nodes set up for server-based fencing" on page 199.

Removing preexisting keys

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

You can also use this procedure to remove the registration and reservation keys of another node or other nodes on shared disks or CP server.

To clear keys after split-brain

1 Stop VCS on all nodes.

```
# hastop -all
```

2 Make sure that the port h is closed on all the nodes. Run the following command on each node to verify that the port h is closed:

```
# gabconfig -a
```

Port h must not appear in the output.

3 Stop I/O fencing on all nodes. Enter the following command on each node:

For RHEL 7, SLES 12, and supported RHEL distributions:

```
# systemctl stop vxfen
```

For earlier versions of RHEL, SLES, and supported RHEL distributions:

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

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

5 Start the vxfenclearpre script:

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

6 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
```

In some cases, 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. You can ignore these informational messages.

```
<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
```

The script cleans up the disks and displays the following status messages.

```
Cleaning up the coordinator disks...

Cleared keys from n out of n disks,
where n is the total number of disks.

Successfully removed SCSI-3 persistent registrations
from the coordinator disks.

Cleaning up the Coordination Point Servers...

...................
[10.209.80.194]:50001: Cleared all registrations
[10.209.75.118]:443: Cleared all registrations

Successfully removed registrations from the Coordination Point Servers.

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

Successfully removed SCSI-3 persistent registration and
reservations from the shared data disks.

See the log file /var/VRTSvcs/log/vxfen/vxfen.log
```

```
You can retry starting fencing module. In order to restart the whole
product, you might want to reboot the system.
```

7 Start the fencing module on all the nodes.

For RHEL 7, SLES 12, and supported RHEL distributions:

```
# systemctl start vxfen
```

For earlier versions of RHEL, SLES, and supported RHEL distributions:

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

8 Start VCS on all nodes.

```
# hastart
```

About the vxfenswap utility

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

This utility supports both disk-based and server-based fencing.

The utility uses SSH, RSH, or hacli for communication between nodes in the cluster. Before you execute the utility, ensure that communication between nodes is set up in one these communication protocols.

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

See the *Storage Foundation for Oracle RAC Configuration and Upgrade Guide* for details on the I/O fencing requirements.

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

- The disk becomes defective or inoperable and you want to switch to a new disk group.
  See "Replacing I/O fencing coordinator disks when the cluster is online" on page 110.
  See "Replacing the coordinator disk group in a cluster that is online" on page 114.
  If you want to replace the coordinator disks when the cluster is offline, you cannot use the vxfenswap utility. You must manually perform the steps that the utility does to replace the coordinator disks.
  See "Replacing defective disks when the cluster is offline" on page 193.
- The keys that are registered on the coordinator disks are lost.

In such a case, the cluster might panic when a network partition occurs. You can replace the coordinator disks with the same disks using the vxfenswap command. During the disk replacement, the missing keys register again without any risk of data corruption.

See "Refreshing lost keys on coordinator disks" on page 120.

In server-based fencing configuration, you can use the vxfenswap utility to perform the following tasks:

- Perform a planned replacement of customized coordination points (CP servers or SCSI-3 disks).

  See "Replacing coordination points for server-based fencing in an online cluster" on page 133.

- Refresh the I/O fencing keys that are registered on the coordination points.

  See "Refreshing registration keys on the coordination points for server-based fencing" on page 131.

You can also use the vxfenswap utility to migrate between the disk-based and the server-based fencing without incurring application downtime in the SF Oracle RAC cluster.

If the vxfenswap operation is unsuccessful, then you can use the `-a cancel` of the `vxfenswap` command to manually roll back the changes that the vxfenswap utility does.

- For disk-based fencing, use the `vxfenswap -g diskgroup -a cancel` command to cancel the vxfenswap operation.

  You must run this command if a node fails during the process of disk replacement, or if you aborted the disk replacement.

- For server-based fencing, use the `vxfenswap -a cancel` command to cancel the vxfenswap operation.

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

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

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

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

1. Make sure system-to-system communication is functioning properly.
2. Determine the value of the FaultTolerance attribute.

   ```
   # hares -display coordpoint -attribute FaultTolerance -localclus
   ```

3. Estimate the number of coordination points you plan to use as part of the fencing configuration.
4. Set the value of the FaultTolerance attribute to 0.

   Note: It is necessary to set the value to 0 because later in the procedure you need to reset the value of this attribute to a value that is lower than the number of coordination points. This ensures that the Coordpoint Agent does not fault.

5. Check the existing value of the LevelTwoMonitorFreq attribute.

   ```
   #hares -display coordpoint -attribute LevelTwoMonitorFreq -localclus
   ```

   Note: Make a note of the attribute value before you proceed to the next step. After migration, when you re-enable the attribute you want to set it to the same value.

   You can also run the `hares -display coordpoint` to find out whether the LevelTwoMonitorFreq value is set.

6. Disable level two monitoring of CoordPoint agent.

   ```
   # hares -modify coordpoint LevelTwoMonitorFreq 0
   ```

7 Make sure that the cluster is online.

```
# vxfenadm -d

I/O Fencing Cluster Information:
================================
Fencing Protocol Version: 201
Fencing Mode: SCSI3
Fencing SCSI3 Disk Policy: dmp
Cluster Members:
  * 0 (sys1)
  1 (sys2)
RFSM State Information:
  node 0 in state 8 (running)
  node 1 in state 8 (running)
```

8 Import the coordinator disk group.

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

```
# vxdg -tfC import `cat /etc/vxfendg`
```

where:

-t specifies that the disk group is imported only until the node restarts.

-f specifies that the import is to be done forcibly, which is necessary if one or more disks is not accessible.

-C specifies that any import locks are removed.

9 If your setup uses `VRTSvxvm` *version*, then skip to step 10. You need not set `coordinator=off` to add or remove disks. For other VxVM versions, perform this step:

Where *`version`* is the specific release version.

Turn off the coordinator attribute value for the coordinator disk group.

```
# vxdg -g vxfencoorddg set coordinator=off
```

10 To remove disks from the coordinator disk group, use the VxVM disk administrator utility `vxdiskadm`.

11 Perform the following steps to add new disks to the coordinator disk group:

- Add new disks to the node.
- Initialize the new disks as VxVM disks.

- Check the disks for I/O fencing compliance.
- Add the new disks to the coordinator disk group and set the coordinator attribute value as "on" for the coordinator disk group.

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

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

12 From one node, start the vxfenswap utility. You must specify the disk group to the utility.

The utility performs the following tasks:

- Backs up the existing `/etc/vxfentab` file.
- Creates a test file `/etc/vxfentab.test` for the disk group that is modified on each node.
- Reads the disk group you specified in the vxfenswap command and adds the disk group to the `/etc/vxfentab.test` file on each node.
- Verifies that the serial number of the new disks are identical on all the nodes. The script terminates if the check fails.
- Verifies that the new disks can support I/O fencing on each node.

13 If the disk verification passes, the utility reports success and asks if you want to commit the new set of coordinator disks.

14 Confirm whether you want to clear the keys on the coordination points and proceed with the vxfenswap operation.

```
Do you want to clear the keys on the coordination points
and proceed with the vxfenswap operation? [y/n] (default: n) y
```

15 Review the message that the utility displays and confirm that you want to commit the new set of coordinator disks. Else skip to step 16.

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

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

16 If you do not want to commit the new set of coordinator disks, answer n.

The vxfenswap utility rolls back the disk replacement operation.

17 If coordinator flag was set to off in step 9, then set it on.

```
# vxdg -g vxfencoorddg set coordinator=on
```

18 Deport the diskgroup.

```
# vxdg deport vxfencoorddg
```

19 Re-enable the LevelTwoMonitorFreq attribute of the CoordPoint agent.You may want to use the value that was set before disabling the attribute.

```
# hares -modify coordpoint LevelTwoMonitorFreq Frequencyvalue
```

where *Frequencyvalue* is the value of the attribute.

20 Set the FaultTolerance attribute to a value that is lower than 50% of the total number of coordination points.

For example, if there are four (4) coordination points in your configuration, then the attribute value must be lower than two (2).If you set it to a higher value than two (2) the CoordPoint agent faults.

Replacing the coordinator disk group in a cluster that is online

You can also replace the coordinator disk group using the vxfenswap utility. The following example replaces the coordinator disk group vxfencoorddg with a new disk group vxfendg.

To replace the coordinator disk group

1 Make sure system-to-system communication is functioning properly.

2 Determine the value of the FaultTolerance attribute.

```
# hares -display coordpoint -attribute FaultTolerance -localclus
```

3 Estimate the number of coordination points you plan to use as part of the fencing configuration.

4 Set the value of the FaultTolerance attribute to 0.

Note: It is necessary to set the value to 0 because later in the procedure you need to reset the value of this attribute to a value that is lower than the number of coordination points. This ensures that the Coordpoint Agent does not fault.

---

5 Check the existing value of the LevelTwoMonitorFreq attribute.

```
# hares -display coordpoint -attribute LevelTwoMonitorFreq -localclus
```

Note: Make a note of the attribute value before you proceed to the next step. After migration, when you re-enable the attribute you want to set it to the same value.

6 Disable level two monitoring of CoordPoint agent.

```
# haconf -makerw

# hares -modify coordpoint LevelTwoMonitorFreq 0

# haconf -dump -makero
```

7 Make sure that the cluster is online.

```
# vxfenadm -d


I/O Fencing Cluster Information:
================================
Fencing Protocol Version: 201
Fencing Mode: SCSI3
Fencing SCSI3 Disk Policy: dmp
Cluster Members:
  * 0 (sys1)
  1 (sys2)
RFSM State Information:
  node 0 in state 8 (running)
  node 1 in state 8 (running)
```

8 Find the name of the current coordinator disk group (typically vxfencoorddg) that is in the `/etc/vxfendg` file.

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

9 Find the alternative disk groups available to replace the current coordinator disk group.

```
# vxdisk -o alldgs list

DEVICE    TYPE     DISK  GROUP     STATUS
sda     auto:cdsdisk    -  (vxfendg)     online
sdb     auto:cdsdisk    -  (vxfendg)     online
sdc     auto:cdsdisk    -  (vxfendg)     online
sdx     auto:cdsdisk    -  (vxfencoorddg)     online
sdy     auto:cdsdisk    -  (vxfencoorddg)     online
sdz     auto:cdsdisk    -  (vxfencoorddg)     online
```

10 Validate the new disk group for I/O fencing compliance. Run the following command:

```
# vxfentsthdw -c vxfendg
```

See "Testing the coordinator disk group using the -c option of vxfentsthdw" on page 95.

11 If the new disk group is not already deported, run the following command to deport the disk group:

```
# vxdg deport vxfendg
```

12 Perform one of the following:

- Create the `/etc/vxfenmode.test` file with new fencing mode and disk policy information.
- Edit the existing the `/etc/vxfenmode` with new fencing mode and disk policy information and remove any preexisting `/etc/vxfenmode.test` file.

Note that the format of the /etc/vxfenmode.test file and the /etc/vxfenmode file is the same.

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

13 From any node, start the vxfenswap utility. For example, if vxfendg is the new disk group that you want to use as the coordinator disk group:

```
# vxfenswap -g vxfendg [-n]
```

The utility performs the following tasks:

- Backs up the existing `/etc/vxfentab` file.

- Creates a test file `/etc/vxfentab.test` for the disk group that is modified on each node.
- Reads the disk group you specified in the vxfenswap command and adds the disk group to the `/etc/vxfentab.test` file on each node.
- Verifies that the serial number of the new disks are identical on all the nodes. The script terminates if the check fails.
- Verifies that the new disk group can support I/O fencing on each node.

14 If the disk verification passes, the utility reports success and asks if you want to replace the coordinator disk group.

15 Confirm whether you want to clear the keys on the coordination points and proceed with the vxfenswap operation.

```
Do you want to clear the keys on the coordination points
and proceed with the vxfenswap operation? [y/n] (default: n) y
```

16 Review the message that the utility displays and confirm that you want to replace the coordinator disk group. Else skip to step 21.

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

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

The utility also updates the `/etc/vxfendg` file with this new disk group.

17 Import the new disk group if it is not already imported before you set the coordinator flag "on".

```
# vxdg -t import vxfendg
```

18 Set the coordinator attribute value as "on" for the new coordinator disk group.

```
# vxdg -g vxfendg set  coordinator=on
```

Set the coordinator attribute value as "off" for the old disk group.

```
# vxdg -g vxfencoorddg set  coordinator=off
```

19 Deport the new disk group.

```
#  vxdg deport vxfendg
```

20 Verify that the coordinator disk group has changed.

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

The swap operation for the coordinator disk group is complete now.

21 If you do not want to replace the coordinator disk group, answer n at the prompt.

The vxfenswap utility rolls back any changes to the coordinator disk group.

22 Re-enable the LevelTwoMonitorFreq attribute of the CoordPoint agent. You may want to use the value that was set before disabling the attribute.

```
# haconf -makerw

# hares -modify coordpoint LevelTwoMonitorFreq Frequencyvalue

# haconf -dump -makero
```

where *Frequencyvalue* is the value of the attribute.

23 Set the FaultTolerance attribute to a value that is lower than 50% of the total number of coordination points.

For example, if there are four (4) coordination points in your configuration, then the attribute value must be lower than two (2).If you set it to a higher value than two (2) the CoordPoint agent faults.

Adding disks from a recovered site to the coordinator disk group

In a campus cluster environment, consider a case where the primary site goes down and the secondary site comes online with a limited set of disks. When the primary site restores, the primary site's disks are also available to act as coordinator disks. You can use the vxfenswap utility to add these disks to the coordinator disk group.

To add new disks from a recovered site to the coordinator disk group

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

```
# vxfenadm -d


I/O Fencing Cluster Information:
================================
Fencing Protocol Version: 201
Fencing Mode: SCSI3
Fencing SCSI3 Disk Policy: dmp
Cluster Members:
  * 0 (sys1)
  1 (sys2)
RFSM State Information:
  node 0 in state 8 (running)
  node 1 in state 8 (running)
```

3. Verify the name of the coordinator disk group.

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

4. Run the following command:

```
# vxdisk -o alldgs list

DEVICE    TYPE    DISK  GROUP    STATUS
sdx    auto:cdsdisk    -  (vxfencoorddg)    online
sdy    auto    -  -    offline
sdz    auto    -  -    offline
```

5. Verify the number of disks used in the coordinator disk group.

```
# vxfenconfig -l
I/O Fencing Configuration Information:
======================================
Count                : 1
Disk List
Disk Name        Major  Minor  Serial Number     Policy

/dev/vx/rdmp/sdx       32  48  R450 00013154 0312     dmp
```

6 When the primary site comes online, start the vxfenswap utility on any node in the cluster:

```
# vxfenswap -g vxfencoorddg [-n]
```

7 Verify the count of the coordinator disks.

```
# vxfenconfig -l
I/O Fencing Configuration Information:
======================================
Single Disk Flag      : 0
Count                 : 3
Disk List
Disk Name         Major  Minor  Serial Number     Policy

/dev/vx/rdmp/sdx       32  48  R450 00013154 0312     dmp
/dev/vx/rdmp/sdy       32  32  R450 00013154 0313     dmp
/dev/vx/rdmp/sdz       32  16  R450 00013154 0314     dmp
```

Refreshing lost keys on coordinator disks

If the coordinator disks lose the keys that are registered, the cluster might panic when a network partition occurs.

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

To refresh lost keys on coordinator disks

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

```
# vxfenadm -d


I/O Fencing Cluster Information:
================================
Fencing Protocol Version: 201
Fencing Mode: SCSI3
Fencing SCSI3 Disk Policy: dmp
Cluster Members:
  * 0 (sys1)
  1 (sys2)
RFSM State Information:
  node 0 in state 8 (running)
  node 1 in state 8 (running)
```

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

```
# vxfenadm -s all -f /etc/vxfentab


Device Name: /dev/vx/rdmp/sdx
Total Number of Keys: 0
No keys...
...
```

4. Copy the `/etc/vxfenmode` file to the `/etc/vxfenmode.test` file.

   This ensures that the configuration details of both the files are the same.

5 On any node, run the following command to start the vxfenswap utility:

```
# vxfenswap -g vxfencoorddg [-n]
```

6 Verify that the keys are atomically placed on the coordinator disks.

```
# vxfenadm -s all -f /etc/vxfentab


Device Name: /dev/vx/rdmp/sdx
Total Number of Keys: 4
...
```

Enabling or disabling the preferred fencing policy

You can enable or disable the preferred fencing feature for your I/O fencing configuration.

You can enable preferred fencing to use system-based race policy, group-based race policy, or site-based policy. If you disable preferred fencing, the I/O fencing configuration uses the default count-based race policy.

Preferred fencing is not applicable to majority-based I/O fencing.

To enable preferred fencing for the I/O fencing configuration

1 Make sure that the cluster is running with I/O fencing set up.

```
# vxfenadm -d
```

2 Make sure that the cluster-level attribute UseFence has the value set to SCSI3.

```
# haclus -value UseFence
```

3 To enable system-based race policy, perform the following steps:

- Make the VCS configuration writable.

  ```
  # haconf -makerw
  ```

- Set the value of the cluster-level attribute PreferredFencingPolicy as System.

  ```
  # haclus -modify PreferredFencingPolicy System
  ```

- Set the value of the system-level attribute FencingWeight for each node in the cluster.

For example, in a two-node cluster, where you want to assign sys1 five times more weight compared to sys2, run the following commands:

```
# hasys -modify sys1 FencingWeight 50
# hasys -modify sys2 FencingWeight 10
```

- Save the VCS configuration.

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

- Verify fencing node weights using:

```
# vxfenconfig -a
```

4 To enable group-based race policy, perform the following steps:

- Make the VCS configuration writable.

```
# haconf -makerw
```

- Set the value of the cluster-level attribute PreferredFencingPolicy as Group.

```
# haclus -modify PreferredFencingPolicy Group
```

- Set the value of the group-level attribute Priority for each service group. For example, run the following command:

```
# hagrp -modify service_group Priority 1
```

Make sure that you assign a parent service group an equal or lower priority than its child service group. In case the parent and the child service groups are hosted in different subclusters, then the subcluster that hosts the child service group gets higher preference.

- Save the VCS configuration.

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

5 To enable site-based race policy, perform the following steps:

- Make the VCS configuration writable.

```
# haconf -makerw
```

- Set the value of the cluster-level attribute PreferredFencingPolicy as Site.

```
# haclus -modify PreferredFencingPolicy Site
```

- Set the value of the site-level attribute Preference for each site.

  ```
  For example,
  # hasite -modify Pune Preference 2
  ```

- Save the VCS configuration.

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

6 To view the fencing node weights that are currently set in the fencing driver, run the following command:

```
# vxfenconfig -a
```

To disable preferred fencing for the I/O fencing configuration

1 Make sure that the cluster is running with I/O fencing set up.

```
# vxfenadm -d
```

2 Make sure that the cluster-level attribute UseFence has the value set to SCSI3.

```
# haclus -value UseFence
```

3 To disable preferred fencing and use the default race policy, set the value of the cluster-level attribute PreferredFencingPolicy as Disabled.

```
# haconf -makerw
# haclus -modify PreferredFencingPolicy Disabled
# haconf -dump -makero
```

About I/O fencing log files

Refer to the appropriate log files for information related to a specific utility. The log file location for each utility or command is as follows:

- vxfen start and stop logs: `/var/VRTSvcs/log/vxfen/vxfen.log`
- vxfenclearpre utility: `/var/VRTSvcs/log/vxfen/vxfen.log`
- vxfenctl utility: `/var/VRTSvcs/log/vxfen/vxfenctl.log*`
- vxfenswap utility: `/var/VRTSvcs/log/vxfen/vxfenswap.log*`
- vxfentsthdw utility: `/var/VRTSvcs/log/vxfen/vxfentsthdw.log*`

The asterisk ,*, represents a number that differs for each invocation of the command.

Migrating from disk-based fencing to server-based fencing using the installer

The following procedure lists steps to perform migration using the installer.

To migrate from disk-based fencing to server-based fencing using the installer

1. Make sure system-to-system communication is functioning properly.
2. Make sure that the SF Oracle RAC cluster is online and uses disk-based fencing.

   ```
   # vxfenadm -d
   ```

   For example, if SF Oracle RAC cluster uses disk-based fencing:

   ```
   I/O Fencing Cluster Information:
   ================================
   Fencing Protocol Version: 201
   Fencing Mode: SCSI3
   Fencing SCSI3 Disk Policy: dmp
   Cluster Members:
     * 0 (sys1)
     1 (sys2)
   RFSM State Information:
     node 0 in state 8 (running)
     node 1 in state 8 (running)
   ```

3. On any node in the cluster, start the installer with the `-fencing` option.

   ```
   # /opt/VRTS/install/installer -fencing
   ```

   Note the location of log files which you can access in the event of any problem with the configuration process.

4. Confirm that you want to proceed with the I/O fencing configuration.

   The installer verifies whether I/O fencing is configured in enabled mode.

5. Confirm that you want to reconfigure I/O fencing.
6. Review the I/O fencing configuration options that the program presents. Type **4** to migrate to server-based I/O fencing.

   ```
   Select the fencing mechanism to be configured in this
   Application Cluster [1-4,q] 4
   ```

7 From the list of coordination points that the installer presents, select the coordination points that you want to replace.

For example:

```
Select the coordination points you would like to remove
from the currently configured coordination points:
   1) emc_clariion0_62
   2) emc_clariion0_65
   3) emc_clariion0_66
   4) All
   5) None
   b) Back to previous menu
Enter the options separated by spaces: [1-5,b,q,?] (5)? 1 2
```

If you want to migrate to server-based fencing with no coordinator disks, type **4** to remove all the coordinator disks.

8 Enter the total number of new coordination points.

If you want to migrate to server-based fencing configuration with a mix of coordination points, the number you enter at this prompt must be a total of both the new CP servers and the new coordinator disks.

9 Enter the total number of new coordinator disks.

If you want to migrate to server-based fencing with no coordinator disks, type **0** at this prompt.

10 Enter the total number of virtual IP addresses or host names of the virtual IP address for each of the CP servers.

11 Enter the virtual IP addresses or host names of the virtual IP address for each of the CP servers.

12 Verify and confirm the coordination points information for the fencing reconfiguration.

13 Review the output as the installer performs the following tasks:

- Removes the coordinator disks from the coordinator disk group.
- Updates the application cluster details on each of the new CP servers.
- Prepares the `vxfenmode.test` file on all nodes.
- Runs the `vxfenswap` script.
  Note the location of the `vxfenswap.log` file which you can access in the event of any problem with the configuration process.
- Completes the I/O fencing migration.

14 If you want to send this installation information to Veritas, answer **y** at the prompt.

```
Would you like to send the information about this installation
to Veritas to help improve installation in the future? [y,n,q,?] (y) y
```

15 After the migration is complete, verify the change in the fencing mode.

```
# vxfenadm -d
```

For example, after the migration from disk-based fencing to server-based fencing in the SF Oracle RAC cluster:

```
I/O Fencing Cluster Information:
================================
Fencing Protocol Version: 201
Fencing Mode: Customized
Fencing Mechanism: cps
Cluster Members:
  * 0 (sys1)
  1 (sys2)
RFSM State Information:
  node 0 in state 8 (running)
  node 1 in state 8 (running)
```

16 Verify the current coordination points that the vxfen driver uses.

```
# vxfenconfig -l
```

Migrating from server-based fencing to disk-based fencing using the installer

The following procedure lists the steps to perform migration using the installer.

To migrate from server-based fencing to disk-based fencing using the installer

1. Make sure system-to-system communication is functioning properly.
2. Make sure that the SF Oracle RAC cluster is configured to use server-based fencing.

   ```
   # vxfenadm -d
   ```

   For example, if the SF Oracle RAC cluster uses server-based fencing, the output appears similar to the following:

   ```
   I/O Fencing Cluster Information:
   ================================
   Fencing Protocol Version: 201
   Fencing Mode: Customized
   Fencing Mechanism: cps
   Cluster Members:
     * 0 (sys1)
     1 (sys2)
   RFSM State Information:
     node 0 in state 8 (running)
     node 1 in state 8 (running)
   ```

3. On any node in the cluster, start the installer with the `-fencing` option.

   ```
   # /opt/VRTS/install/installer -fencing
   ```

   Note the location of log files which you can access in the event of any problem with the configuration process.
4. Confirm that you want to proceed with the I/O fencing configuration.

   The installer verifies whether I/O fencing is configured in enabled mode.
5. Confirm that you want to reconfigure I/O fencing.
6. Review the I/O fencing configuration options that the program presents. Type **4** to migrate to disk-based I/O fencing.

   ```
   Select the fencing mechanism to be configured in this
   Application Cluster [1-4,q] 4
   ```

7 From the list of coordination points that the installer presents, select the coordination points that you want to replace.

For example:

```
Select the coordination points you would like to remove
from the currently configured coordination points:
   1) emc_clariion0_62
   2) [10.209.80.197]:14250,[10.209.80.199]:14300
   3) [10.209.80.198]:14250
   4) All
   5) None
   b) Back to previous menu
Enter the options separated by spaces: [1-5,b,q,?] (5)? 2 3
```

8 Enter the total number of new coordination points.

9 Enter the total number of new coordinator disks.

10 From the list of available disks that the installer presents, select two disks which you want to configure as coordinator disks.

For example:

```
List of available disks:
   1) emc_clariion0_61
   2) emc_clariion0_65
   3) emc_clariion0_66
   b) Back to previous menu
Select 2 disk(s) as coordination points. Enter the disk options
separated by spaces: [1-3,b,q]2 3
```

11 Verify and confirm the coordination points information for the fencing reconfiguration.

12 To migrate to disk-based fencing, select the I/O fencing mode as SCSI3.

```
Select the vxfen mode: [1-2,b,q,?] (1) 1
```

The installer initializes the coordinator disks and the coordinator disk group, and deports the disk group. Press **Enter** to continue.

13 Review the output as the installer prepares the `vxfenmode.test` file on all nodes and runs the `vxfenswap` script.

Note the location of the `vxfenswap.log` file which you can access in the event of any problem with the configuration process.

The installer cleans up the application cluster information from the CP servers.

14 If you want to send this installation information to Veritas, answer **y** at the prompt.

```
Would you like to send the information about this installation
to Veritas to help improve installation in the future? [y,n,q,?] (y) y
```

15 After the migration is complete, verify the change in the fencing mode.

```
# vxfenadm -d
```

For example, after the migration from server-based fencing to disk-based fencing in the SF Oracle RAC cluster:

```
I/O Fencing Cluster Information:
================================
Fencing Protocol Version: 201
Fencing Mode: SCSI3
Fencing SCSI3 Disk Policy: dmp
Cluster Members:
  * 0 (sys1)
  1 (sys2)
RFSM State Information:
  node 0 in state 8 (running)
  node 1 in state 8 (running)
```

16 Verify the current coordination points that the vxfen driver uses.

```
# vxfenconfig -l
```

Administering the CP server

This section provides the following CP server administration information:

- CP server administration user types and privileges
- CP server administration command (cpsadm)

This section also provides instructions for the following CP server administration tasks:

- Refreshing registration keys on the coordination points for server-based fencing
- Coordination Point replacement for an online cluster

- Migrating from non-secure to secure setup for CP server and SF Oracle RAC cluster communication

Refreshing registration keys on the coordination points for server-based fencing

Replacing keys on a coordination point (CP server) when the SF Oracle RAC cluster is online involves refreshing that coordination point's registrations. You can perform a planned refresh of registrations on a CP server without incurring application downtime on the SF Oracle RAC cluster. You must refresh registrations on a CP server if the CP server agent issues an alert on the loss of such registrations on the CP server database.

The following procedure describes how to refresh the coordination point registrations.

To refresh the registration keys on the coordination points for server-based fencing

1. Ensure that the SF Oracle RAC cluster nodes and users have been added to the new CP server(s). Run the following commands:

   ```
   # cpsadm -s cp_server -a list_nodes

   # cpsadm -s cp_server -a list_users
   ```

   If the SF Oracle RAC cluster nodes are not present here, prepare the new CP server(s) for use by the SF Oracle RAC cluster.

   See the *Storage Foundation for Oracle RAC Configuration and Upgrade Guide* for instructions.

2. Ensure that fencing is running on the cluster in customized mode using the coordination points mentioned in the `/etc/vxfenmode` file.

   If the /etc/vxfenmode.test file exists, ensure that the information in it and the /etc/vxfenmode file are the same. Otherwise, vxfenswap utility uses information listed in /etc/vxfenmode.test file.

   For example, enter the following command:

   ```
   # vxfenadm -d

   ================================
   Fencing Protocol Version: 201
   Fencing Mode: CUSTOMIZED
   Cluster Members:
   * 0 (sys1)
   1 (sys2)
   RFSM State Information:
   node 0 in state 8 (running)
   node 1 in state 8 (running)
   ```

3. List the coordination points currently used by I/O fencing :

   ```
   # vxfenconfig -l
   ```

4. Copy the `/etc/vxfenmode` file to the `/etc/vxfenmode.test` file.

   This ensures that the configuration details of both the files are the same.

5 Run the `vxfenswap` utility from one of the nodes of the cluster.

The `vxfenswap` utility requires secure ssh connection to all the cluster nodes. Use `-n` to use rsh instead of default ssh.

For example:

```
# vxfenswap [-n]
```

The command returns:

```
VERITAS vxfenswap version <version> <platform>
The logfile generated for vxfenswap is
/var/VRTSvcs/log/vxfen/vxfenswap.log.
19156
Please Wait...
VXFEN vxfenconfig NOTICE Driver will use customized fencing
- mechanism cps
Validation of coordination points change has succeeded on
all nodes.
You may commit the changes now.
WARNING: This may cause the whole cluster to panic
if a node leaves membership before the change is complete.
```

6 You are then prompted to commit the change. Enter **y** for yes.

The command returns a confirmation of successful coordination point replacement.

7 Confirm the successful execution of the vxfenswap utility. If CP agent is configured, it should report ONLINE as it succeeds to find the registrations on coordination points. The registrations on the CP server and coordinator disks can be viewed using the `cpsadm` and `vxfenadm` utilities respectively.

Note that a running online coordination point refreshment operation can be canceled at any time using the command:

```
# vxfenswap -a cancel
```

Replacing coordination points for server-based fencing in an online cluster

Use the following procedure to perform a planned replacement of customized coordination points (CP servers or SCSI-3 disks) without incurring application downtime on an online SF Oracle RAC cluster.

Note: If multiple clusters share the same CP server, you must perform this replacement procedure in each cluster.

You can use the vxfenswap utility to replace coordination points when fencing is running in customized mode in an online cluster, with vxfen_mechanism=cps. The utility also supports migration from server-based fencing (vxfen_mode=customized) to disk-based fencing (vxfen_mode=scsi3) and vice versa in an online cluster.

However, if the SF Oracle RAC cluster has fencing disabled (vxfen_mode=disabled), then you must take the cluster offline to configure disk-based or server-based fencing.

You can cancel the coordination point replacement operation at any time using the `vxfenswap -a cancel` command.

See "About the vxfenswap utility" on page 109.

To replace coordination points for an online cluster

1 Ensure that the SF Oracle RAC cluster nodes and users have been added to the new CP server(s). Run the following commands:

```
# cpsadm -s cpserver -a list_nodes
# cpsadm -s cpserver -a list_users
```

If the SF Oracle RAC cluster nodes are not present here, prepare the new CP server(s) for use by the SF Oracle RAC cluster.

See the *Storage Foundation for Oracle RAC Configuration and Upgrade Guide* for instructions.

2 Ensure that fencing is running on the cluster using the old set of coordination points and in customized mode.

For example, enter the following command:

```
# vxfenadm -d
```

The command returns:

```
I/O Fencing Cluster Information:
================================
Fencing Protocol Version: <version>
Fencing Mode: Customized
Cluster Members:
* 0 (sys1)
1 (sys2)
RFSM State Information:
node 0 in state 8 (running)
node 1 in state 8 (running)
```

3 Create a new `/etc/vxfenmode.test` file on each SF Oracle RAC cluster node with the fencing configuration changes such as the CP server information.

Review and if necessary, update the vxfenmode parameters for security, the coordination points, and if applicable to your configuration, vxfendg.

Refer to the text information within the `vxfenmode` file for additional information about these parameters and their new possible values.

4 From one of the nodes of the cluster, run the `vxfenswap` utility.

The `vxfenswap` utility requires secure ssh connection to all the cluster nodes. Use –n to use rsh instead of default ssh. Use –p <*protocol*>, where <*protocol*> can be ssh, rsh, or hacli.

```
# vxfenswap [-n | -p <protocol>]
```

5 Review the message that the utility displays and confirm whether you want to commit the change.

- If you do not want to commit the new fencing configuration changes, press Enter or answer n at the prompt.

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

  The `vxfenswap` utility rolls back the migration operation.

- If you want to commit the new fencing configuration changes, answer y at the prompt.

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

  If the utility successfully completes the operation, the utility moves the`/etc/vxfenmode.test` file to the `/etc/vxfenmode` file.

6 Confirm the successful execution of the `vxfenswap` utility by checking the coordination points currently used by the vxfen driver.

For example, run the following command:

```
# vxfenconfig -l
```

Migrating from non-secure to secure setup for CP server and SF Oracle RAC cluster communication

The following procedure describes how to migrate from a non-secure to secure set up for the coordination point server (CP server) and SF Oracle RAC cluster. The procedure is only applicable to Veritas Product Authentication Services (AT)-based communication between CP servers and SF Oracle RAC cluster.

To migrate from non-secure to secure setup for CP server and SF Oracle RAC cluster

1. Stop VCS on all cluster nodes that use the CP servers.

   ```
   # hastop -all
   ```

2. Stop fencing on all the SF Oracle RAC cluster nodes of all the clusters.

   For RHEL 7, SLES 12, and supported RHEL distributions:

   ```
   # systemctl stop vxfen
   ```

   For earlier versions of RHEL, SLES, and supported RHEL distributions:

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

3. Stop all the CP servers using the following command on each CP server:

   ```
   # hagrp -offline CPSSG -any
   ```

4. Ensure that security is configured for communication on CP Servers as well as all the clients.

   See the *Cluster Server Configuration and Upgrade Guide* for more information.

5. - If CP server is hosted on an SFHA cluster, perform this step on each CP server.
     Bring the mount resource in the CPSSG service group online.

     ```
     # hares -online cpsmount -sys local_system_name
     ```

     Complete the remaining steps.
   - If CP server is hosted on a single-node VCS cluster, skip to step 8 and complete the remaining steps.

6. After the mount resource comes online, move the `credentials` directory from the default location to shared storage.

   ```
   # mv /var/VRTSvcs/vcsauth/data/CPSERVER /etc/VRTSvcs/db/
   ```

7. Create softlinks on all the nodes of the CP servers.

   ```
   # ln -s /etc/VRTScps/db/CPSERVER \
   /var/VRTSvcs/vcsauth/data/CPSERVER
   ```

8. Edit `/etc/vxcps.conf` on each CP server to set security=1.

9 Start CP servers by using the following command:

```
# hagrp -online CPSSG -any
```

10 Edit `/etc/VRTSvcs/conf/config/main.cf` on the first node of the cluster and remove the UseFence=SCSI3 attribute.

Start VCS on the first node and then on all other nodes of the cluster.

11 Reconfigure fencing on each cluster by using the installer.

```
# /opt/VRTS/install/installer -fencing
```

Administering CFS

This section describes some of the major aspects of Cluster File System (CFS) administration.

This section provides instructions for the following CFS administration tasks:

- Adding CFS file systems to VCS configuration
  See "Adding CFS file systems to a VCS configuration" on page 138.
- Resizing CFS file systems
  See "Resizing CFS file systems" on page 139.
- Verifying the status of CFS file systems
  See "Verifying the status of CFS file system nodes and their mount points" on page 139.

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

See "Troubleshooting CFS" on page 204.

Adding CFS file systems to a VCS configuration

Run the following command to add a Cluster File System (CFS) file system to the Cluster Server (VCS) `main.cf` file without using an editor.

For example:

```
# cfsmntadm add oradatadg oradatavol \
/oradata1 all=suid,rw

Mount Point is being added...
  /oradata1 added to the cluster-configuration
```

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

Resizing CFS file systems

If you see a message on the console indicating that a Cluster File System (CFS) file system is full, you may want to resize the file system. The `vxresize` command lets you resize a CFS file system. It extends the file system and the underlying volume.

See the `vxresize` (1M) manual page for information on various options.

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

```
# vxresize -g oradatadg oradatavol +2G
```

where *oradatadg* is the CVM disk group

where *oradatavol* is the volume

where *+2G* indicates the increase in volume size by 2 Gigabytes.

The following command shrinks an Oracle data CFS file system (the Oracle data volume is CFS mounted):

```
# vxresize -g oradatadg oradatavol -2G
```

where *-2G* indicates the decrease in volume size by 2 Gigabytes

Verifying the status of CFS file system nodes and their mount points

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

```
# cfscluster status

Node             :  sys1
Cluster Manager  :  running
CVM state        :  running

MOUNT POINT           SHARED VOLUME  DISK GROUP        STATUS
/app/crshome          crsbinvol      bindg             MOUNTED
/ocrvote              ocrvotevol     ocrvotedg         MOUNTED
/app/oracle/orahome   orabinvol      bindg             MOUNTED
/oradata1             oradatavol     oradatadg         MOUNTED
/arch                 archvol        oradatadg         MOUNTED
```

```
Node              :  sys2
Cluster Manager   :  running
CVM state         :  running

MOUNT POINT           SHARED VOLUME  DISK GROUP        STATUS
/app/crshome          crsbinvol      bindg             MOUNTED
/ocrvote              ocrvotevol     ocrvotedg         MOUNTED
/app/oracle/orahome   orabinvol      bindg             MOUNTED
/oradata1             oradatavol     oradatadg         MOUNTED
/arch                 archvol        oradatadg         MOUNTED
```

Administering CVM

This section provides instructions for the following CVM administration tasks:

- Listing all the CVM shared disks
  See "Listing all the CVM shared disks" on page 141.
- Establishing CVM cluster membership manually
  See "Establishing CVM cluster membership manually" on page 141.
- Changing CVM master manually
  See "Changing the CVM master manually" on page 141.
- Importing a shared disk group manually
  See "Importing a shared disk group manually" on page 144.
- Deporting a shared disk group manually
  See "Deporting a shared disk group manually " on page 145.
- Starting shared volumes manually
  See "Starting shared volumes manually" on page 145.
- Verifying if CVM is running in an SF Oracle RAC cluster
  See "Verifying if CVM is running in an SF Oracle RAC cluster" on page 145.
- Verifying CVM membership state
  See "Verifying CVM membership state" on page 146.
- Verifying the state of CVM shared disk groups
  See "Verifying the state of CVM shared disk groups" on page 146.
- Verifying the activation mode
  See "Verifying the activation mode" on page 146.

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

See “Troubleshooting Cluster Volume Manager in SF Oracle RAC clusters” on page 200.

Listing all the CVM shared disks

Use the following command to list all the Cluster Volume Manager shared disks:

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

Establishing CVM cluster membership manually

In most cases you do not have to start Cluster Volume Manager (CVM) manually; it normally starts when Cluster Server (VCS) is started.

Run the following command to start CVM manually:

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

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

To verify whether CVM is started properly, run the following command:

```
# vxclustadm nidmap
Name          CVM Nid     CM Nid      State
sys1        0           0         Joined: Master
sys2        1           1         Joined: Slave
```

Changing the CVM master manually

You can change the Cluster Volume Manager (CVM) master manually from one node in the cluster to another node, while the cluster is online. CVM migrates the master node, and reconfigures the cluster.

Veritas recommends that you switch the master when the cluster is not handling Veritas Volume Manager (VxVM) configuration changes or cluster reconfiguration operations. In most cases, CVM aborts the operation to change the master, if CVM detects that any configuration changes are occurring in the VxVM or the cluster. After the master change operation starts reconfiguring the cluster, other commands that require configuration changes will fail until the master switch completes.

See “Errors during CVM master switching” on page 144.

To change the master online, the cluster must be cluster protocol version 100 or greater.

To change the CVM master manually

1 To view the current master, use one of the following commands:

```
# vxclustadm nidmap
Name                CVM Nid    CM Nid     State
sys1                0          0          Joined: Slave
sys2                1          1          Joined: Master
```

```
# vxdctl -c mode
mode: enabled: cluster active - MASTER
master: sys2
```

In this example, the CVM master is sys2.

2 From any node on the cluster, run the following command to change the CVM master:

```
# vxclustadm setmaster nodename
```

where *nodename* specifies the name of the new CVM master.

The following example shows changing the master on a cluster from sys2 to sys1:

```
# vxclustadm setmaster sys1
```

3 To monitor the master switching, use the following command:

```
# vxclustadm -v nodestate
 state: cluster member
        nodeId=0
        masterId=0
        neighborId=1
        members[0]=0xf
        joiners[0]=0x0
        leavers[0]=0x0
        members[1]=0x0
        joiners[1]=0x0
        leavers[1]=0x0
        reconfig_seqnum=0x9f9767
        vxfen=off
state: master switching in progress
reconfig: vxconfigd in join
```

In this example, the state indicates that switching of the master is in progress.

4 To verify whether the master has successfully changed, use one of the following commands:

```
# vxclustadm nidmap
Name              CVM Nid    CM Nid    State
sys1            0          0         Joined: Master
sys2            1          1         Joined: Slave
```

```
# vxdctl -c mode
mode: enabled: cluster active - MASTER
master: sys1
```

Considerations for changing the master manually

If the master is not running on the node best suited to be the master of the cluster, you can manually change the master. Here are some scenarios when this might occur.

- The currently running master lost access to some of its disks.
  By default, CVM uses I/O shipping to handle this scenario. However, you may want to failover the application to a node that has access to the disks. When you move the application, you may also want to relocate the master role to a new node. For example, you may want the master node and the application to be on the same node.

You can use the master switching operation to move the master role without causing the original master node to leave the cluster. After the master role and the application are both switched to other nodes, you may want to remove the original node from the cluster. You can unmount the file systems and cleanly shut down the node. You can then do maintenance on the node.

- The master node is not scaling well with the overlap of application load and the internally-generated administrative I/Os.
  You may choose to reevaluate the placement strategy and relocate the master node.

Errors during CVM master switching

Veritas recommends that you switch the master when the cluster is not handling Veritas Volume Manager (VxVM) or cluster configuration changes.

In most cases, Cluster Volume Manager (CVM) aborts the operation to change the master, if CVM detects any configuration changes in progress. CVM logs the reason for the failure into the system logs. In some cases, the failure is displayed in the `vxclustadm setmaster` output as follows:

```
# vxclustadm setmaster sys1
VxVM vxclustadm ERROR V-5-1-15837 Master switching, a reconfiguration or
a transaction is in progress.
Try again
```

In some cases, if the master switching operation is interrupted with another reconfiguration operation, the master change fails. In this case, the existing master remains the master of the cluster. After the reconfiguration is complete, reissue the `vxclustadm setmaster` command to change the master.

If the master switching operation has started the reconfiguration, any command that initiates a configuration change fails with the following error:

```
Node processing a master-switch request. Retry operation.
```

If you see this message, retry the command after the master switching has completed.

Importing a shared disk group manually

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

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

Deporting a shared disk group manually

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

```
# vxdg deport dg_name
```

Note that the deport of a shared disk group removes the SCSI-3 PGR keys on the disks.

Starting shared volumes manually

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

```
# vxvol -g dg_name startall
```

To verify that the volumes are started, run the following command:

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

The volumes that are started will display the state "enabled."

Verifying if CVM is running in an SF Oracle RAC cluster

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

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

```
# vxdctl -c mode
mode: enabled: cluster inactive
# vxclustadm -v nodestate
state: out of cluster
```

On the master node, the following output is displayed:

```
# vxdctl -c mode

mode: enabled: cluster active - MASTER
master: sys1
```

On the slave nodes, the following output is displayed:

```
# vxdctl -c mode

mode: enabled: cluster active - SLAVE
master: sys2
```

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

```
# vxclustadm nidmap

Name       CVM Nid    CM Nid     State
sys1          0          0     Joined: Master
sys2          1          1     Joined: Slave
```

Verifying CVM membership state

The state of Cluster Volume Manager (CVM) can be verified as follows:

```
# vxclustadm -v nodestate

state: joining
        nodeId=0
        masterId=0
        neighborId=0
        members=0x1
        joiners=0x0
        leavers=0x0
        reconfig_seqnum=0x0
        reconfig: vxconfigd in join
```

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

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

Verifying the state of CVM shared disk groups

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

```
# vxdg list |grep shared

orabinvol_dg enabled,shared,cds 1052685125.1485.csha3
```

Verifying the activation mode

In an SF Oracle RAC cluster, the activation of shared disk groups should be set to “shared-write” on each of the cluster nodes.

To verify whether the “shared-write” activation is set:

```
# vxdg list dg_name |grep activation
```

```
local-activation: shared-write
```

If "shared-write" activation is not set, run the following command:

```
# vxdg -g dg_name set activation=sw
```

Administering Flexible Storage Sharing

Installing SF Oracle RAC automatically enables the Flexible Storage Sharing feature (FSS). No additional installation steps are required. LLT, GAB, and fencing must be configured before administering FSS. The fencing coordination points can either be SCSI-3 PR capable shared storage or CP servers.

See the *Storage Foundation for Oracle RAC Configuration and Upgrade Guide*.

FSS administrative tasks include exporting a disk for FSS, setting the FSS option on a disk group, setting the host prefix for intuitive naming of devices connected to the host, displaying exported disks and network shared disks, and optionally tuning FSS memory consumption.

See "About Flexible Storage Sharing" on page 29.

See "About Flexible Storage Sharing disk support" on page 147.

About Flexible Storage Sharing disk support

CVM relies on unique disk IDs (UDIDS) for identifying disks across different nodes in the cluster. FSS only supports disks that have the capability of generating unique IDs.

FSS supports the following disks:

- Disks that are listed in the Hardware Compatibility List (HCL): https://www.veritas.com/support/en_US/article.000126344
- Disks that have a JBOD definition specified using the `vxddladm addjbod` CLI. You must ensure that the specification is such that it provides a unique way of identifying a specific disk in your environment.
- SCSI-3 disks that provide an IEEE certified NAA ID in the VPD page 0x83 inquiry data

The `vxddladm checkfss diskname` command can be used to test if the disk complies with the required conditions. If the conditions are not met, adding the disks to an FSS configuration using the `vxdisk export diskname` command fails.

About the volume layout for Flexible Storage Sharing disk groups

By default, a volume in disk groups with the FSS attribute set is mirrored across hosts. This default layout ensures that data is available if any one host becomes unavailable. Associated instant data change object (DCO) log volumes are also created by default.

The following volume attributes are assumed by default:

- `mirror=host`
- `nmirror=2`
- `logtype=dco`
- `ndcomirror=2`
- `dcoversion=30`

You can specify the hosts on which to allocate the volume by using the host disk class.

See "Using the host disk class and allocating storage" on page 152.

Traditionally with disk groups without the FSS attribute set, the default volume layout is concatenated. However, you can still choose to create concatenated volumes in disk groups with the FSS attribute set by explicitly using the `layout=concat` option of the `vxassist` command.

By default, the mirrored volume is allocated across hosts. If host-specific storage is not available to meet this criteria, then the volume is allocated on external storage, with the default layout as concatenated as with traditional disk groups.

Existing disk classes, such as `dm`, can be used with FSS. The host prefix in a disk access name indicates the host to which the disk is connected.

For example, you can create a volume with one plex on a local disk (disk1), and another plex on a remote disk(hostA_disk2) where the host prefix (hostA) for the remote disk represents another host in the cluster:

```
# vxassist -g mydg make vol1 10g layout=mirror dm:disk1 dm:hostA_disk2
```

See "Administering mirrored volumes using vxassist" on page 152.

You can also use the `vxdisk list` *`accessname`* command to display connectivity information, such as information about which hosts are connected to the disk.

Setting the host prefix

The host prefix provides a way to intuitively identify the origin of the disk, for example, the host to which the disk is physically connected. The `vxdctl` command sets the

host prefix. By setting the instance for the host prefix, you can set an alternate host identifier that satisfies the disk access name length constraint.

Note: Disks from an array enclosure are not displayed with a host prefix.

In the case of direct attached storage (DAS) disks, a host prefix is added to the disk at the time of disk discovery itself to avoid disk naming conflicts.

In the case of array based enclosures connected through the SAN or Fibre Channel (FC), it is possible to connect different enclosures of the same type (same vendor, same model) to different nodes exclusively and thus create a DAS-type topology. In such cases, there is also the possibility of the same VxVM device name getting assigned to different disks on different nodes. If such disks are exported for use in an FSS environment, the same confusion over names can arise. In such cases however, the hostprefix will not be attached to the exported disks. The naming conflicts are resolved by adding a serial number to the VxVM device name for disks with same name. However, it is recommended that the name of the enclosure be changed for easy identification and readability.

The difference of naming behavior between DAS disks and enclosure based disks exists due to the following reasons. It is possible for the enclosure to be connected to only one node at any given instance of a CVM cluster. However, with a node join there could be two (or more) nodes connected to the enclosure at another instance. With nodes dynamically joining and leaving the cluster, connectivity to array enclosures can also change dynamically. Therefore, it is not reliable to use the connectivity information to decide if the topology is SAN or DAS, to decide whether the host-prefix needs to be added or not. As a result, CVM does not add a hostprefix to VxVM devices based on enclosure connectivity. Instead when a naming conflict occurs, a serial number is added to the VxVM device name. On the other hand DAS disks can be attached to only one node at a time, and thus it is safe to add a hostprefix by default (without waiting for the naming conflict to occur).

By default Cluster Volume Manager (CVM) uses the host name from the Cluster Server (VCS) configuration file as the host prefix. If the hostid in `/etc/vx/volboot` file is greater than 15 characters, and if a shorter host prefix is not set using `vxdctl`, Cluster Manager node IDs (CMID) are used as prefixes.

For more information, see the `vxdctl` (1M) manual page.

The following command sets/modifies the logical name for the host as the failure domain:

```
# vxdctl set hostprefix=logicalname
```

To unset the logical name for the host as the failure domain, use the following command:

```
# vxdctl unset hostprefix
```

The `vxdctl list` command displays the logical name set as the host prefix.

Exporting a disk for Flexible Storage Sharing

To use disks that are not physically shared across all nodes of a cluster (disks that are connected to one or more nodes of the cluster), the disks must first be exported for network sharing. Exporting a device makes the device available to all nodes in the cluster. The `vxdisk` command lets you export or unexport one or more disks for or from network sharing.

Note: To ensure data protection, FSS uses PGR-based data disk fencing. For disks that are purely locally attached, fencing is implicitly handled within FSS, since all the I/Os to the disk go through the node to which the disk is directly connected. Therefore, devices that do not support SCSI3 PGR are supported with FSS with fencing. For disks that are connected to multiple hosts but do not support SCSI3 PGR, there is no way to ensure that fencing is enabled and thus this configuration is not supported when fencing is enabled.

The following command exports one or more disks for network sharing:

```
# vxdisk export accessname1 accessname2
```

where *accessname1* and *accessname2* are the access names of the disks you want to export for network sharing.

In addition, you can use the `-o alldisks` and `-o local` options to export all local and all shared disks, and all locally connected disks, respectively.

Note: Boot disks, opaque disks, disks part of imported or deported disks groups, and non-VxVM disks cannot be exported.

Alternately, the following command unexports one or more disks from network sharing:

```
# vxdisk unexport accessname1 accessname2
```

where *accessname1* and *accessname2* are the disk access names of the disks you want to unexport from network sharing.

In addition, you can use the `-o alldisks` and `-o local` options to unexport all local and shared disks, and all locally connected disks, respectively.

Disks can also be configured for FSS by exporting them during initialization using the `vxdisksetup` and `vxdisk init` commands.

Initialize the disks with network sharing enabled, using one of the following commands:

```
# vxdisksetup -i disk_address export
```

```
# vxdisk [-f] init accessname export
```

where *disk_address* is the device that corresponds to the disk being operated on and *accessname* is the system-specific name that relates to the disk address.

After exporting a disk, you can use the `vxdisk list` and `vxprint` commands to list network shared disks, and identify disks that are remote to the node on which you run the command.

See "Displaying exported disks and network shared disk groups" on page 154.

Once a disk is exported, you can create shared disk groups using any combination of network shared disks and physically shared disks. Before adding an exported disk to a disk group, the Flexible Storage Sharing attribute needs to be set on the shared disk group.

See "Setting the Flexible Storage Sharing attribute on a disk group" on page 151.

Setting the Flexible Storage Sharing attribute on a disk group

The FSS attribute needs to be set on a disk group before any exported disks can be added to the disk group. This prevents the accidental addition of exported disks to disk groups. In addition, the tunable parameter `storage_connectivity` must be set to asymmetric. The FSS attribute can only be set on shared disk groups.

The `vxdg` command lets you create a disk group with the FSS attribute turned on or set the FSS attribute on an existing disk group.

Note: The disk group version must be set to 190 or higher in order to create or set an FSS disk group. When setting the FSS attribute on a disk group or importing an existing disk group as an FSS disk group, you may need to upgrade the disk group version.

The following command sets the FSS attribute on a disk group during initialization:

```
# vxdg -s -o fss init diskgroup [medianame=] accessname
```

where *diskgroup* is the disk group name, *medianame* is the administrative name chosen for a VM disk, and *accessname* is the device name or disk access name.

The following command sets the FSS attribute on a disk group:

```
# vxdg -g diskgroup set fss=on
```

The following command sets the FSS attribute on a disk group during a disk group import:

```
# vxdg -s -o fss=on import diskgroup
```

After setting the FSS attribute on a disk group, you can use the `vxdg list` command on a specified disk group to list all hosts in the cluster contributing their local disks to the disk group, and storage enclosures from which disks have been added to the disk group. Once you have created a disk group with the FSS attribute turned on or set the FSS attribute on an existing disk group, the ioship policy is set to on and the disk detach policy is set to local detach policy.

See "Displaying exported disks and network shared disk groups" on page 154.

Using the host disk class and allocating storage

You can use the `vxassist` command to use the host disk class for storage allocation. The host class specifies the host from which to allocate storage. The host class only applies to disk groups configured for FSS.

The host disk class instance is the same as the host name from the Cluster Server (VCS) configuration file, which can be displayed using the `vxclustadm nidmap` command.

Mirrored volumes can be created with mirrors across hosts using the host disk class using the `mirror=host` parameter:

```
# vxassist -g mydg make vol1 10g host:vm240v5 host:vm240v6

# vxassist -g mydg make vol1 10g mirror=host
```

Administering mirrored volumes using vxassist

By default, a volume in disk groups with the FSS attribute set is mirrored across hosts. You can use the `vxassist` command to create mirrored volume sets spanning across mixed media types, or using a combination of internal disks and external shared disks (i.e. SAN-attached disks).

Note: In SF Oracle RAC environments that use n-way mirroring, set the `nmirror` value to the number of nodes in the cluster when you create a mirrored volume. For example, if you have four nodes in the cluster, set the `nmirror` value to `4`.

To create a mirrored volume with HDD and SSD disks

1. Create a volume on either the HDD or the SSD disk:

   ```
   # vxassist -g diskgroup make vol1 maxsize layout=concat init=none
   ```

   Where *diskgroup* is the diskgroup name. HDD is selected by default.

2. Add a plex based on SSD(s):

   ```
   # vxassist -g diskgroup mirror vol1 mediatype:ssd
   ```

3. Activate the volume:

   ```
   # vxvol -g diskgroup init active vol1
   ```

Note: The mirrored volume may not preserve allocation constraints set.

If you have created a mirrored volume on an SSD disk using the `init=none` option, you can manually add a new mirror for a DCO volume using the following command:

```
# vxassist -g diskgroup mirror volume-dcl diskname
```

Where *diskgroup* is the diskgroup name and *diskname* is the name of a disk from the host on which the new data mirror was added. Veritas recommends that you specify the name of a disk on which the data mirror was created.

To create a mirrored volume using internal disks and external physically shared disks

1. Create a volume on either the internal disk or the external physically shared disks:

   ```
   # vxassist -g diskgroup make vol1 maxsize layout=concat init=none
   host:host1
   ```

   Where *diskgroup* is the diskgroup name.

2. Add a plex based on external shared disks:

   ```
   # vxassist -g diskgroup mirror vol1 enclr:emc0
   ```

3. Activate the volume:

   ```
   # vxvol -g diskgroup init active vol1
   ```

Note: Specifying the host disk class allocates the volume on the internal storage that is connected to the specified host.

See "Using the host disk class and allocating storage" on page 152.

Displaying exported disks and network shared disk groups

The `vxdisk list` and `vxprint` commands list network shared disks identifying the disks that are remote to the host from which the command is run. The `vxdisk list` command also provides an option to list disks while filtering out all remote disks.

To display exported disks, use the `vxdisk list` command:

```
# vxdisk list
DEVICE          TYPE          DISK     GROUP  STATUS
disk_01         auto:cdsdisk  -        -      online exported
disk_02         auto:cdsdisk  -        -      online exported
vm240v6_disk_01 auto:cdsdisk  -        -      online remote
vm240v6_disk_02 auto:cdsdisk  -        -      online remote
```

The disk name includes a prefix that indicates the host to which the disk is attached. For example, for disk `vm240v6_disk_01`, `vm240v6` is the host prefix. The `exported` status flag denotes disks that have been exported for FSS. The `remote` flag denotes disks that are not local to the host on which the command is run.

If the *accessname* argument is specified, disk connectivity information is displayed in the long listing output. This information is available only if the node on which the command is run is part of a CVM cluster.

The `-o local` option of the `vxdisk list` command filters out all remote disks.

For example:

```
# vxdisk -o local list
DEVICE          TYPE          DISK     GROUP  STATUS
disk_01         auto:cdsdisk  -        -      online exported
disk_02         auto:cdsdisk  -        -      online
```

The `-o fullshared` option displays all disks that are shared across all active nodes.

The `-o partialshared` option displays all disks that are partially shared. Partially shared disks are connected to more than one node but not all the active nodes in the cluster.

Alternately, you can use the `vxprint` command to display remote disks in a disk group:

```
# vxprint
Disk group: sdg

TY NAME              ASSOC           KSTATE   LENGTH     PLOFFS  STATE     TUTIL0  PUTIL0
dg sdg               sdg             -        -          -       -         -       -
dm disk_1            vm240v6_disk_1  -        2027264    -       REMOTE    -       -
dm disk_4            vm240v6_disk_4  -        2027264    -       REMOTE    -       -
dm disk_5            disk5           -        2027264    -       -         -       -
```

The `vxdg list` command displays hosts in the cluster that contribute their local disks to the disk group and storage enclosures from which disks have been added to the disk group. The hosts contributing their local disks to the disk groups and storage enclosures from which disks have been added to the disk group are listed under the `storage-sources` field.

Example output from this command is as follows:

```
Group:     mydg
dgid:      1343697721.24.vm240v5
import-id: 33792.24
flags:     shared cds
version:   190
alignment: 8192 (bytes)
detach-policy:local
ioship: on
fss: on
local-activation: shared-write
storage-sources: vm240v5 vm240v6 emc0
```

Tuning LLT for memory and performance in FSS environments

In remote direct memory access (RDMA) environments, you can limit the memory consumption for shipping I/O over the network by assigning the buffer pool memory in the LLT configuration file. The `LLT_BUFPOOL_MAXMEM` tunable lets you specify a minimum amount of memory that can be pre-allocated and the maximum amount of memory that can be allocated for the LLT buffer pool. This buffer pool is used to allocate memory for RDMA operations and packet allocation, which are delivered to the LLT clients. The default value is 4GB, the minimum value is 1GB, and the maximum value is 10GB. You must specify the value in GB.

For more information on tunables and LLT configuration files, see the appendix: “Tuning LLT for memory and performance in FSS environment” in the *Cluster Server Configuration and Upgrade Guide*.

Backing up and restoring disk group configuration data

The disk group configuration backup and restoration feature allows you to back up and restore all configuration data for disk groups, and for VxVM objects such as volumes that are configured within the disk groups. The `vxconfigbackupd` daemon monitors changes to the VxVM configuration and automatically records any configuration changes that occur. By default, `vxconfigbackup` stores 5 copies of the configuration backup and restoration (cbr) data. You can customize the number of cbr copies, between 1 to 5 copies.

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

VxVM provides the utilities, `vxconfigbackup` and `vxconfigrestore`, for backing up and restoring a VxVM configuration for a disk group.

See the *Veritas InfoScale Troubleshooting Guide*.

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

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

Backing up and restoring Flexible Storage Sharing disk group configuration data

The disk group configuration backup and restoration feature also lets you back up and restore configuration data for Flexible Storage Sharing (FSS) disk groups. The `vxconfigbackupd` daemon automatically records any configuration changes that occur on all cluster nodes. When restoring FSS disk group configuration data, you must first restore the configuration data on the secondary (slave) nodes in the cluster, which creates remote disks by exporting any locally connected disks. After restoring the configuration data on the secondary nodes, you must restore the configuration data on the primary (master) node that will import the disk group.

To back up FSS disk group configuration data

- To back up FSS disk group configuration data on all cluster nodes that have connectivity to at least one disk in the disk group, type the following command:

  ```
  # /etc/vx/bin/vxconfigbackup -T diskgroup
  ```

To restore the configuration data for an FSS disk group

1. Identify the master node:

   ```
   # vxclustadm nidmap
   ```

2. Check if the primary node has connectivity to at least one disk in the disk group. The disk can be a direct attached storage (DAS) disk, partially shared disk, or fully shared disks.

3. If the primary node does not have connectivity to any disk in the disk group, switch the primary node to a node that has connectivity to at least one DAS or partially shared disk, using the following command:

   ```
   # vxclustadm setmaster node_name
   ```

4. Restore the configuration data on all the secondary nodes:

   ```
   # vxconfigrestore diskgroup
   ```

   Note: You must restore the configuration data on all secondary nodes that have connectivity to at least one disk in the disk group.

5. Restore the configuration data on the primary node:

   ```
   # vxconfigrestore diskgroup
   ```

6. Verify the configuration data:

   ```
   # vxprint -g diskgroup
   ```

7. If the configuration data is correct, commit the configuration:

   ```
   # vxconfigrestore -c diskgroup
   ```

To abort or decommit configuration restoration for an FSS disk group

1 Identify the master node:

```
# vxclustadm nidmap
```

2 Abort or decommit the configuration data on the master node:

```
# vxconfigrestore -d diskgroup
```

3 Abort or decommit the configuration data on all secondary nodes.

```
# vxconfigrestore -d diskgroup
```

Note: You must abort or decommit the configuration data on all secondary nodes that have connectivity to at least one disk in the disk group, and all secondary nodes from which you triggered the precommit.

See the *Veritas InfoScale 7.4 Troubleshooting Guide*.

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

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

Administering SF Oracle RAC global clusters

This section provides instructions for the following global cluster administration tasks:

- About setting up a fire drill
  See "About setting up a disaster recovery fire drill" on page 159.
- Configuring the fire drill service group using the wizard
  See "About configuring the fire drill service group using the Fire Drill Setup wizard" on page 160.
- Verifying a successful fire drill
  See "Verifying a successful fire drill" on page 161.
- Scheduling a fire drill
  See "Scheduling a fire drill" on page 162.

For a sample fire drill service group configuration:

See "Sample fire drill service group configuration" on page 162.

About setting up a disaster recovery fire drill

The disaster recovery fire drill procedure tests the fault-readiness of a configuration by mimicking a failover from the primary site to the secondary site. This procedure is done without stopping the application at the primary site and disrupting user access, interrupting the flow of replicated data, or causing the secondary site to need resynchronization.

The initial steps to create a fire drill service group on the secondary site that closely follows the configuration of the original application service group and contains a point-in-time copy of the production data in the Replicated Volume Group (RVG). Bringing the fire drill service group online on the secondary site demonstrates the ability of the application service group to fail over and come online at the secondary site, should the need arise. Fire drill service groups do not interact with outside clients or with other instances of resources, so they can safely come online even when the application service group is online.

You must conduct a fire drill only at the secondary site; do not bring the fire drill service group online on the node hosting the original application.

You can override the FireDrill attribute and make fire drill resource-specific.

Before you perform a fire drill in a disaster recovery setup that uses Volume Replicator, perform the following steps:

- Configure the fire drill service group.
  See "About configuring the fire drill service group using the Fire Drill Setup wizard" on page 160.
- Set the value of the ReuseMntPt attribute to 1 for all Mount resources.
- After the fire drill service group is taken offline, reset the value of the ReuseMntPt attribute to 0 for all Mount resources.

VCS also supports HA fire drills to verify a resource can fail over to another node in the cluster.

Note: You can conduct fire drills only on regular VxVM volumes; volume sets (vset) are not supported.

VCS provides hardware replication agents for array-based solutions, such as Hitachi Truecopy, EMC SRDF, and so on . If you are using hardware replication agents to monitor the replicated data clusters, refer to the VCS replication agent documentation for details on setting up and configuring fire drill.

About configuring the fire drill service group using the Fire Drill Setup wizard

Use the Fire Drill Setup Wizard to set up the fire drill configuration.

The wizard performs the following specific tasks:

- Creates a Cache object to store changed blocks during the fire drill, which minimizes disk space and disk spindles required to perform the fire drill.
- Configures a VCS service group that resembles the real application group.

The wizard works only with application groups that contain one disk group. The wizard sets up the first RVG in an application. If the application has more than one RVG, you must create space-optimized snapshots and configure VCS manually, using the first RVG as reference.

You can schedule the fire drill for the service group using the `fdsched` script.

See "Scheduling a fire drill" on page 162.

Running the fire drill setup wizard

To run the wizard

1. Start the RVG Secondary Fire Drill wizard on the Volume Replicator secondary site, where the application service group is offline and the replication group is online as a secondary:

   ```
   # /opt/VRTSvcs/bin/fdsetup
   ```

2. Read the information on the Welcome screen and press the **Enter** key.
3. The wizard identifies the global service groups. Enter the name of the service group for the fire drill.
4. Review the list of volumes in disk group that could be used for a space-optimized snapshot. Enter the volumes to be selected for the snapshot. Typically, all volumes used by the application, whether replicated or not, should be prepared, otherwise a snapshot might not succeed.

   Press the **Enter** key when prompted.

5 Enter the cache size to store writes when the snapshot exists. The size of the cache must be large enough to store the expected number of changed blocks during the fire drill. However, the cache is configured to grow automatically if it fills up. Enter disks on which to create the cache.

Press the **Enter** key when prompted.

6 The wizard starts running commands to create the fire drill setup.

Press the **Enter** key when prompted.

The wizard creates the application group with its associated resources. It also creates a fire drill group with resources for the application (Oracle, for example), the CFSMount, and the RVGSnapshot types.

The application resources in both service groups define the same application, the same database in this example. The wizard sets the FireDrill attribute for the application resource to 1 to prevent the agent from reporting a concurrency violation when the actual application instance and the fire drill service group are online at the same time.

About configuring local attributes in the fire drill service group

The fire drill setup wizard does not recognize localized attribute values for resources. If the application service group has resources with local (per-system) attribute values, you must manually set these attributes after running the wizard.

Verifying a successful fire drill

Bring the fire drill service group online on a node that does not have the application running. Verify that the fire drill service group comes online. This action validates that your disaster recovery solution is configured correctly and the production service group will fail over to the secondary site in the event of an actual failure (disaster) at the primary site.

If the fire drill service group does not come online, review the VCS engine log to troubleshoot the issues so that corrective action can be taken as necessary in the production service group.

You can also view the fire drill log, located at `/tmp/fd-servicegroup.pid`

Remember to take the fire drill offline once its functioning has been validated. Failing to take the fire drill offline could cause failures in your environment. For example, if the application service group were to fail over to the node hosting the fire drill service group, there would be resource conflicts, resulting in both service groups faulting.

Scheduling a fire drill

You can schedule the fire drill for the service group using the fdsched script. The fdsched script is designed to run only on the lowest numbered node that is currently running in the cluster. The scheduler runs the command `hagrp - online` *`firedrill_group`* `-any` at periodic intervals.

To schedule a fire drill

1. Add the file `/opt/VRTSvcs/bin/fdsched` to your crontab.
2. To make fire drills highly available, add the `fdsched` file to each node in the cluster.

Sample fire drill service group configuration

The sample configuration in this section describes a fire drill service group configuration on the secondary site. The configuration uses VVR for replicating data between the sites.

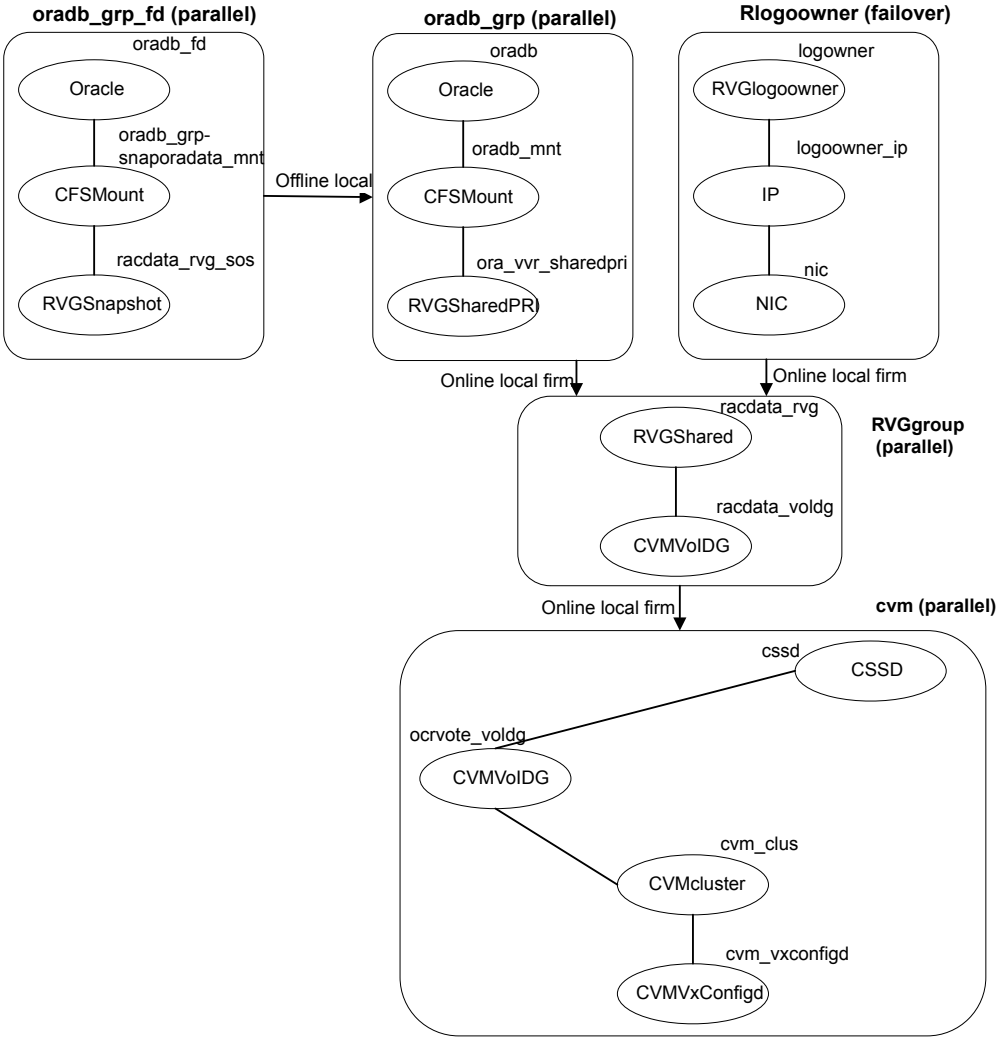
The sample service group describes the following configuration:

- Two SF Oracle RAC clusters, comprising two nodes each, hosted at different geographical locations.
- A single Oracle database that is stored on CFS.
- The database is managed by the VCS agent for Oracle.
  The agent starts, stops, and monitors the database.
- The database uses the Oracle UDP IPC for database cache fusion.
- A common IP address is used by Oracle Clusterware and database cache fusion. The private IP address is managed by the PrivNIC agent for high availability.
- One virtual IP address must be configured under the `ClusterService` group on each site for inter-cluster communication.
- For Oracle 11gR2 and 12cR1, the Oracle Cluster Registry (OCR) and voting files are stored on CFS. However, for Oracle 12cR2, the OCR and voting disks are stored on Oracle ASM, as Oracle 12cR2 does not support storage of OCR and voting files directly on CFS.
- Volume Replicator (VVR) is used to replicate data between the sites.
- The shared volumes replicated across the sites are configured under the RVG group.
- The replication link used by VVR for communicating log information between sites are configured under the `rlogowner` group. This is a failover group that will be online on only one of the nodes in the cluster at each site.

- The database group is configured as a global group by specifying the clusters on the primary and secondary sites as values for the ClusterList group attribute.
- The fire drill service group `oradb_grp_fd` creates a snapshot of the replicated data on the secondary site and starts the database using the snapshot. An offline local dependency is set between the fire drill service group and the application service group to make sure a fire drill does not block an application failover in case a disaster strikes the primary site.

Figure 2-1 illustrates the configuration.

Figure 2-1 Service group configuration for fire drill



Section 2

Performance and troubleshooting

Chapter 3

Troubleshooting SF Oracle RAC

This chapter includes the following topics:

About troubleshooting SF Oracle RAC

Use the information in this chapter to diagnose setup or configuration problems that you might encounter. For issues that arise from the component products, it may be necessary to refer to the appropriate documentation to resolve it.

Gathering information from an SF Oracle RAC cluster for support analysis

Use troubleshooting scripts to gather information about the configuration and status of your cluster and its modules. The scripts identify RPM 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.

- Gathering configuration information using SORT Data Collector
  See "Gathering configuration information using SORT Data Collector" on page 167.
- Gathering SF Oracle RAC information for support analysis
  See "Gathering SF Oracle RAC information for support analysis" on page 167.
- Gathering VCS information for support analysis
  See "Gathering VCS information for support analysis" on page 168.
- Gathering LLT and GAB information for support analysis
  See "Gathering LLT and GAB information for support analysis" on page 168.
- Gathering IMF information for support analysis
  See "Gathering IMF information for support analysis" on page 169.

Gathering configuration information using SORT Data Collector

SORT Data Collector now supersedes the VRTSexplorer utility.

Run the Data Collector with the `VRTSexplorer` option to gather system and configuration information from a node to diagnose or analyze issues in the cluster.

If you find issues in the cluster that require professional help, run the Data Collector and send the tar file output to Veritas Technical Support to resolve the issue.

Visit the SORT Website and download the UNIX Data Collector appropriate for your operating system:

https://sort.veritas.com

For more information:

https://sort.veritas.com/public/help/wwhelp/wwhimpl/js/html/wwhelp.htm

Gathering SF Oracle RAC information for support analysis

You must run the `getdbac` script to gather information about the SF Oracle RAC modules.

To gather SF Oracle RAC information for support analysis

- Run the following command on each node:

  ```
  # /opt/VRTSvcs/bin/getdbac -local
  ```

  The script saves the output to the default file `/tmp/vcsopslog.time_stamp.tar.Z`

  If you are unable to resolve the issue, contact Veritas Technical Support with the file.

Gathering VCS information for support analysis

You must run the `hagetcf` command to gather information when you encounter issues with VCS. Veritas Technical Support uses the output of these scripts to assist with analyzing and solving any VCS problems. The `hagetcf` command gathers information about the installed software, cluster configuration, systems, logs, and related information and creates a gzip file.

See the `hagetcf`(1M) manual page for more information.

To gather VCS information for support analysis

- Run the following command on each node:

  ```
  # /opt/VRTSvcs/bin/hagetcf
  ```

  The command prompts you to specify an output directory for the gzip file. You may save the gzip file to either the default `/tmp` directory or a different directory.

  Troubleshoot and fix the issue.

  If the issue cannot be fixed, then contact Veritas technical support with the file that the `hagetcf` command generates.

Gathering LLT and GAB information for support analysis

You must run the `getcomms` script to gather LLT and GAB information when you encounter issues with LLT and GAB. The `getcomms` script also collects core dump and stack traces along with the LLT and GAB information.

To gather LLT and GAB information for support analysis

1 If you had changed the default value of the GAB_FFDC_LOGDIR parameter, you must again export the same variable before you run the `getcomms` script.

See "GAB message logging" on page 174.

2 Run the following command to gather information:

```
# /opt/VRTSgab/getcomms
```

The script uses rsh by default. Make sure that you have configured passwordless rsh. If you have passwordless ssh between the cluster nodes, you can use the `-ssh` option. To gather information on the node that you run the command, use the `-local` option.

Troubleshoot and fix the issue.

If the issue cannot be fixed, then contact Veritas technical support with the file `/tmp/commslog.<time_stamp>.tar` that the `getcomms` script generates.

Gathering IMF information for support analysis

You must run the `getimf` script to gather information when you encounter issues with IMF (Intelligent Monitoring Framework).

To gather IMF information for support analysis

- Run the following command on each node:

```
# /opt/VRTSamf/bin/getimf
```

Troubleshoot and fix the issue.

If the issue cannot be fixed, then contact Veritas technical support with the file that the `getimf` script generates.

SF Oracle RAC log files

Table 3-1 lists the various log files and their location. The log files contain useful information for identifying issues and resolving them.

Table 3-1 List of log files

Log file	Location	Description
Oracle installation error log	*oraInventory_path*\/logs/\installActions*date_time*.log	Contains errors that occurred during Oracle RAC installation. It clarifies the nature of the error and when it occurred during the installation. **Note:** Verify if there are any installation errors logged in this file, since they may prove to be critical errors. If there are any installation problems, send this file to Tech Support for debugging the issue.
Oracle Clusterware/Grid Infrastructure logs	For Oracle RAC 11g Release 2 and 12c: $GRIDHOME/log/*node_name* For Oracle RAC 12.1.0.2 $GRIDBASE/diag/crs/node_name/crs/trace	Contains messages pertinent to the health of the clusterware.
Oracle alert log	*$ORACLE_BASE*/diag/rdbms/*db_name*/*instance_name*/trace/alert_*instance_name*.log The log path is configurable.	Contains messages and errors reported by database operations.
VCS engine log file	/var/VRTSvcs/log/engine_A.log	Contains all actions performed by the high availability daemon had. **Note:** Verify if there are any CVM or PrivNIC errors logged in this file, since they may prove to be critical errors.
CVM log files	/var/adm/vx/cmdlog /var/adm/vx/ddl.log /var/adm/vx/translog /var/adm/vx/dmpevents.log /var/VRTSvcs/log/engine_A.log	The cmdlog file contains the list of CVM commands. For more information on collecting important CVM logs: See "Collecting important CVM logs" on page 171.

Table 3-1 List of log files *(continued)*

Log file	Location	Description

VCS agent log files	`/var/VRTSvcs/log/agenttype_A.log`
	where *agenttype* is the type of the VCS agent.
	For example, the log files for the CFS agent can be located at:
	`/var/VRTSvcs/log/CFSMount_A.log`
		Contains messages and errors related to the agent functions.
		For more information, see the *Storage Foundation Administrator's Guide*.

OS system log	`/var/log/messages`	Contains messages and errors arising from operating system modules and drivers.

I/O fencing kernel logs	`/var/VRTSvcs/log/vxfen/vxfen.log`
	Obtain the logs by running the following command:
	`# /opt/VRTSvcs/vxfen/bin/\`
	`vxfendebug -p`
		Contains messages, errors, or diagnostic information for I/O fencing.

VCSMM log files	`/var/VRTSvcs/log/vcsmmconfig.log`	Contains messages, errors, or diagnostic information for VCSMM.

Collecting important CVM logs

You need to stop and restart the cluster to collect detailed CVM TIME_JOIN messages.

To collect detailed CVM TIME_JOIN messages

1 On all the nodes in the cluster, perform the following steps.

- Edit the `/opt/VRTSvcs/bin/CVMCluster/online` script.
  Insert the '-T' option to the following string.
  Original string: `clust_run=`LANG=C LC_MESSAGES=C $VXCLUSTADM -m vcs -t $TRANSPORT startnode 2> $CVM_ERR_FILE``

Modified string: **clust_run=`LANG=C LC_MESSAGES=C $VXCLUSTADM -T -d -m vcs -t $TRANSPORT startnode 2> $CVM_ERR_FILE`**

2. Stop the cluster.

   # hastop -all

3. Start the cluster.

   # hastart

   At this point, CVM TIME_JOIN messages display in the `/var/log/messages`file and on the console.

You can also enable vxconfigd daemon logging as follows:

```
# vxdctl debug 9 /var/adm/vx/vxconfigd_debug.out
```

The debug information that is enabled is accumulated in the system console log and in the text file `/var/adm/vx/vxconfigd_debug.out`. '9' represents the level of debugging. '1' represents minimal debugging. '9' represents verbose output.

Caution: Turning on vxconfigd debugging degrades VxVM performance. Use vxconfigd debugging with discretion in a production environment.

To disable vxconfigd debugging:

```
# vxdctl debug 0
```

The CVM kernel message dump can be collected on a live node as follows:

```
# /etc/vx/diag.d/kmsgdump -d 2000 > \
/var/adm/vx/kmsgdump.out
```

About SF Oracle RAC kernel and driver messages

SF Oracle RAC drivers such as GAB print messages to the console if the kernel and driver messages are configured to be displayed on the console. Make sure that the kernel and driver messages are logged to the console.

For details on how to configure console messages, see the operating system documentation.

VCS message logging

VCS generates two types of logs: the engine log and the agent log. Log file names are appended by letters; letter A indicates the first log file, B the second, and C the third. After three files, the least recent file is removed and another file is created.

For example, if engine_A.log is filled to its capacity, it is renamed as engine_B.log and the engine_B.log is renamed as engine_C.log. In this example, the least recent file is engine_C.log and therefore it is removed. You can update the size of the log file using the LogSize cluster level attribute.

The engine log is located at /var/VRTSvcs/log/engine_A.log. The format of engine log messages is:

Timestamp (Year/MM/DD) | Mnemonic | Severity | UMI | Message Text

- *Timestamp*: the date and time the message was generated.
- *Mnemonic*: the string ID that represents the product (for example, VCS).
- *Severity*: levels include CRITICAL, ERROR, WARNING, NOTICE, and INFO (most to least severe, respectively).
- *UMI*: a unique message ID.
- *Message Text*: the actual message generated by VCS.

A typical engine log resembles:

```
2011/07/10 16:08:09 VCS INFO V-16-1-10077 Received new
cluster membership
```

The agent log is located at `/var/VRTSvcs/log/<agent>.log`. The format of agent log messages resembles:

Timestamp (Year/MM/DD) | Mnemonic | Severity | UMI | Agent Type | Resource Name | Entry Point | Message Text

A typical agent log resembles:

```
2011/07/10 10:38:23 VCS WARNING V-16-2-23331
Oracle:VRT:monitor:Open for ora_lgwr failed, setting
cookie to null.
```

Note that the logs on all nodes may not be identical because

- VCS logs local events on the local nodes.
- All nodes may not be running when an event occurs.

VCS prints the warning and error messages to STDERR.

If the VCS engine, Command Server, or any of the VCS agents encounter some problem, then First Failure Data Capture (FFDC) logs are generated and dumped along with other core dumps and stack traces to the following location:

- For VCS engine: `$VCS_DIAG/diag/had`
- For Command Server: `$VCS_DIAG/diag/CmdServer`

- For VCS agents: `$VCS_DIAG/diag/agents/type`, where *type* represents the specific agent type.

The default value for variable $VCS_DIAG is `/var/VRTSvcs/`.

If the debug logging is not turned on, these FFDC logs are useful to analyze the issues that require professional support.

GAB message logging

If GAB encounters some problem, then First Failure Data Capture (FFDC) logs are also generated and dumped.

When you have configured GAB, GAB also starts a GAB logging daemon (`/opt/VRTSgab/gablogd`). GAB logging daemon is enabled by default. You can change the value of the GAB tunable parameter `gab_ibuf_count` to disable the GAB logging daemon.

See "About GAB load-time or static tunable parameters" on page 220.

This GAB logging daemon collects the GAB related logs when a critical events such as an iofence or failure of the master of any GAB port occur, and stores the data in a compact binary form. You can use the `gabread_ffdc` utility as follows to read the GAB binary log files:

```
/opt/VRTSgab/gabread_ffdc-kernel_version  binary_logs_files_location
```

You can change the values of the following environment variables that control the GAB binary log files:

- GAB_FFDC_MAX_INDX: Defines the maximum number of GAB binary log files The GAB logging daemon collects the defined number of log files each of eight MB size. The default value is 20, and the files are named `gablog.1` through `gablog.20`. At any point in time, the most recent file is the `gablog.1` file.
- GAB_FFDC_LOGDIR: Defines the log directory location for GAB binary log files The default location is:

  `/var/log/gab_ffdc`

  Note that the gablog daemon writes its log to the `glgd_A.log` and `glgd_B.log` files in the same directory.

You can either define these variables in the following GAB startup file or use the `export` command. You must restart GAB for the changes to take effect.

```
/etc/sysconfig/gab
```

About debug log tags usage

The following table illustrates the use of debug tags:

Entity	Debug logs used
Agent functions	DBG_1 to DBG_21
Agent framework	DBG_AGTRACE
	DBG_AGDEBUG
	DBG_AGINFO
Icmp agent	DBG_HBFW_TRACE
	DBG_HBFW_DEBUG
	DBG_HBFW_INFO
HAD	DBG_AGENT (for agent-related debug logs)
	DBG_ALERTS (for alert debug logs)
	DBG_CTEAM (for GCO debug logs)
	DBG_GAB, DBG_GABIO (for GAB debug messages)
	DBG_GC (for displaying global counter with each log message)
	DBG_INTERNAL (for internal messages)
	DBG_IPM (for Inter Process Messaging)
	DBG_JOIN (for Join logic)
	DBG_LIC (for licensing-related messages)
	DBG_NTEVENT (for NT Event logs)
	DBG_POLICY (for engine policy)
	DBG_RSM (for RSM debug messages)
	DBG_TRACE (for trace messages)
	DBG_SECURITY (for security-related messages)
	DBG_LOCK (for debugging lock primitives)
	DBG_THREAD (for debugging thread primitives)
	DBG_HOSTMON (for HostMonitor debug logs)

Enabling debug logs for agents

This section describes how to enable debug logs for VCS agents.

To enable debug logs for agents

1 Set the configuration to read-write:

```
# haconf -makerw
```

2 Enable logging and set the desired log levels. Use the following command syntax:

```
# hatype -modify $typename LogDbg DBG_1 DBG_2 DBG_4 DBG_21
```

The following example shows the command line for the IPMultiNIC resource type.

```
# hatype -modify IPMultiNIC LogDbg DBG_1 DBG_2 DBG_4 DBG_21
```

See the description of the `LogDbg` attribute for more information.

3 For script-based agents, run the `halog` command to add the messages to the engine log:

```
# halog -addtags DBG_1 DBG_2 DBG_4 DBG_21
```

4 Save the configuration.

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

If DBG_AGDEBUG is set, the agent framework logs for an instance of the agent appear in the agent log on the node on which the agent is running.

Enabling debug logs for the VCS engine

You can enable debug logs for the VCS engine, VCS agents, and HA commands in two ways:

- To enable debug logs at run-time, use the `halog -addtags` command.
- To enable debug logs at startup, use the VCS_DEBUG_LOG_TAGS environment variable. You must set the VCS_DEBUG_LOG_TAGS before you start HAD or before you run HA commands.
  Examples:

```
# export VCS_DEBUG_LOG_TAGS="DBG_TRACE DBG_POLICY"
# hastart
```

```
# export VCS_DEBUG_LOG_TAGS="DBG_AGINFO DBG_AGDEBUG DBG_AGTRACE"
# hastart
```

```
# export VCS_DEBUG_LOG_TAGS="DBG_IPM"
# hagrp -list
```

Note: Debug log messages are verbose. If you enable debug logs, log files might fill up quickly.

Enabling debug logs for IMF

Run the following commands to enable additional debug logs for Intelligent Monitoring Framework (IMF). The messages get logged in the agent-specific log file `/var/VRTSvcs/log/<agentname>_A.log`.

To enable additional debug logs

1 For Process, Mount, and Application agents:

```
# hatype -modify <agentname> LogDbg
DBG_AGDEBUG DBG_AGTRACE DBG_AGINFO DBG_1 DBG_2
DBG_3 DBG_4 DBG_5 DBG_6 DBG_7
```

2 For Oracle and Netlsnr agents:

```
# hatype -modify <agentname> LogDbg
DBG_AGDEBUG DBG_AGTRACE DBG_AGINFO DBG_1 DBG_2
DBG_3 DBG_4 DBG_5 DBG_6 DBG_7
DBG_8 DBG_9 DBG_10
```

3 For CFSMount agent:

```
# hatype -modify <agentname> LogDbg
DBG_AGDEBUG DBG_AGTRACE DBG_AGINFO DBG_1 DBG_2
DBG_3 DBG_4 DBG_5 DBG_6 DBG_7
DBG_8 DBG_9 DBG_10 DBG_11 DBG_12
DBG_13 DBG_14 DBG_15 DBG_16
DBG_17 DBG_18 DBG_19 DBG_20 DBG_21
```

4 For CVMvxconfigd agent, you do not have to enable any additional debug logs.

5 For AMF driver in-memory trace buffer:

```
# amfconfig -S errlevel all all
```

If you had enabled AMF driver in-memory trace buffer, you can view the additional logs using the `amfconfig -p dbglog` command.

Message catalogs

VCS includes multilingual support for message catalogs. These binary message catalogs (BMCs), are stored in the following default locations. The variable *language* represents a two-letter abbreviation.

`/opt/VRTS/messages/`*`language`*`/`*`module_name`*

The VCS command-line interface displays error and success messages in VCS-supported languages. The `hamsg` command displays the VCS engine logs in VCS-supported languages.

The BMCs are:

amf.bmc	Asynchronous Monitoring Framework (AMF) messages
fdsetup.bmc	fdsetup messages
gab.bmc	GAB command-line interface messages
gcoconfig.bmc	gcoconfig messages
hagetcf.bmc	hagetcf messages
hagui.bmc	hagui messages
haimfconfig.bmc	haimfconfig messages
hazonesetup.bmc	hazonesetup messages
hazoneverify.bmc	hazoneverify messages
llt.bmc	LLT command-line interface messages
uuidconfig.bmc	uuidconfig messages
VRTSvcsAgfw.bmc	Agent framework messages
VRTSvcsAlerts.bmc	VCS alert messages
VRTSvcsApi.bmc	VCS API messages

VRTSvcsce.bmc	VCS cluster extender messages
VRTSvcsCommon.bmc	Common module messages
VRTSvcs<*platform*>Agent.bmc	VCS bundled agent messages
VRTSvcsHad.bmc	VCS engine (HAD) messages
VRTSvcsHbfw.bmc	Heartbeat framework messages
VRTSvcsNetUtils.bmc	VCS network utilities messages
VRTSvcs<*platformagent_name*>.bmc	VCS enterprise agent messages
VRTSvcsTriggers.bmc	VCS trigger messages
VRTSvcsVirtUtils.bmc	VCS virtualization utilities messages
VRTSvcsWac.bmc	Wide-area connector process messages
vxfen*.bmc	Fencing messages

Restarting the installer after a failed network connection

If an installation is aborted because of a failed network connection, restarting the installer will detect the previous installation. The installer prompts to resume the installation. If you choose to resume the installation, the installer proceeds from the point where the installation aborted. If you choose not to resume, the installation starts from the beginning.

Installer cannot create UUID for the cluster

The installer displays the following error message if the installer cannot find the uuidconfig.pl script before it configures the UUID for the cluster:

```
Couldn't find uuidconfig.pl for uuid configuration,
please create uuid manually before start vcs
```

You may see the error message during SF Oracle RAC configuration, upgrade, or when you add a node to the cluster using the installer.

Workaround: To start SF Oracle RAC, you must run the uuidconfig.pl script manually to configure the UUID on each cluster node.

To configure the cluster UUID when you create a cluster manually

- On one node in the cluster, perform the following command to populate the cluster UUID on each node in the cluster.

  ```
  # /opt/VRTSvcs/bin/uuidconfig.pl -clus -configure nodeA
  nodeB ... nodeN
  ```

  Where nodeA, nodeB, through nodeN are the names of the cluster nodes.

Troubleshooting SF Oracle RAC pre-installation check failures

Table 3-2 provides guidelines for resolving failures that may be reported when you start the SF Oracle RAC installation program.

Table 3-2 Troubleshooting pre-installation check failures

Error message	Cause	Resolution
Checking system communication...Failed	Passwordless secure shell or remote shell communication is not set up between the systems in the cluster.	Set up passwordless SSH or RSH communication between the systems in the cluster. For instructions, see the appendix "Configuring the secure shell or the remote shell for communications" in the *Storage Foundation for Oracle RAC Configuration and Upgrade Guide*.
Checking release compatibility...Failed	The current system architecture or operating system version is not supported. For example, 32-bit architectures are not supported in this release.	Make sure that the systems meet the hardware and software criteria required for the release. For information, see the chapter "Requirements and planning" in the *Storage Foundation for Oracle RAC Configuration and Upgrade Guide*.

Table 3-2 Troubleshooting pre-installation check failures *(continued)*

Error message	Cause	Resolution
Checking installed product	For information on the messages displayed for this check and the appropriate resolutions: See Table 3-3 on page 183.	For information on the messages displayed for this check and the appropriate resolutions: See Table 3-3 on page 183.
Checking platform version	The version of the operating system installed on the system is not supported.	Make sure that the systems meet the software criteria required for the release. For information, see the chapter "Requirements and planning" in the *Storage Foundation for Oracle RAC Configuration and Upgrade Guide*.

Table 3-2 Troubleshooting pre-installation check failures *(continued)*

Error message	Cause	Resolution
Performing product prechecks	One of the following checks failed: ■ Time synchronization ■ CPU speed checks ■ System architecture checks ■ OS patch level checks	For information on resolving these issues: ■ To resolve time synchronizaton issues, synchronize the time settings across the nodes. **Note:** For Oracle RAC 11g Release 2 and later versions, it is mandatory to configure NTP for synchronizing time on all nodes in the cluster. ■ To resolve CPU frequency issues, ensure that the nodes have the same type of processor. ■ To resolve system architecture check issues, ensure that the nodes have the same architecture and the same number and type of processors. ■ To resolve OS patch level issues, ensure that you have updated your system with the required OS patch levels. For information, see the chapter "Requirements and planning" in the *Storage Foundation for Oracle RAC Configuration and Upgrade Guide*.

Table 3-3 Checking installed product - messages

Message	Resolution
`Entered systems have different products installed: prod_name-prod_ver-sys_name Systems running different products must be upgraded independently.`	Two or more systems specified have different products installed. For example, SFCFSHA is installed on some systems while SF Oracle RAC is installed on other systems. You need to upgrade the systems separately. For instructions on upgrading SF Oracle RAC, see the chapter "Upgrading SF Oracle RAC" in the *Storage Foundation for Oracle RAC Configuration and Upgrade Guide*. For instructions on upgrading other installed products, see the appropriate product documentation.
`Entered systems have different versions of $prod_name installed: prod_name-prod_ver-sys_name. Systems running different product versions must be upgraded independently.`	Two or more systems specified have different versions of SF Oracle RAC installed. For example, SF Oracle RAC 7.4 is installed on some systems while SF Oracle RAC 6.0 is installed on other systems. You need to upgrade the systems separately. For instructions on upgrading SF Oracle RAC, see the chapter "Upgrading SF Oracle RAC" in the *Storage Foundation for Oracle RAC Configuration and Upgrade Guide*.
`prod_name is installed. Upgrading prod_name directly to another_prod_name is not supported.`	A product other than SF Oracle RAC is installed. For example, SFCFSHA 7.4 is installed on the systems. The SF Oracle RAC installer does not support direct upgrade from other products to SF Oracle RAC. You need to first upgrade to version 7.4 of the installed product, then upgrade to SF Oracle RAC 7.4.0. For instructions on upgrading to SF Oracle RAC, see the section "Upgrading from Storage Foundation products to SF Oracle RAC" in the *Storage Foundation for Oracle RAC Configuration and Upgrade Guide*.

Table 3-3 Checking installed product - messages *(continued)*

Message	Resolution
`prod_name is installed. component_prod_name is part of prod_name. Upgrading only component_prod_name version may partially upgrade the installed RPMs on the systems.`	If a previous version of SF Oracle RAC is installed, the installer supports partial product upgrade. For example, you can upgrade VCS in the stack to version 7.4. If you want to upgrade the complete SF Oracle RAC stack later, you can run the `installer` program.

Troubleshooting LLT health check warning messages

Table 3-4 lists the warning messages displayed during the health check and corresponding recommendations for resolving the issues.

Table 3-4 Troubleshooting LLT warning messages

Warning	Possible causes	Recommendation
`Warning: OS timer is not called for num seconds`	CPU and memory consumption on the node is high.	Check for applications that may be throttling the CPU and memory resources on the node.
`Warning: Kernel failed to allocate memory num time(s)`	The available memory on the node is insufficient.	Reduce memory consumption on the node and free up allocated memory.
`Flow-control occurred num time(s)and back-enabled num time(s) on port port number for node node number`	■ The network bandwidth between the local node and the peer node (*node number*) is insufficient. ■ The CPU and memory consumption on the peer node (*node number*) is very high.	■ Allocate more bandwidth for communication between the local node and the peer node. ■ Check for applications that may be throttling the CPU and memory resources on the node.

Table 3-4 Troubleshooting LLT warning messages *(continued)*

Warning	Possible causes	Recommendation
`Warning: Connectivity with node node id on link link id is flaky num time(s).` `The distribution is: (0-4 s) <num> (4-8 s) <num> (8-12 s) <num> (12-16 s) <num> (>=16 s)`	The private interconnect between the local node and the peer node is unstable.	Check the connectivity between the local node and the peer node. Replace the link, if needed.
`One or more link connectivity with peer node(s) node name is in trouble.`	The private interconnects between the local node and peer node may not have sufficient bandwidth.	Check the private interconnects between the local node and the peer node.
`Link connectivity with node id is on only one link. Veritas recommends configuring a minimum of 2 links.`	■ One of the configured private interconnects is non-operational. ■ Only one private interconnect has been configured under LLT.	■ If the private interconnect is faulty, replace the link. ■ Configure a minimum of two private interconnects.
`Only one link is configured under LLT. Veritas recommends configuring a minimum of 2 links.`	■ One of the configured private interconnects is non-operational. ■ Only one private interconnect has been configured under LLT.	■ If the private interconnect is faulty, replace the link. ■ Configure a minimum of two private interconnects.

Table 3-4 Troubleshooting LLT warning messages *(continued)*

Warning	Possible causes	Recommendation

`Retransmitted % percentage of total transmitted packets.`
`Sent % percentage of total transmitted packet when no link is up.`
`% percentage of total received packets are with bad checksum.`
`% percentage of total received packets are out of window.`
`% percentage of total received packets are misaligned.`
	■ The network interface card or the network links are faulty.	■ Verify the network connectivity between nodes.
		Check the network interface card for anomalies.

`% percentage of total received packets are with DLPI error.`	The DLPI driver or NIC may be faulty or corrupted.	Check the DLPI driver and NIC for anomalies.
`% per of total transmitted packets are with large xmit latency (>16ms) for port port id` `%per received packets are with large recv latency (>16ms) for port port id.`	The CPU and memory consumption on the node may be high or the network bandwidth is insufficient.	■ Check for applications that may be throttling CPU and memory usage. ■ Make sure that the network bandwidth is adequate for facilitating low-latency data transmission.

Table 3-4 Troubleshooting LLT warning messages *(continued)*

Warning	Possible causes	Recommendation
`LLT is not running.`	SF Oracle RAC is not configured properly.	Reconfigure SF Oracle RAC. For instructions, see the *Storage Foundation for Oracle RAC Configuration and Upgrade Guide*.

Troubleshooting I/O fencing

The following sections discuss troubleshooting the I/O fencing problems. Review the symptoms and recommended solutions.

SCSI reservation errors during bootup

When restarting a node of an SF Oracle RAC cluster, SCSI reservation errors may be observed such as:

```
date system name kernel: scsi3 (0,0,6) : RESERVATION CONFLICT
```

This message is printed for each disk that is a member of any shared disk group which is protected by SCSI-3 PR I/O fencing. This message may be safely ignored.

The vxfentsthdw utility fails when SCSI TEST UNIT READY command fails

While running the vxfentsthdw utility, you may see a message that resembles as follows:

```
Issuing SCSI TEST UNIT READY to disk reserved by other node
FAILED.
Contact the storage provider to have the hardware configuration
fixed.
```

The disk array does not support returning success for a `SCSI TEST UNIT READY` command when another host has the disk reserved using SCSI-3 persistent reservations. This happens with the Hitachi Data Systems 99XX arrays if bit 186 of the system mode option is not enabled.

Node is unable to join cluster while another node is being ejected

A cluster that is currently fencing out (ejecting) a node from the cluster prevents a new node from joining the cluster until the fencing operation is completed. The following are example messages that appear on the console for the new node:

```
...VxFEN ERROR V-11-1-25 ... Unable to join running cluster
since cluster is currently fencing
a node out of the cluster.
```

If you see these messages when the new node is booting, the vxfen startup script on the node makes up to five attempts to join the cluster.

To manually join the node to the cluster when I/O fencing attempts fail

- If the vxfen script fails in the attempts to allow the node to join the cluster, restart vxfen driver with the command:

  For RHEL 7, SLES 12, and supported RHEL distributions:

  ```
  # systemctl stop vxfen
  # systemctl start vxfen
  ```

  For earlier versions of RHEL, SLES, and supported RHEL distributions:

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

  If the command fails, restart the new node.

System panics to prevent potential data corruption

When a node experiences a split-brain condition and is ejected from the cluster, it panics and displays the following console message:

```
VXFEN:vxfen_plat_panic: Local cluster node ejected from cluster to
prevent potential data corruption.
```

A node experiences the split-brain condition when it loses the heartbeat with its peer nodes due to failure of all private interconnects or node hang. Review the behavior of I/O fencing under different scenarios and the corrective measures to be taken.

Cluster ID on the I/O fencing key of coordinator disk does not match the local cluster's ID

If you accidentally assign coordinator disks of a cluster to another cluster, then the fencing driver displays an error message similar to the following when you start I/O fencing:

```
000068 06:37:33 2bdd5845  0 ... 3066 0 VXFEN WARNING V-11-1-56
 Coordinator disk has key with cluster id 48813
 which does not match local cluster id 57069
```

The warning implies that the local cluster with the cluster ID 57069 has keys. However, the disk also has keys for cluster with ID 48813 which indicates that nodes from the cluster with cluster id 48813 potentially use the same coordinator disk.

You can run the following commands to verify whether these disks are used by another cluster. Run the following commands on one of the nodes in the local cluster. For example, on sys1:

```
sys1> # lltstat -C
57069

sys1> # cat /etc/vxfentab
/dev/vx/rdmp/disk_7
/dev/vx/rdmp/disk_8
/dev/vx/rdmp/disk_9

sys1> # vxfenadm -s /dev/vx/rdmp/disk_7
Reading SCSI Registration Keys...
Device Name: /dev/vx/rdmp/disk_7
Total Number Of Keys: 1
key[0]:
 [Numeric Format]:  86,70,48,49,52,66,48,48
 [Character Format]: VFBEAD00
 [Node Format]: Cluster ID: 48813 Node ID: 0 Node Name: unknown
```

Where *disk_7*, *disk_8*, and *disk_9* represent the disk names in your setup.

Recommended action: You must use a unique set of coordinator disks for each cluster. If the other cluster does not use these coordinator disks, then clear the keys using the vxfenclearpre command before you use them as coordinator disks in the local cluster.

See "About the vxfenclearpre utility" on page 105.

Fencing startup reports preexisting split-brain

The vxfen driver functions to prevent an ejected node from rejoining the cluster after the failure of the private network links and before the private network links are repaired.

For example, suppose the cluster of system 1 and system 2 is functioning normally when the private network links are broken. Also suppose system 1 is the ejected system. When system 1 restarts before the private network links are restored, its membership configuration does not show system 2; however, when it attempts to register with the coordinator disks, it discovers system 2 is registered with them. Given this conflicting information about system 2, system 1 does not join the cluster and returns an error from vxfenconfig that resembles:

```
vxfenconfig: ERROR: There exists the potential for a preexisting
split-brain. The coordinator disks list no nodes which are in
the current membership. However, they also list nodes which are
not in the current membership.

I/O Fencing Disabled!
```

Also, the following information is displayed on the console:

```
<date> <system name> vxfen: WARNING: Potentially a preexisting
<date> <system name> split-brain.
<date> <system name> Dropping out of cluster.
<date> <system name> Refer to user documentation for steps
<date> <system name> required to clear preexisting split-brain.
<date> <system name>
<date> <system name> I/O Fencing DISABLED!
<date> <system name>
<date> <system name> gab: GAB:20032: Port b closed
```

However, the same error can occur when the private network links are working and both systems go down, system 1 restarts, and system 2 fails to come back up. From the view of the cluster from system 1, system 2 may still have the registrations on the coordination points.

Assume the following situations to understand preexisting split-brain in server-based fencing:

- There are three CP servers acting as coordination points. One of the three CP servers then becomes inaccessible. While in this state, one client node leaves the cluster, whose registration cannot be removed from the inaccessible CP server. When the inaccessible CP server restarts, it has a stale registration from the node which left the SF Oracle RAC cluster. In this case, no new nodes can join the cluster. Each node that attempts to join the cluster gets a list of

registrations from the CP server. One CP server includes an extra registration (of the node which left earlier). This makes the joiner node conclude that there exists a preexisting split-brain between the joiner node and the node which is represented by the stale registration.

- All the client nodes have crashed simultaneously, due to which fencing keys are not cleared from the CP servers. Consequently, when the nodes restart, the vxfen configuration fails reporting preexisting split brain.

These situations are similar to that of preexisting split-brain with coordinator disks, where you can solve the problem running the `vxfenclearpre` command. A similar solution is required in server-based fencing using the `cpsadm` command.

See "Clearing preexisting split-brain condition" on page 191.

Clearing preexisting split-brain condition

Review the information on how the VxFEN driver checks for preexisting split-brain condition.

See "Fencing startup reports preexisting split-brain" on page 190.

See "About the vxfenclearpre utility" on page 105.

Table 3-5 describes how to resolve a preexisting split-brain condition depending on the scenario you have encountered:

Table 3-5 Recommened solution to clear pre-existing split-brain condition

Scenario	Solution
Actual potential split-brain condition—system 2 is up and system 1 is ejected	**1** Determine if system1 is up or not. **2** If system 1 is up and running, shut it down and repair the private network links to remove the split-brain condition. **3** Restart system 1.

Table 3-5 Recommened solution to clear pre-existing split-brain condition *(continued)*

Scenario	Solution
Apparent potential split-brain condition—system 2 is down and system 1 is ejected (Disk-based fencing is configured)	**1** Physically verify that system 2 is down. Verify the systems currently registered with the coordination points. Use the following command for coordinator disks: `# vxfenadm -s all -f /etc/vxfentab` The output of this command identifies the keys registered with the coordinator disks. **2** Clear the keys on the coordinator disks as well as the data disks in all shared disk groups using the `vxfenclearpre` command. The command removes SCSI-3 registrations and reservations. See "About the vxfenclearpre utility" on page 105. **3** Make any necessary repairs to system 2. **4** Restart system 2.
Apparent potential split-brain condition—system 2 is down and system 1 is ejected (Server-based fencing is configured)	**1** Physically verify that system 2 is down. Verify the systems currently registered with the coordination points. Use the following command for CP servers: `# cpsadm -s cp_server -a list_membership -c cluster_name` where *cp_server* is the virtual IP address or virtual hostname on which CP server is configured, and *cluster_name* is the VCS name for the SF Oracle RAC cluster (application cluster). The command lists the systems registered with the CP server. **2** Clear the keys on the coordinator disks as well as the data disks in all shared disk groups using the `vxfenclearpre` command. The command removes SCSI-3 registrations and reservations. After removing all stale registrations, the joiner node will be able to join the cluster. **3** Make any necessary repairs to system 2. **4** Restart system 2.

Registered keys are lost on the coordinator disks

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

To refresh the missing keys

- Use the vxfenswap utility to replace the coordinator disks with the same disks. The vxfenswap utility registers the missing keys during the disk replacement.

  See "Refreshing lost keys on coordinator disks" on page 120.

Replacing defective disks when the cluster is offline

If the disk in the coordinator disk group becomes defective or inoperable and you want to switch to a new diskgroup in a cluster that is offline, then perform the following procedure.

In a cluster that is online, you can replace the disks using the vxfenswap utility.

See "About the vxfenswap utility" on page 109.

Review the following information to replace coordinator disk in the coordinator disk group, or to destroy a coordinator disk group.

Note the following about the procedure:

- When you add a disk, add the disk to the disk group vxfencoorddg and retest the group for support of SCSI-3 persistent reservations.
- You can destroy the coordinator disk group such that no registration keys remain on the disks. The disks can then be used elsewhere.

To replace a disk in the coordinator disk group when the cluster is offline

1. Log in as superuser on one of the cluster nodes.
2. If VCS is running, shut it down:

   ```
   # hastop -all
   ```

   Make sure that the port h is closed on all the nodes. Run the following command to verify that the port h is closed:

   ```
   # gabconfig -a
   ```

3. Stop the VCSMM driver on each node:

   For RHEL 7, SLES 12, and supported RHEL distributions:

   ```
   # systemctl stop vcsmm
   ```

   For earlier versions of RHEL, SLES, and supported RHEL distributions:

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

4 Stop I/O fencing on each node:

For RHEL 7, SLES 12, and supported RHEL distributions:

```
# systemctl stop vxfen
```

For earlier versions of RHEL, SLES, and supported RHEL distributions:

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

This removes any registration keys on the disks.

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

```
# vxdg -tfC import `cat /etc/vxfendg`
```

where:

-t specifies that the disk group is imported only until the node restarts.

-f specifies that the import is to be done forcibly, which is necessary if one or more disks is not accessible.

-C specifies that any import locks are removed.

6 To remove disks from the disk group, use the VxVM disk administrator utility, vxdiskadm.

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

- Verify whether the coordinator attribute is set to on.

  ```
  # vxdg list vxfencoorddg |  grep flags: | grep coordinator
  ```

- Destroy the coordinator disk group.

  ```
  # vxdg -o coordinator destroy vxfencoorddg
  ```

7 Add the new disk to the node and initialize it as a VxVM disk.

Then, add the new disk to the vxfencoorddg disk group:

- If you destroyed the disk group in step 6, then create the disk group again and add the new disk to it.
  See the *Storage Foundation for Oracle RAC Configuration and Upgrade Guide* for detailed instructions.
- If the diskgroup already exists, then add the new disk to it.

  ```
  # vxdg -g vxfencoorddg -o coordinator adddisk disk_name
  ```

8 Test the recreated disk group for SCSI-3 persistent reservations compliance.

See "Testing the coordinator disk group using the -c option of vxfentsthdw" on page 95.

9 After replacing disks in a coordinator disk group, deport the disk group:

```
# vxdg deport `cat /etc/vxfendg`
```

10 On each node, start the I/O fencing driver:

For RHEL 7, SLES 12, and supported RHEL distributions:

```
# systemctl start vxfen
```

For earlier versions of RHEL, SLES, and supported RHEL distributions:

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

11 On each node, start the VCSMM driver:

For RHEL 7, SLES 12, and supported RHEL distributions:

```
# systemctl start vcsmm
```

For earlier versions of RHEL, SLES, and supported RHEL distributions:

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

12 Verify that the I/O fencing module has started and is enabled.

```
# gabconfig -a
```

Make sure that port b and port o memberships exist in the output for all nodes in the cluster.

```
# vxfenadm -d
```

Make sure that I/O fencing mode is not disabled in the output.

13 If necessary, restart VCS on each node:

```
# hastart
```

Troubleshooting I/O fencing health check warning messages

Table 3-6 lists the warning messages displayed during the health check and corresponding recommendations for resolving the issues.

Table 3-6 Troubleshooting I/O fencing warning messages

Warning	Possible causes	Recommendation
`VxFEN is not running on all nodes in the cluster.`	I/O fencing is not enabled in the cluster.	Configure fencing. For instructions, see the *Storage Foundation for Oracle RAC Configuration and Upgrade Guide*.
`VxFEN is not configured in SCSI3 mode.`	■ If you are using disk-based I/O fencing, it is not configured in the SCSI3 mode.	Start fencing in SCSI3 mode on all nodes in the cluster.

`VxFEN is running with only one coordinator disk. Loss of this disk will prevent cluster reconfiguration on loss of a node. Veritas recommends configuring a minimum of 3 coordinator disks.`
`VxFEN is running with even number of coordinator disks. There must be odd number of coordinator disks.`
`Replace the disk` *`disk name`* `using OCDR procedure.`
	■ I/O fencing is enabled with only one coordinator disk.
	■ The number of disks configured for I/O fencing are even.
		■ Configure a minimum of three coordinator disks.
		■ Add or remove coordinators disks to meet the odd coordinator disk criterion. Use the `vxfenswap` utility to add or remove disks.
		See "About the vxfenswap utility" on page 109.
		■ If the coordinator disk is faulty, replace the disk using the `vxfenswap` utility.
		See "About the vxfenswap utility" on page 109.

`The coordinator disk (`*`disk name`*`) does not have the required key for the local node.`	The key may have been accidentally removed.	Refresh the key on the coordinator disk. For instructions: See "Refreshing lost keys on coordinator disks" on page 120.

Table 3-6 Troubleshooting I/O fencing warning messages *(continued)*

Warning	Possible causes	Recommendation
`SCSI3 write-exclusive reservation is missing on shared disk (disk_name)`	The SCSI3 reservation is accidentally removed.	Shut down and restart SF Oracle RAC. When CVM restarts, CVM re-registers the key and places the reservation.

Troubleshooting CP server

All CP server operations and messages are logged in the /var/VRTScps/log directory in a detailed and easy to read format. The entries are sorted by date and time. The logs can be used for troubleshooting purposes or to review for any possible security issue on the system that hosts the CP server.

The following files contain logs and text files that may be useful in understanding and troubleshooting a CP server:

- /var/VRTScps/log/cpserver_[ABC].log
- /var/VRTSvcs/log/vcsauthserver.log (Security related)
- If the vxcpserv process fails on the CP server, then review the following diagnostic files:
  - /var/VRTScps/diag/FFDC_CPS_*pid*_vxcpserv.log
  - /var/VRTScps/diag/stack_*pid*_vxcpserv.txt

Note: If the vxcpserv process fails on the CP server, these files are present in addition to a core file. VCS restarts vxcpserv process automatically in such situations.

The file /var/VRTSvcs/log/vxfen/vxfend_[ABC].log contains logs that may be useful in understanding and troubleshooting fencing-related issues on a SF Oracle RAC cluster (client cluster) node.

See “Troubleshooting issues related to the CP server service group” on page 198.

See “Checking the connectivity of CP server” on page 198.

See “Issues during fencing startup on SF Oracle RAC cluster nodes set up for server-based fencing” on page 199.

See “Issues during online migration of coordination points” on page 199.

Troubleshooting issues related to the CP server service group

If you cannot bring up the CPSSG service group after the CP server configuration, perform the following steps:

- Verify that the CPSSG service group and its resources are valid and properly configured in the VCS configuration.
- Check the VCS engine log (`/var/VRTSvcs/log/engine_[ABC].log`) to see if any of the CPSSG service group resources are FAULTED.
- Review the sample dependency graphs to make sure the required resources are configured correctly.

Checking the connectivity of CP server

You can test the connectivity of CP server using the `cpsadm` command.

You must have set the environment variables CPS_USERNAME and CPS_DOMAINTYPE to run the `cpsadm` command on the SF Oracle RAC cluster (client cluster) nodes.

To check the connectivity of CP server

- Run the following command to check whether a CP server is up and running at a process level:

  ```
  # cpsadm -s cp_server -a ping_cps
  ```

  where *cp_server* is the virtual IP address or virtual hostname on which the CP server is listening.

Troubleshooting server-based fencing on the SF Oracle RAC cluster nodes

The file /var/VRTSvcs/log/vxfen/vxfend_[ABC].log contains logs files that may be useful in understanding and troubleshooting fencing-related issues on a SF Oracle RAC cluster (application cluster) node.

Issues during fencing startup on SF Oracle RAC cluster nodes set up for server-based fencing

Table 3-7 Fencing startup issues on SF Oracle RAC cluster (client cluster) nodes

Issue	Description and resolution
`cpsadm` command on the SF Oracle RAC cluster gives connection error	If you receive a connection error message after issuing the `cpsadm` command on the SF Oracle RAC cluster, perform the following actions: ■ Ensure that the CP server is reachable from all the SF Oracle RAC cluster nodes. ■ Check the /etc/vxfenmode file and ensure that the SF Oracle RAC cluster nodes use the correct CP server virtual IP or virtual hostname and the correct port number. ■ For HTTPS communication, ensure that the virtual IP and ports listed for the server can listen to HTTPS requests.
Authorization failure	Authorization failure occurs when the nodes on the client clusters and or users are not added in the CP server configuration. Therefore, fencing on the SF Oracle RAC cluster (client cluster) node is not allowed to access the CP server and register itself on the CP server. Fencing fails to come up if it fails to register with a majority of the coordination points. To resolve this issue, add the client cluster node and user in the CP server configuration and restart fencing.
Authentication failure	If you had configured secure communication between the CP server and the SF Oracle RAC cluster (client cluster) nodes, authentication failure can occur due to the following causes: ■ The client cluster requires its own private key, a signed certificate, and a Certification Authority's (CA) certificate to establish secure communication with the CP server. If any of the files are missing or corrupt, communication fails. ■ If the client cluster certificate does not correspond to the client's private key, communication fails. ■ If the CP server and client cluster do not have a common CA in their certificate chain of trust, then communication fails.

Issues during online migration of coordination points

During online migration of coordination points using the vxfenswap utility, the operation is automatically rolled back if a failure is encountered during validation of coordination points from any of the cluster nodes.

Validation failure of the new set of coordination points can occur in the following circumstances:

- The /etc/vxfenmode.test file is not updated on all the SF Oracle RAC cluster nodes, because new coordination points on the node were being picked up from

an old /etc/vxfenmode.test file. The /etc/vxfenmode.test file must be updated with the current details. If the /etc/vxfenmode.test file is not present, vxfenswap copies configuration for new coordination points from the /etc/vxfenmode file.

- The coordination points listed in the /etc/vxfenmode file on the different SF Oracle RAC cluster nodes are not the same. If different coordination points are listed in the /etc/vxfenmode file on the cluster nodes, then the operation fails due to failure during the coordination point snapshot check.
- There is no network connectivity from one or more SF Oracle RAC cluster nodes to the CP server(s).
- Cluster, nodes, or users for the SF Oracle RAC cluster nodes have not been added on the new CP servers, thereby causing authorization failure.

Vxfen service group activity after issuing the vxfenswap command

The Coordination Point agent reads the details of coordination points from the `vxfenconfig -l` output and starts monitoring the registrations on them.

Thus, during vxfenswap, when the vxfenmode file is being changed by the user, the Coordination Point agent does not move to FAULTED state but continues monitoring the old set of coordination points.

As long as the changes to vxfenmode file are not committed or the new set of coordination points are not reflected in `vxfenconfig -l` output, the Coordination Point agent continues monitoring the old set of coordination points it read from `vxfenconfig -l` output in every monitor cycle.

The status of the Coordination Point agent (either ONLINE or FAULTED) depends upon the accessibility of the coordination points, the registrations on these coordination points, and the fault tolerance value.

When the changes to vxfenmode file are committed and reflected in the `vxfenconfig -l` output, then the Coordination Point agent reads the new set of coordination points and proceeds to monitor them in its new monitor cycle.

Troubleshooting Cluster Volume Manager in SF Oracle RAC clusters

This section discusses troubleshooting CVM problems.

Restoring communication between host and disks after cable disconnection

If a fiber 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. On all nodes, use the `fdisk -l` command to scan for new disks.

   It may take a few minutes before the host is capable of seeing the disk.
3. On all nodes, issue the following command to rescan the disks:

   ```
   # vxdisk scandisks
   ```

4. On the master node, reattach the disks to the disk group they were in and retain the same media name:

   ```
   # vxreattach
   ```

   This may take some time. For more details, see `vxreattach` (1M) manual page.

Shared disk group cannot be imported in SF Oracle RAC cluster

If you see a message resembling:

```
vxvm:vxconfigd:ERROR:vold_vcs_getnodeid(/dev/vx/rdmp/disk_name):
local_node_id < 0
Please make sure that CVM and vxfen are configured
and operating correctly
```

First, make sure that CVM is running. You can see the CVM nodes in the cluster by running the vxclustadm nidmap command.

```
# vxclustadm nidmap
Name          CVM Nid     CM Nid      State
sys1         1           0           Joined: Master
sys2         0           1           Joined: Slave
```

This above ouput shows that CVM is healthy, with system sys1 as the CVM master. If CVM is functioning correctly, then the output above is displayed when CVM cannot retrieve the node ID of the local system from the `vxfen` driver. This usually happens when port b is not configured.

To verify vxfen driver is configured

- Check the GAB ports with the command:

```
# gabconfig -a
```

Port b must exist on the local system.

Error importing shared disk groups in SF Oracle RAC cluster

The following message may appear when importing shared disk group:

```
VxVM vxdg ERROR V-5-1-12205 Disk group disk group name: import
failed: No valid disk found containing disk group
```

You may need to remove keys written to the disk.

For information about removing keys written to the disk:

See "Removing preexisting keys" on page 106.

Unable to start CVM in SF Oracle RAC cluster

If you cannot start CVM, check the consistency between the `/etc/llthosts` and `main.cf` files for node IDs.

You may need to remove keys written to the disk.

For information about removing keys written to the disk:

See "Removing preexisting keys" on page 106.

CVM group is not online after adding a node to the SF Oracle RAC cluster

The possible causes for the CVM group being offline after adding a node to the cluster are as follows:

- The cssd resource is configured as a critical resource in the cvm group.
- Other resources configured in the cvm group as critical resources are not online.

To resolve the issue if cssd is configured as a critical resource

1. Log onto one of the nodes in the existing cluster as the root user.
2. Configure the cssd resource as a non-critical resource in the cvm group:

```
# haconf -makerw
# hares -modify cssd Critical 0
# haconf -dump -makero
```

To resolve the issue if other resources in the group are not online

1. Log onto one of the nodes in the existing cluster as the root user.
2. Bring the resource online:

```
# hares -online resource_name -sys system_name
```

3. Verify the status of the resource:

```
# hastatus -resource  resource_name
```

4. If the resource is not online, configure it as a non-critical resource only if the resource is not critical :

```
# haconf -makerw
# hares -modify resource_name Critical 0
# haconf -dump -makero
```

CVMVolDg not online even though CVMCluster is online in SF Oracle RAC cluster

When the CVMCluster resource goes online, then all shared disk groups that have the auto-import flag set are automatically imported. If the disk group import fails for some reason, the CVMVolDg resources fault. Clearing and taking the CVMVolDg type resources offline does not resolve the problem.

To resolve the resource issue

1. Fix the problem causing the import of the shared disk group to fail.
2. Offline the cvm group containing the resource of type CVMVolDg as well as the service group containing the CVMCluster resource type.
3. Bring the cvm group containing the CVMCluster resource online.
4. Bring the cvm group containing the CVMVolDg resource online.

Shared disks not visible in SF Oracle RAC cluster

If the shared disks in `/dev` are not visible, perform the following tasks:

Make sure that all shared LUNs are discovered by the HBA and SCSI layer. This can be verified by running the `ls -ltr` command on any of the disks under `/dev/*`.

For example:

```
# ls -ltr /dev/disk_name
```

If all LUNs are not discovered by SCSI, the problem might be corrected by specifying `dev_flags` or `default_dev_flags` and *max_luns* parameters for the SCSI driver.

If the LUNs are not visible in `/dev/*` files, it may indicate a problem with SAN configuration or zoning.

Troubleshooting CFS

This section discusses troubleshooting CFS problems.

Incorrect order in root user's <library> path

An incorrect order in the root user's <library> path can cause the system to hang while changing the primary node in the Cluster File System or the RAC cluster.

If the <library> path of the root user contains an entry pointing to a Cluster File System (CFS) file system before the /usr/lib entry, the system may hang when trying to perform one of the following tasks:

- Changing the primary node for the CFS file system
- Unmounting the CFS files system on the primary node
- Stopping the cluster or the service group on the primary node

This configuration issue occurs primarily in a RAC environment with Oracle binaries installed on a shared CFS file system.

The following is an example of a <library path> that may cause the system to hang:

```
LIBPATH=/app/oracle/orahome/lib:/usr/lib:/usr/ccs/lib
```

In the above example, /app/oracle is a CFS file system, and if the user tries to change the primary node for this file system, the system will hang. The user is still able to ping and telnet to the system, but simple commands such as `ls` will not respond. One of the first steps required during the changing of the primary node is freezing the file system cluster wide, followed by a quick issuing of the `fsck` command to replay the intent log.

Since the initial entry in <library> path is pointing to the frozen file system itself, the `fsck` command goes into a deadlock situation. In fact, all commands (including `ls`) which rely on the <library> path will hang from now on.

The recommended procedure to correct for this problem is as follows: Move any entries pointing to a CFS file system in any user's (especially root) <library> path towards the end of the list after the entry for /usr/lib

Therefore, the above example of a <library path> would be changed to the following:

```
LIBPATH=/usr/lib:/usr/ccs/lib:/app/oracle/orahome/lib
```

Troubleshooting interconnects

This section discusses troubleshooting interconnect problems.

Example entries for mandatory devices

If you are using eth2 and eth3 for interconnectivity, use the following procedure examples to set mandatory devices.

To set mandatory devices entry in the /etc/sysconfig/network/config

Enter:

```
MANDATORY_DEVICES="eth2-00:04:23:AD:4A:4C
eth3-00:04:23:AD:4A:4D"
```

To set a persistent name entry in an interface file:

Enter the following information in the file: `/etc/sysconfig/network/ifcfg-eth-id-00:09:3d:00:cd:22` (name of the eth0 interface file on SLES systems) and in the file: `/etc/sysconfig/network-scripts/ifcfg-eth0` (name of the eth0 interface file on RHEL and supported RHEL-compatible distributions):

```
BOOTPROTO='static'
BROADCAST='10.212.255.255'
IPADDR='10.212.88.22'
MTU=' '
NETMASK='255.255.254.0'
NETWORK='10.212.88.0'
REMOTE_IP=' '
STARTMODE='onboot'
UNIQUE='RFE1.bBSepP2NetB'
```

```
_nm_name='bus-pci-0000:06:07.0'
PERSISTENT_NAME=eth0
```

Troubleshooting Oracle

This section discusses troubleshooting Oracle.

Error when starting an Oracle instance in SF Oracle RAC

If the VCSMM driver (the membership module) is not configured, an error displays while starting the Oracle instance that resembles:

```
ORA-29702: error occurred in Cluster Group Operation
```

To start the VCSMM driver:

For RHEL 7, SLES 12, and supported RHEL distributions:

```
# systemctl start vcsmm
```

For earlier versions of RHEL, SLES and supported RHEL distributions:

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

The command included in the `/etc/vcsmmtab` file enables the VCSMM driver to be started at system boot.

Clearing Oracle group faults

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

```
# hagrp -clear oracle_grp
```

```
# hagrp -online oracle_grp -sys node_name
```

Oracle log files show shutdown called even when not shutdown manually

The Oracle enterprise agent calls shutdown if monitoring of the Oracle resources fails. On all cluster nodes, review the following VCS and Oracle agent log files for any errors or status:

```
/var/VRTSvcs/log/engine_A.log
/var/VRTSvcs/log/Oracle_A.log
```

DBCA fails while creating an Oracle RAC database

Verify that the `hostname -i` command returns the public IP address of the current node. This command is used by the installer and the output is stored in the OCR. If `hostname -i` returns 127.0.0.1, it causes the DBCA to fail.

Oracle's clusterware processes fail to start

Verify that the Oracle RAC configuration meets the following configuration requirements:

- The correct private IP address is configured on the private link.
- The OCR and voting disks shared disk groups are accessible.
- The file systems containing OCR and voting disks are mounted.

Check the CSSD log files to learn more.

You can find the CSSD log files at `$GRID_HOME/log/node_name/cssd/*`

Consult the Oracle RAC documentation for more information.

Oracle Clusterware fails after restart

If the Oracle Clusterware fails to start after boot up, check for the occurrence of the following strings in the `/var/log/messages` file.

String value in the file: `Oracle CSSD failure. Rebooting for cluster integrity`	Oracle Clusterware may fail due to Oracle CSSD failure. The Oracle CSSD failure may be caused by one of the following events: ■ Communication failure occurred and Oracle Clusterware fenced out the node. ■ OCR and Vote disk became unavailable. ■ `ocssd` was killed manually. ■ Killing the `init.cssd` script.
String value in the file: `Waiting for file system containing`	The Oracle Clusterware installation is on a shared disk and the `init` script is waiting for that file system to be made available.
String value in the file: `Oracle Cluster Ready Services disabled by corrupt install`	The following file is not available or has corrupt entries: `/etc/oracle/scls_scr/\ hostname/root/crsstart.`

String value in the file: `OCR initialization failed accessing OCR device`	The shared file system containing the OCR is not available and Oracle Clusterware is waiting for it to become available.

Troubleshooting the Virtual IP (VIP) configuration in an SF Oracle RAC cluster

When troubleshooting issues with the VIP configuration, use the following commands and files:

- Check for network problems on all nodes:

  ```
  # ifconfig -a
  ```

- Verify the `/etc/hosts` file on each node. Or, make sure that the virtual host name is registered with the DNS server as follows:
- Verify the virtual host name on each node.

  ```
  # ping virtual_host_name
  ```

- Check the output of the following command:

  ```
  $GRID_HOME/bin/crsctl stat res -t
  ```

- On the problem node, use the command:

  ```
  $ $GRID_HOME/bin/srvctl start nodeapps -n node_name
  ```

Troubleshooting Oracle Clusterware health check warning messages in SF Oracle RAC clusters

Table 3-8 lists the warning messages displayed during the health check and corresponding recommendations for resolving the issues.

Table 3-8 Troubleshooting Oracle Clusterware warning messages

Warning	Possible causes	Recommendation
`Oracle Clusterware is not running.`	Oracle Clusterware is not started. Oracle Clusterware is waiting for dependencies such as OCR or voting disk or private IP addresses to be available.	■ Bring the cvm group online to start Oracle Clusterware: `# hagrp -online cvm \ -sys system_name` ■ Check the VCS log file `/var/VRTSvcs/log/engine_A.log` to determine the cause and fix the issue.
`No CSSD resource is configured under VCS.`	The CSSD resource is not configured under VCS.	Configure the CSSD resource under VCS and bring the resource online. See the *Storage Foundation for Oracle RAC Configuration and Upgrade Guide*.
`The CSSD resource name is not running.`	■ VCS is not running. ■ The dependent resources, such as the CFSMount or CVMVolDg resource for OCR and voting disk are not online. **Note:** For Oracle 12cR2, the CFSMount resources, is not applicable.	■ Start VCS: `# hastart` ■ Check the VCS log file `/var/VRTSvcs/log/engine_A.log` to determine the cause and fix the issue.
`Mismatch between LLT links llt nics and Oracle Clusterware links crs nics.`	The private interconnects used by Oracle Clusterware are not configured over LLT interfaces.	The private interconnects for Oracle Clusterware must use LLT links. Configure the private IP addresses on one of the LLT links. See the *Storage Foundation for Oracle RAC Configuration and Upgrade Guide*.
`Mismatch between Oracle Clusterware links crs nics and PrivNIC links private nics.`	The private IP addresses used by Oracle Clusterware are not configured under PrivNIC.	Configure the PrivNIC resource to monitor the private IP address used by Oracle Clusterware. See the *Storage Foundation for Oracle RAC Configuration and Upgrade Guide*.

Table 3-8 Troubleshooting Oracle Clusterware warning messages *(continued)*

Warning	Possible causes	Recommendation
`Mismatch between CRS nodes` *`crs nodes`* `and LLT nodes` *`llt nodes`*`.`	The host names configured during the Oracle Clusterware installation are not the same as the host names configured under LLT.	Make sure that the host names configured during the Oracle Clusterware installation are the same as the host names configured under LLT.

Troubleshooting ODM in SF Oracle RAC clusters

This section discusses troubleshooting ODM.

File System configured incorrectly for ODM shuts down Oracle

Linking Oracle RAC with the Veritas ODM libraries provides the best file system performance.

Review the instructions on creating the link and confirming that Oracle uses the libraries. Shared file systems in RAC clusters without ODM libraries linked to Oracle RAC may exhibit slow performance and are not supported.

If ODM cannot find the resources it needs to provide support for cluster file systems, it does not allow Oracle to identify cluster files and causes Oracle to fail at startup.

To verify cluster status, run the following command and review the output:

```
# cat /dev/odm/cluster

cluster status: enabled
```

If the status is "enabled," ODM is supporting cluster files. Any other cluster status indicates that ODM is not supporting cluster files. Other possible values include:

`pending`	ODM cannot yet communicate with its peers, but anticipates being able to eventually.
`failed`	ODM cluster support has failed to initialize properly. Check console logs.

`disabled`	ODM is not supporting cluster files. If you think ODM should be supporting the cluster files: ■ Check `/dev/odm` mount options using `mount`. If the `nocluster` option is being used, it can force the `disabled` cluster support state.

If `/dev/odm` is not mounted, no status can be reported.

Troubleshooting Flex ASM in SF Oracle RAC clusters

Table 3-9 lists the Flex ASM issues you may encounter.

Table 3-9 Troubleshooting Flex ASM in SF Oracle RAC clusters

Issue	Resolution
Flex ASM instance incorrectly reported as faulted during migration	If the migration of a Flex ASM instance to another node takes a long time, it may be reported as faulted though the instance is being taken offline. To resolve the issue, raise the tolerance limit to 5 (or a suitable value). This will allow for 5 (or more) monitor cycles before the instance is reported as offline or faulted.

Chapter 4

Prevention and recovery strategies

This chapter includes the following topics:

Verification of GAB ports in SF Oracle RAC cluster

The following ports need to be up on all the nodes of SF Oracle RAC cluster:

- Port a (GAB)
- Port b (I/O fencing)
- Port d (ODM)
- Port f (CFS)
- Port h (VCS)
- Port m (CVM - for SmartIO VxVM cache coherency)
- Port o (VCSMM - membership module for SF Oracle RAC)

- Port v (CVM - kernel messaging)
- Port w (CVM - vxconfigd)
- Port u (CVM - to ship commands from slave node to master node)
- Port y (CVM - I/O shipping)

The following command can be used to verify the state of GAB ports:

```
# gabconfig -a
```

GAB Port Memberships

```
Port a gen 7e6e7e05 membership 01
Port b gen 58039502 membership 01
Port d gen 588a7d02 membership 01
Port f gen 1ea84702 membership 01
Port h gen cf430b02 membership 01
Port m gen cf430b03 membership 01
Port o gen de8f0202 membership 01
Port u gen de4f0203 membership 01
Port v gen db411702 membership 01
Port w gen cf430b02 membership 01
Port y gen 73f449   membership 01
```

The data indicates that all the GAB ports are up on the cluster having nodes 0 and 1.

For more information on the GAB ports in SF Oracle RAC cluster, see the *Storage Foundation for Oracle RAC Configuration and Upgrade Guide*.

Examining GAB seed membership

The number of systems that participate in the cluster is specified as an argument to the `gabconfig` command in /etc/gabtab. In the following example, two nodes are expected to form a cluster:

```
# cat /etc/gabtab
```

```
/sbin/gabconfig -c -n2
```

GAB waits until the specified number of nodes becomes available to automatically create the port “a” membership. Port “a” indicates GAB membership for an SF Oracle RAC cluster node. Every GAB reconfiguration, such as a node joining or leaving increments or decrements this seed membership in every cluster member node.

A sample port ‘a’ membership as seen in `gabconfig -a` is shown:

```
Port a gen 7e6e7e01 membership 01
```

In this case, 7e6e7e01 indicates the “membership generation number” and 01 corresponds to the cluster’s “node map”. All nodes present in the node map reflects the same membership ID as seen by the following command:

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

The semi-colon is used as a placeholder for a node that has left the cluster. In the following example, node 0 has left the cluster:

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

Port a gen 7e6e7e04 membership ;1
```

When the last node exits the port “a” membership, there are no other nodes to increment the membership ID. Thus the port “a” membership ceases to exist on any node in the cluster.

When the last and the final system is brought back up from a complete cluster cold shutdown state, the cluster will seed automatically and form port “a” membership on all systems. Systems can then be brought down and restarted in any combination so long as at least one node remains active at any given time.

The fact that all nodes share the same membership ID and node map certifies that all nodes in the node map participates in the same port “a” membership. This consistency check is used to detect “split-brain” and “pre-existing split-brain” scenarios.

Split-brain occurs when a running cluster is segregated into two or more partitions that have no knowledge of the other partitions. The pre-existing network partition is detected when the "cold" nodes (not previously participating in cluster) start and are allowed to form a membership that might not include all nodes (multiple sub-clusters), thus resulting in a potential split-brain.

Note: I/O fencing prevents data corruption resulting from any split-brain scenarios.

Manual GAB membership seeding

It is possible that one of the nodes does not come up when all the nodes in the cluster are restarted, due to the "minimum seed requirement" safety that is enforced by GAB. Human intervention is needed to safely determine that the other node is in fact not participating in its own mini-cluster.

The following should be carefully validated before manual seeding, to prevent introducing split-brain and subsequent data corruption:

- Verify that none of the other nodes in the cluster have a port “a” membership
- Verify that none of the other nodes have any shared disk groups imported
- Determine why any node that is still running does not have a port “a” membership

Run the following command to manually seed GAB membership:

```
# gabconfig -cx
```

Refer to gabconfig (1M) for more details.

Evaluating VCS I/O fencing ports

I/O Fencing (VxFEN) uses a dedicated port that GAB provides for communication across nodes in the cluster. You can see this port as port ‘b’ when `gabconfig -a` runs on any node in the cluster. The entry corresponding to port ‘b’ in this membership indicates the existing members in the cluster as viewed by I/O Fencing.

GAB uses port "a" for maintaining the cluster membership and must be active for I/O Fencing to start.

To check whether fencing is enabled in a cluster, the ‘-d’ option can be used with vxfenadm (1M) to display the I/O Fencing mode on each cluster node. Port “b” membership should be present in the output of `gabconfig -a` and the output should list all the nodes in the cluster.

If the GAB ports that are needed for I/O fencing are not up, that is, if port “a” is not visible in the output of `gabconfig -a` command, LLT and GAB must be started on the node.

The following commands can be used to start LLT and GAB respectively:

To start LLT on each node:

For RHEL 7, SLES 12, and supported RHEL distributions:

```
# systemctl start llt
```

For earlier versions of RHEL, SLES, and supported RHEL distributions:

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

If LLT is configured correctly on each node, the console output displays:

```
LLT INFO V-14-1-10009 LLT Protocol available
```

To start GAB, on each node:

For RHEL 7, SLES 12, and supported RHEL distributions:

```
# systemctl start gab
```

For earlier versions of RHEL, SLES, and supported RHEL distributions:

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

If GAB is configured correctly on each node, the console output displays:

```
GAB INFO V-15-1-20021 GAB available
GAB INFO V-15-1-20026 Port a registration waiting for seed port
membership
```

Verifying normal functioning of VCS I/O fencing

It is mandatory to have VCS I/O fencing enabled in SF Oracle RAC cluster to protect against split-brain scenarios. VCS I/O fencing can be assumed to be running normally in the following cases:

- Fencing port 'b' enabled on the nodes

  ```
  # gabconfig -a
  ```

  To verify that fencing is enabled on the nodes:

  ```
  # vxfenadm -d
  ```

- Registered keys present on the coordinator disks

  ```
  # vxfenadm -s all -f /etc/vxfentab
  ```

Managing SCSI-3 PR keys in SF Oracle RAC cluster

I/O Fencing places the SCSI-3 PR keys on coordinator LUNs. The format of the key follows the naming convention wherein ASCII “A” is prefixed to the LLT ID of the system that is followed by 7 dash characters.

For example:

node 0 uses A-------

node 1 uses B-------

In an SF Oracle RAC/SF CFS/SF HA environment, VxVM/CVM registers the keys on data disks, the format of which is ASCII “A” prefixed to the LLT ID of the system followed by the characters “PGRxxxx” where ‘xxxx’ = i such that the disk group is the ith shared group to be imported.

For example: node 0 uses APGR0001 (for the first imported shared group).

In addition to the registration keys, VCS/CVM also installs a reservation key on the data LUN. There is one reservation key per cluster as only one node can reserve the LUN.

The following command lists the keys on a data disk group:

```
# vxdg list |grep data

sys1_data1 enabled,shared,cds 1201715530.28.pushover
```

Select the data disk belonging to sys1_data1:

```
# vxdisk -o alldgs list |grep sys1_data1
```

The following command lists the PR keys:

```
# vxdisk -o listreserve list sdh

.....

.....
```

Alternatively, the PR keys can be listed using `vxfenadm` command:

```
#echo "/dev/vx/dmp/sdh" > /tmp/disk71

# vxfenadm -s all -f /tmp/disk71

Device Name: /dev/vx/dmp/sdh
Total Number Of Keys: 2
key[0]:
        Key Value [Numeric Format]:  66,80,71,82,48,48,48,52
        Key Value [Character Format]: BPGR0004
key[1]:
        Key Value [Numeric Format]:  65,80,71,82,48,48,48,52
        Key Value [Character Format]: APGR0004
```

Evaluating the number of SCSI-3 PR keys on a coordinator LUN, if there are multiple paths to the LUN from the hosts

The utility vxfenadm (1M) can be used to display the keys on the coordinator LUN. The key value identifies the node that corresponds to each key. Each node installs a registration key on all the available paths to the LUN. Thus, the total number of registration keys is the sum of the keys that are installed by each node in the above manner.

See “About the vxfenadm utility” on page 100.

Detecting accidental SCSI-3 PR key removal from coordinator LUNs

The keys currently installed on the coordinator disks can be read using the following command:

```
# vxfenadm -s all -f /etc/vxfentab
```

There should be a key for each node in the operating cluster on each of the coordinator disks for normal cluster operation. There will be two keys for every node if you have a two-path DMP configuration.

Identifying a faulty coordinator LUN

The utility vxfentsthdw (1M) provided with I/O fencing can be used to identify faulty coordinator LUNs. This utility must be run from any node in the cluster. The coordinator LUN, which needs to be checked, should be supplied to the utility.

See “About the vxfentsthdw utility” on page 92.

Chapter 5

Tunable parameters

This chapter includes the following topics:

About SF Oracle RAC tunable parameters

Tunable parameters can be configured to enhance the performance of specific SF Oracle RAC features. This chapter discusses how to configure the following SF Oracle RAC tunables:

- GAB
- LLT
- VXFEN

Veritas recommends that you do not change the tunable kernel parameters without assistance from Veritas support personnel. Several of the tunable parameters preallocate memory for critical data structures, and a change in their values could increase memory use or degrade performance.

Warning: Do not adjust the SF Oracle RAC tunable parameters for LMX and VXFEN as described below to enhance performance without assistance from Veritas support personnel.

About GAB tunable parameters

GAB provides various configuration and tunable parameters to modify and control the behavior of the GAB module.

These tunable parameters not only provide control of the configurations like maximum possible number of nodes in the cluster, but also provide control on how GAB behaves when it encounters a fault or a failure condition. Some of these tunable parameters are needed when the GAB module is loaded into the system. Any changes to these load-time tunable parameters require either unload followed by reload of GAB module or system reboot. Other tunable parameters (run-time) can be changed while GAB module is loaded, configured, and cluster is running. Any changes to such a tunable parameter will have immediate effect on the tunable parameter values and GAB behavior.

These tunable parameters are defined in the following file:

`/etc/sysconfig/gab`

See "About GAB load-time or static tunable parameters" on page 220.

See "About GAB run-time or dynamic tunable parameters" on page 222.

About GAB load-time or static tunable parameters

Table 5-1 lists the static tunable parameters in GAB that are used during module load time. Use the `gabconfig -e` command to list all such GAB tunable parameters.

You can modify these tunable parameters only by adding new values in the GAB configuration file. The changes take effect only on reboot or on reload of the GAB module.

Table 5-1 GAB static tunable parameters

GAB parameter	Description	Values (default and range)
numnids	Maximum number of nodes in the cluster	Default: 128 Range: 1-128
numports	Maximum number of ports in the cluster	Default: 32 Range: 1-32

Table 5-1 GAB static tunable parameters *(continued)*

GAB parameter	Description	Values (default and range)
flowctrl	Number of pending messages in GAB queues (send or receive) before GAB hits flow control. This can be overwritten while cluster is up and running with the `gabconfig -Q` option. Use the `gabconfig` command to control value of this tunable.	Default: 128 Range: 1-1024
logbufsize	GAB internal log buffer size in bytes	Default: 48100 Range: 8100-65400
msglogsize	Maximum messages in internal message log	Default: 256 Range: 128-4096
isolate_time	Maximum time to wait for isolated client Can be overridden at runtime See "About GAB run-time or dynamic tunable parameters" on page 222.	Default: 120000 msec (2 minutes) Range: 160000-240000 (in msec)
kill_ntries	Number of times to attempt to kill client Can be overridden at runtime See "About GAB run-time or dynamic tunable parameters" on page 222.	Default: 5 Range: 3-10
conn_wait	Maximum number of wait periods (as defined in the stable timeout parameter) before GAB disconnects the node from the cluster during cluster reconfiguration	Default: 12 Range: 1-256

Table 5-1 GAB static tunable parameters *(continued)*

GAB parameter	Description	Values (default and range)
ibuf_count	Determines whether the GAB logging daemon is enabled or disabled The GAB logging daemon is enabled by default. To disable, change the value of gab_ibuf_count to 0. The disable login to the gab daemon while cluster is up and running with the `gabconfig -K` option. Use the `gabconfig` command to control value of this tunable.	Default: 8 Range: 0-32
kstat_size	Number of system statistics to maintain in GAB	Default: 60 Range: 0 - 240

About GAB run-time or dynamic tunable parameters

You can change the GAB dynamic tunable parameters while GAB is configured and while the cluster is running. The changes take effect immediately on running the `gabconfig` command. Note that some of these parameters also control how GAB behaves when it encounters a fault or a failure condition. Some of these conditions can trigger a PANIC which is aimed at preventing data corruption.

You can display the default values using the `gabconfig -l` command. To make changes to these values persistent across reboots, you can append the appropriate command options to the `/etc/gabtab` file along with any existing options. For example, you can add the `-k` option to an existing `/etc/gabtab` file that might read as follows:

```
gabconfig -c -n4
```

After adding the option, the `/etc/gabtab` file looks similar to the following:

```
gabconfig -c -n4 -k
```

Table 5-2 describes the GAB dynamic tunable parameters as seen with the `gabconfig -l` command, and specifies the command to modify them.

Table 5-2 GAB dynamic tunable parameters

GAB parameter	Description and command

Control port seed	This option defines the minimum number of nodes that can form the cluster. This option controls the forming of the cluster. If the number of nodes in the cluster is less than the number specified in the `gabtab` file, then the cluster will not form. For example: if you type `gabconfig -c -n4`, then the cluster will not form until all four nodes join the cluster. If this option is enabled using the `gabconfig -x` command then the node will join the cluster even if the other nodes in the cluster are not yet part of the membership.
	
	Use the following command to set the number of nodes that can form the cluster:
	
	`gabconfig -n count`
	
	Use the following command to enable control port seed. Node can form the cluster without waiting for other nodes for membership:
	
	`gabconfig -x`

Halt on process death	Default: Disabled
	
	This option controls GAB's ability to halt (panic) the system on user process death. If _had and _hashadow are killed using `kill -9`, the system can potentially lose high availability. If you enable this option, then the GAB will PANIC the system on detecting the death of the client process. The default behavior is to disable this option.
	
	Use the following command to enable halt system on process death:
	
	`gabconfig -p`
	
	Use the following command to disable halt system on process death:
	
	`gabconfig -P`

Table 5-2 GAB dynamic tunable parameters *(continued)*

GAB parameter	Description and command

Missed heartbeat halt	Default: Disabled
	
	If this option is enabled then the system will panic on missing the first heartbeat from the VCS engine or the `vxconfigd` daemon in a CVM environment. The default option is to disable the immediate panic.
	
	This GAB option controls whether GAB can panic the node or not when the VCS engine or the vxconfigd daemon miss to heartbeat with GAB. If the VCS engine experiences a hang and is unable to heartbeat with GAB, then GAB will NOT PANIC the system immediately. GAB will first try to abort the process by sending SIGABRT (kill_ntries - default value 5 times) times after an interval of "iofence_timeout" (default value 15 seconds). If this fails, then GAB will wait for the "isolate timeout" period which is controlled by a global tunable called isolate_time (default value 2 minutes). If the process is still alive, then GAB will PANIC the system.
	
	If this option is enabled GAB will immediately HALT the system in case of missed heartbeat from client.
	
	Use the following command to enable system halt when process heartbeat fails:
	
	`gabconfig -b`
	
	Use the following command to disable system halt when process heartbeat fails:
	
	`gabconfig -B`

Table 5-2 GAB dynamic tunable parameters *(continued)*

GAB parameter	Description and command

Halt on rejoin	Default: Disabled
	
	This option allows the user to configure the behavior of the VCS engine or any other user process when one or more nodes rejoin a cluster after a network partition. By default GAB will not PANIC the node running the VCS engine. GAB kills the userland process (the VCS engine or the vxconfigd process). This recycles the user port (port h in case of the VCS engine) and clears up messages with the old generation number programmatically. Restart of the process, if required, must be handled outside of GAB control, e.g., for hashadow process restarts _had.
	
	When GAB has kernel clients (such as fencing, VxVM, or VxFS), then the node will always PANIC when it rejoins the cluster after a network partition. The PANIC is mandatory since this is the only way GAB can clear ports and remove old messages.
	
	Use the following command to enable system halt on rejoin:
	
	`gabconfig -j`
	
	Use the following command to disable system halt on rejoin:
	
	`gabconfig -J`

Keep on killing	Default: Disabled If this option is enabled, then GAB prevents the system from PANICKING when the VCS engine or the vxconfigd process fail to heartbeat with GAB and GAB fails to kill the VCS engine or the vxconfigd process. GAB will try to continuously kill the VCS engine and will not panic if the kill fails. Repeat attempts to kill process if it does not die `gabconfig -k`

Table 5-2 GAB dynamic tunable parameters *(continued)*

GAB parameter	Description and command

Quorum flag	Default: Disabled
	This is an option in GAB which allows a node to IOFENCE (resulting in a PANIC) if the new membership set is < 50% of the old membership set. This option is typically disabled and is used when integrating with other products
	Enable iofence quorum
	`gabconfig -q`
	Disable iofence quorum
	`gabconfig -d`

GAB queue limit	Default: Send queue limit: 128
	Default: Recv queue limit: 128
	GAB queue limit option controls the number of pending message before which GAB sets flow. Send queue limit controls the number of pending message in GAB send queue. Once GAB reaches this limit it will set flow control for the sender process of the GAB client. GAB receive queue limit controls the number of pending message in GAB receive queue before GAB send flow control for the receive side.
	Set the send queue limit to specified value
	`gabconfig -Q sendq:value`
	Set the receive queue limit to specified value
	`gabconfig -Q recvq:value`

IOFENCE timeout	Default: 15000(ms) This parameter specifies the timeout (in milliseconds) for which GAB will wait for the clients to respond to an IOFENCE message before taking next action. Based on the value of kill_ntries , GAB will attempt to kill client process by sending SIGABRT signal. If the client process is still registered after GAB attempted to kill client process for the value of kill_ntries times, GAB will halt the system after waiting for additional isolate_timeout value. Set the iofence timeout value to specified value in milliseconds. `gabconfig -f value`

Table 5-2 GAB dynamic tunable parameters *(continued)*

GAB parameter	Description and command
Stable timeout	Default: 5000(ms) Specifies the time GAB waits to reconfigure membership after the last report from LLT of a change in the state of local node connections for a given port. Any change in the state of connections will restart GAB waiting period. Set the stable timeout to specified value `gabconfig -t stable`
Isolate timeout	Default: 120000(ms) This tunable specifies the timeout value for which GAB will wait for client process to unregister in response to GAB sending SIGKILL signal. If the process still exists after isolate timeout GAB will halt the system `gabconfig -S isolate_time:value`
Kill_ntries	Default: 5 This tunable specifies the number of attempts GAB will make to kill the process by sending SIGABRT signal. `gabconfig -S kill_ntries:value`
Driver state	This parameter shows whether GAB is configured. GAB may not have seeded and formed any membership yet.
Partition arbitration	This parameter shows whether GAB is asked to specifically ignore jeopardy. See the `gabconfig` (1M) manual page for details on the `-s` flag.

About LLT tunable parameters

LLT provides various configuration and tunable parameters to modify and control the behavior of the LLT module. This section describes some of the LLT tunable parameters that can be changed at run-time and at LLT start-time.

See "About Low Latency Transport (LLT)" on page 21.

The tunable parameters are classified into two categories:

- LLT timer tunable parameters

See “About LLT timer tunable parameters” on page 228.

- LLT flow control tunable parameters
  See “About LLT flow control tunable parameters” on page 232.

See “Setting LLT timer tunable parameters” on page 235.

About LLT timer tunable parameters

Table 5-3 lists the LLT timer tunable parameters. The timer values are set in .01 sec units. The command `lltconfig -T query` can be used to display current timer values.

Table 5-3 LLT timer tunable parameters

LLT parameter	Description	Default	When to change	Dependency with other LLT tunable parameters
peerinact	LLT marks a link of a peer node as “inactive," if it does not receive any packet on that link for this timer interval. Once a link is marked as "inactive," LLT will not send any data on that link.	1600	■ Change this value for delaying or speeding up node/link inactive notification mechanism as per client’s notification processing logic. ■ Increase the value for planned replacement of faulty network cable /switch. ■ In some circumstances, when the private networks links are very slow or the network traffic becomes very bursty, increase this value so as to avoid false notifications of peer death. Set the value to a high value for planned replacement of faulty network cable or faulty switch.	The timer value should always be higher than the peertrouble timer value.

Table 5-3 LLT timer tunable parameters *(continued)*

LLT parameter	Description	Default	When to change	Dependency with other LLT tunable parameters
rpeerinact	Mark RDMA channel of a RDMA link as "inactive", if the node does not receive any packet on that link for this timer interval. Once RDMA channel is marked as "inactive", LLT does not send any data on the RDMA channel of that link, however, it may continue to send data over non-RDMA channel of that link until peerinact expires. You can view the status of the RDMA channel of a RDMA link using lltstat -nvv -r command. This parameter is supported only on selected versions of Linux.	700	Decrease the value of this tunable for speeding up the RDMA link failure recovery. If the links are unstable, and they are going up and down frequently then do not decrease this value.	This timer value should always be greater than peertrouble timer value and less than peerinact value.
peertrouble	LLT marks a high-pri link of a peer node as "troubled", if it does not receive any packet on that link for this timer interval. Once a link is marked as "troubled", LLT will not send any data on that link till the link is up.	200	■ In some circumstances, when the private networks links are very slow or nodes in the cluster are very busy, increase the value. ■ Increase the value for planned replacement of faulty network cable /faulty switch.	This timer value should always be lower than peerinact timer value. Also, It should be close to its default value.
peertroublelo	LLT marks a low-pri link of a peer node as "troubled", if it does not receive any packet on that link for this timer interval. Once a link is marked as "troubled", LLT will not send any data on that link till the link is available.	400	■ In some circumstances, when the private networks links are very slow or nodes in the cluster are very busy, increase the value. ■ Increase the value for planned replacement of faulty network cable /faulty switch.	This timer value should always be lower than peerinact timer value. Also, It should be close to its default value.

Table 5-3 LLT timer tunable parameters *(continued)*

LLT parameter	Description	Default	When to change	Dependency with other LLT tunable parameters
heartbeat	LLT sends heartbeat packets repeatedly to peer nodes after every heartbeat timer interval on each highpri link.	50	In some circumstances, when the private networks links are very slow (or congested) or nodes in the cluster are very busy, increase the value.	This timer value should be lower than peertrouble timer value. Also, it should not be close to peertrouble timer value.
heartbeatlo	LLT sends heartbeat packets repeatedly to peer nodes after every heartbeatlo timer interval on each low pri link.	100	In some circumstances, when the networks links are very slow or nodes in the cluster are very busy, increase the value.	This timer value should be lower than peertroublelo timer value. Also, it should not be close to peertroublelo timer value.
timetoreqhb	If LLT does not receive any packet from the peer node on a particular link for "timetoreqhb" time period, it attempts to request heartbeats (sends 5 special heartbeat requests (hbreqs) to the peer node on the same link) from the peer node. If the peer node does not respond to the special heartbeat requests, LLT marks the link as “expired” for that peer node. The value can be set from the range of 0 to (peerinact -200). The value 0 disables the request heartbeat mechanism.	1400	Decrease the value of this tunable for speeding up node/link inactive notification mechanism as per client’s notification processing logic. Disable the request heartbeat mechanism by setting the value of this timer to 0 for planned replacement of faulty network cable /switch. In some circumstances, when the private networks links are very slow or the network traffic becomes very bursty, don’t change the value of this timer tunable.	This timer is set to ‘peerinact - 200’ automatically every time when the peerinact timer is changed.
reqhbtime	This value specifies the time interval between two successive special heartbeat requests. See the timetoreqhb parameter for more information on special heartbeat requests.	40	Veritas recommends that you do not change this value.	Not applicable

Table 5-3 LLT timer tunable parameters *(continued)*

LLT parameter	Description	Default	When to change	Dependency with other LLT tunable parameters
timetosendhb	LLT sends out of timer context heartbeats to keep the node alive when LLT timer does not run at regular interval. This option specifies the amount of time to wait before sending a heartbeat in case of timer not running. If this timer tunable is set to 0, the out of timer context heartbeating mechanism is disabled.	200	Disable the out of timer context heart-beating mechanism by setting the value of this timer to 0 for planned replacement of faulty network cable /switch. In some circumstances, when the private networks links are very slow or nodes in the cluster are very busy, increase the value	This timer value should not be more than peerinact timer value. Also, it should not be close to the peerinact timer value.
sendhbcap	This value specifies the maximum time for which LLT will send contiguous out of timer context heartbeats.	18000	Veritas recommends that you do not change this value.	NA
oos	If the out-of-sequence timer has expired for a node, LLT sends an appropriate NAK to that node. LLT does not send a NAK as soon as it receives an oos packet. It waits for the oos timer value before sending the NAK.	10	Do not change this value for performance reasons. Lowering the value can result in unnecessary retransmissions/negative acknowledgement traffic. You can increase the value of oos if the round trip time is large in the cluster (for example, campus cluster).	Not applicable
retrans	LLT retransmits a packet if it does not receive its acknowledgement for this timer interval value.	10	Do not change this value. Lowering the value can result in unnecessary retransmissions. You can increase the value of retrans if the round trip time is large in the cluster (for example, campus cluster).	Not applicable
service	LLT calls its service routine (which delivers messages to LLT clients) after every service timer interval.	100	Do not change this value for performance reasons.	Not applicable

Table 5-3 LLT timer tunable parameters *(continued)*

LLT parameter	Description	Default	When to change	Dependency with other LLT tunable parameters
arp	LLT flushes stored address of peer nodes when this timer expires and relearns the addresses.	0	This feature is disabled by default.	Not applicable
arpreq	LLT sends an arp request when this timer expires to detect other peer nodes in the cluster.	3000	Do not change this value for performance reasons.	Not applicable

About LLT flow control tunable parameters

Table 5-4 lists the LLT flow control tunable parameters. The flow control values are set in number of packets. The command `lltconfig -F query` can be used to display current flow control settings.

Table 5-4 LLT flow control tunable parameters

LLT parameter	Description	Default	When to change	Dependency with other LLT tunable parameters
highwater	When the number of packets in transmit queue for a node reaches highwater, LLT is flow controlled.	200	If a client generates data in bursty manner, increase this value to match the incoming data rate. Note that increasing the value means more memory consumption so set an appropriate value to avoid wasting memory unnecessarily. Lowering the value can result in unnecessary flow controlling the client.	This flow control value should always be higher than the lowwater flow control value.
lowwater	When LLT has flow controlled the client, it will not start accepting packets again till the number of packets in the port transmit queue for a node drops to lowwater.	100	Veritas does not recommend to change this tunable.	This flow control value should be lower than the highwater flow control value. The value should not be close the highwater flow control value.

Table 5-4 LLT flow control tunable parameters *(continued)*

LLT parameter	Description	Default	When to change	Dependency with other LLT tunable parameters
rporthighwater	When the number of packets in the receive queue for a port reaches highwater, LLT is flow controlled.	200	If a client generates data in bursty manner, increase this value to match the incoming data rate. Note that increasing the value means more memory consumption so set an appropriate value to avoid wasting memory unnecessarily. Lowering the value can result in unnecessary flow controlling the client on peer node.	This flow control value should always be higher than the rportlowwater flow control value.
rportlowwater	When LLT has flow controlled the client on peer node, it will not start accepting packets for that client again till the number of packets in the port receive queue for the port drops to rportlowwater.	100	Veritas does not recommend to change this tunable.	This flow control value should be lower than the rpothighwater flow control value. The value should not be close the rporthighwater flow control value.

Table 5-4 LLT flow control tunable parameters *(continued)*

LLT parameter	Description	Default	When to change	Dependency with other LLT tunable parameters
window	This is the maximum number of un-ACKed packets LLT will put in flight.	50	For performance reason, the adaptive window feature is enabled, by default for port 5 (cfs) and port 24(cvm). You can manually enable adaptive window for other ports by changing the value of the LLT_AW_PORT_LIST parameter in the /etc/sysconfig/llt file. ■ To disable the adaptive window, change the value of the LLT_ENABLE_AWINDOW parameter to zero. ■ To enable adaptive window for ports other than 5 and 24, add the port numbers in LLT_AW_PORT_LIST separated by comma. For example: LLT_AW_PORT_LIST=: '"5,24,0,1,5,14"'. ■ To set window size for ports other than those listed in the LLT_AW_PORT_LIST parameter, change the value of the window parameter as per the private network speed. Example: set-flow window: 2000 Change the value as per the private networks speed. Lowering the value irrespective of network speed may result in unnecessary retransmission of out of window sequence packets.	This flow control value should not be higher than the difference between the highwater flow control value and the lowwater flow control value. The value of this parameter (window) should be aligned with the value of the bandwidth delay product.

Table 5-4 LLT flow control tunable parameters *(continued)*

LLT parameter	Description	Default	When to change	Dependency with other LLT tunable parameters
linkburst	It represents the number of back-to-back packets that LLT sends on a link before the next link is chosen.	32	For performance reasons, its value should be either 0 or at least 32.	This flow control value should not be higher than the difference between the highwater flow control value and the lowwater flow control value.
ackval	LLT sends acknowledgement of a packet by piggybacking an ACK packet on the next outbound data packet to the sender node. If there are no data packets on which to piggyback the ACK packet, LLT waits for ackval number of packets before sending an explicit ACK to the sender.	10	Do not change this value for performance reasons. Increasing the value can result in unnecessary retransmissions.	Not applicable
sws	To avoid Silly Window Syndrome, LLT transmits more packets only when the count of un-acked packet goes to below of this tunable value.	40	For performance reason, its value should be changed whenever the value of the window tunable is changed as per the formula given below: sws = window *4/5.	Its value should be lower than that of window. Its value should be close to the value of window tunable.
largepktlen	When LLT has packets to delivers to multiple ports, LLT delivers one large packet or up to five small packets to a port at a time. This parameter specifies the size of the large packet.	1024	Veritas does not recommend to change this tunable.	Not applicable

Setting LLT timer tunable parameters

You can set the LLT tunable parameters either with the `lltconfig` command or in the `/etc/llttab` file. You can use the `lltconfig` command to change a parameter on the local node at run time. Veritas recommends you run the command on all the nodes in the cluster to change the values of the parameters. To set an LLT parameter across system reboots, you must include the parameter definition in the `/etc/llttab` file. Default values of the parameters are taken if nothing is specified in /etc/llttab. The parameters values specified in the /etc/llttab file come into effect

at LLT start-time only. Veritas recommends that you specify the same definition of the tunable parameters in the `/etc/lltab` file of each node.

To get and set a timer tunable:

- To get the current list of timer tunable parameters using lltconfig command:

```
# lltconfig -T query
```

- To set a timer tunable parameter using the `lltconfig` command:

```
# lltconfig -T timer tunable:value
```

- To set a timer tunable parameter in the `/etc/lltab` file:

```
set-timer  timer tunable:value
```

To get and set a flow control tunable

- To get the current list of flow control tunable parameters using lltconfig command:

```
# lltconfig -F query
```

- To set a flow control tunable parameter using the `lltconfig` command:

```
# lltconfig -F flowcontrol tunable:value
```

- To set a flow control tunable parameter in the `/etc/lltab` file:

```
set-flow  flowcontrol tunable:value
```

See the `lltconfig`(1M) and `lltab`(1M) manual pages.

About VXFEN tunable parameters

The section describes the VXFEN tunable parameters and how to reconfigure the VXFEN module.

Table 5-5 describes the tunable parameters for the VXFEN driver.

Table 5-5 VXFEN tunable parameters

vxfen Parameter	Description and Values: Default, Minimum, and Maximum
dbg_log_size	Size of debug log in bytes ■ Values Default: 524288(512KB) Minimum: 65536(64KB) Maximum: 1048576(1MB)
vxfen_max_delay	Specifies the maximum number of seconds that the smaller sub-cluster waits before racing with larger sub-clusters for control of the coordinator disks when a network partition occurs. This value must be greater than the vxfen_min_delay value. ■ Values Default: 60 Minimum: 1 Maximum: 600
vxfen_min_delay	Specifies the minimum number of seconds that the smaller sub-cluster waits before racing with larger sub-clusters for control of the coordinator disks when a network partition occurs. This value must be smaller than or equal to the vxfen_max_delay value. ■ Values Default: 1 Minimum: 1 Maximum: 600
vxfen_vxfnd_tmt	Specifies the time in seconds that the I/O fencing driver VxFEN waits for the I/O fencing daemon VXFEND to return after completing a given task. ■ Values Default: 60 Minimum: 10 Maximum: 600

Table 5-5 VXFEN tunable parameters *(continued)*

vxfen Parameter	Description and Values: Default, Minimum, and Maximum
panic_timeout_offst	Specifies the time in seconds based on which the I/O fencing driver VxFEN computes the delay to pass to the GAB module to wait until fencing completes its arbitration before GAB implements its decision in the event of a split-brain. You can set this parameter in the `vxfenmode` file and use the `vxfenadm` command to check the value. Depending on the vxfen_mode, the GAB delay is calculated as follows: ■ For scsi3 mode: 1000 * (panic_timeout_offst + vxfen_max_delay) ■ For customized mode: 1000 * (panic_timeout_offst + max (vxfen_vxfnd_tmt, vxfen_loser_exit_delay)) ■ Default: 10

In the event of a network partition, the smaller sub-cluster delays before racing for the coordinator disks. The time delay allows a larger sub-cluster to win the race for the coordinator disks. The vxfen_max_delay and vxfen_min_delay parameters define the delay in seconds.

Configuring the VXFEN module parameters

After adjusting the tunable kernel driver parameters, you must reconfigure the VXFEN module for the parameter changes to take effect.

The following example procedure changes the value of the vxfen_min_delay parameter.

On each Linux node, edit the file /etc/sysconfig/vxfen to change the value of the vxfen driver tunable global parameters, vxfen_max_delay and vxfen_min_delay.

Note: You must restart the VXFEN module to put any parameter change into effect.

To configure the VxFEN parameters and reconfigure the VxFEN module

1 Shut down all Oracle service groups on the node.

```
# hagrp -offline oragrp -sys  sys1
```

2 Stop all Oracle client processes, such as sqlplus, svrmgrl, and gsd, on the node.

3. Stop all the applications that are not configured under VCS. Use native application commands to stop the application.
4. Stop VCS on all the nodes. Run the following command on each node:

   ```
   # hastop -local
   ```

5. Stop the VxFEN driver.

   For RHEL 7, SLES 12, and supported RHEL distributions:

   ```
   # systemctl stop vxfen
   ```

   For earlier versions of RHEL, SLES, and supported RHEL distributions:

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

6. Edit the /etc/sysconfig/vxfen file.

   For example, change the entry from:

   ```
   vxfen_min_delay=1
   ```

   to:

   ```
   vxfen_min_delay=30
   ```

7. Start the VXFEN module.

   For RHEL 7, SLES 12, and supported RHEL distributions:

   ```
   # systemctl start vxfen
   ```

   For earlier versions of RHEL, SLES, and supported RHEL distributions:

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

8. Bring the service groups online.

   ```
   # hagrp -online oragrp -sys  sys1
   ```

9. Start all the applications that are not configured under VCS. Use native application commands to start the applications.
10. Start VCS.

    ```
    #  hastart
    ```

Tuning guidelines for campus clusters

An important consideration while tuning an SF Oracle RAC campus cluster is setting the LLT peerinact time. Follow the guidelines below to determine the optimum value of peerinact time:

- Calculate the roundtrip time using lltping (1M).
- Evaluate LLT heartbeat time as half of the round trip time.
- Set the LLT peer trouble time as 2-4 times the heartbeat time.
- LLT peerinact time should be set to be more than 4 times the heart beat time.

Section 3

Reference

Appendix A

List of SF Oracle RAC health checks

This appendix includes the following topics:

LLT health checks

This section lists the health checks performed for LLT, the messages displayed for each check, and a brief description of the check.

Note: Warning messages indicate issues in the components or the general health of the cluster. Follow the troubleshooting recommendations to resolve the issues.

See "Troubleshooting LLT health check warning messages" on page 184.

Table A-1 lists the health checks performed for LLT.

Table A-1 List of health checks for LLT

List of health checks	Message	Description
LLT timer subsystem scheduling check	`Warning: OS timer is not called for num seconds`	Checks whether the LLT module runs in accordance with the scheduled operating system timer. The message indicates that the operating system timer is not called for the specified interval. The parameter `timer_threshold` contains the optimum threshold for this check.
OS memory availability check for the packet transmission	`Warning: Kernel failed to allocate memory num time(s)`	Checks whether the kernel has allocated sufficient memory to LLT for cluster communication. The message indicates that the kernel attempts at allocating the requisite memory has failed (*num*) times. The parameter `no_mem_to_xmit_allow` contains the optimum threshold for this check.
Flow control status monitoring	`Flow-control occurred num time(s) and back-enabled num time(s) on port port number for node node number`	Checks whether LLT has sufficient bandwidth to accept incoming data packets and transmit the data packets. Flow control also depends on the peer nodes' ability to accept incoming data traffic. The message indicates that packet transmission and reception was controlled (*num*) and normalized (*num*) times. The parameter `max_canput` contains the optimum threshold for this check.

Table A-1 List of health checks for LLT *(continued)*

List of health checks	Message	Description
Flaky link monitoring	`Warning: Connectivity with node node id on link link id is flaky num time(s).` `The distribution is: (0-4 s) <num> (4-8 s) <num> (8-12 s) <num> (12-16 s) <num> (>=16 s)`	Checks whether the private interconnects are stable. The message indicates that connectivity (*link*) with the peer node (*node id*) is monitored (*num*) times within a stipulated duration (for example, 0-4 seconds). The parameter `flakylink_allow` contains the optimum threshold for this check.
Node status check	`One or more link connectivity with peer node(s) node name is in trouble.`	Checks the current node membership. The message indicates connectivity issues with the peer node (*node name*). This check does not require a threshold.
Link status check	`Link connectivity with node name is on only one link. Veritas recommends configuring a minimum of 2 links.`	Checks the number of private interconnects that are currently active. The message indicates that only one private interconnect exists or is operational between the local node and the peer node *node id*. This check does not require a threshold.
Connectivity check	`Only one link is configured under LLT. Veritas recommends configuring a minimum of 2 links.`	Checks the number of private interconnects that were configured under LLT during configuration. The message indicates that only one private interconnect exists or is configured under LLT. This check does not require a threshold.

Table A-1 List of health checks for LLT *(continued)*

List of health checks	Message	Description

LLT packet related checks	`Retransmitted % percentage of total transmitted packets.`
	`Sent % percentage of total transmitted packet when no link is up.`
	`% percentage of total received packets are with bad checksum.`
	`% percentage of total received packets are out of window.`
	`% percentage of total received packets are misaligned.`
		Checks whether the data packets transmitted by LLT reach peer nodes without any error. If there is an error in packet transmission, it indicates an error in the private interconnects.
		The message indicates the percentage of data packets transmitted or received and the associated transmission errors, such as bad checksum and failed link.
		The following parameters contain the optimum thresholds for this check: `retrans_pct`, `send_nolinkup_pct`, `recv_oow_pct`, `recv_misaligned_pct`, `recv_badcksum_pct`

DLPI layer related checks	`% percentage of total received packets are with DLPI error.`	Checks whether the data link layer (DLP) is causing errors in packet transmission. The message indicates that some of the received packets (%) contain DLPI error. You can set a desired percentage value in the configuration file. The parameter `recv_dlpierror_pct` contains the optimum threshold for this check.
Traffic distribution over the links	`Traffic distribution over links: %% Send data on link`*`num percentage`* `%% Recv data on link`*`num percentage`*	Checks the distribution of traffic over all the links configured under LLT. The message displays the percentage of data (%) sent and recd on a particular link (*num*) This check does not require a threshold.

Table A-1 List of health checks for LLT *(continued)*

List of health checks	Message	Description
LLT Ports status check	`% per of total transmitted packets are with large xmit latency (>16ms) for port port id` `%per received packets are with large recv latency (>16ms) for port port id.`	Checks the latency period for transmitting or receiving packets. The message indicates that some percentage (%) of the transmitted/received packets exceed the stipulated latency time. The following parameters contain the optimum thresholds for this check: `hirecvlatencycnt_pct`, `hixmitlatencycnt_pct`.
System load monitoring	`Load Information: Average : num,num,num`	Monitors the system workload at the stipulated periodicity (1 second, 5 seconds, 15 seconds) This check does not require a threshold.

I/O fencing health checks

This section lists the health checks performed for I/O fencing, the messages displayed for each check, and a brief description of the check.

Note: Warning messages indicate issues in the components or the general health of the cluster.

For recommendations on resolving the issues, see the troubleshooting chapter in this document.

Table A-2 lists the health checks performed for I/O fencing.

Table A-2 List of health checks for I/O fencing

List of health checks	Message	Description
Checks related to the working status of VxFEN	`VxFEN is not configured in SCSI3 mode.` `VxFEN is not running on all nodes in the cluster.`	Checks whether fencing is configured in SCSI3 mode and running on all nodes in the cluster.

Checks related to coordinator disk health	`VxFEN is running with only one coordinator disk. Loss of this disk will prevent cluster reconfiguration on loss of a node. Veritas recommends configuring a minimum of 3 coordinator disks.`
	`VxFEN is running with even number of coordinator disks. There must be odd number of coordinator disks.`
	`Replace the disk` *`disk name`* `using OCDR procedure.`
		Checks how many coordinator disks are configured for fencing and the status of the disks.

Verify the keys on the coordinator disks	`The coordinator disk (`*`disk name`*`) does not have the required key for the local node.`	Checks whether the coordinator disks configured under I/O fencing has the key of the current node.
Verify SCSI3 write-exclusive reservation on shared disk	`SCSI3 write-exclusive reservation is missing on shared disk (`*`disk_name`*`)`	Checks if the shared disk is accessible to the node for write operations.

PrivNIC health checks in SF Oracle RAC clusters

This section lists the health checks performed for PrivNIC, the messages displayed for each check, and a brief description of the check.

Note: Warning messages indicate issues in the components or the general health of the cluster.

For recommendations on resolving the issues, see the troubleshooting chapter in this document.

Table A-3 lists the health checks performed for PrivNIC.

Table A-3 List of health checks for PrivNIC

List of health checks	Message	Description
Checks related to the working status of VCS	`VCS is not running on the local system.`	Checks whether VCS is running or not.
Checks related to the status of the PrivNIC resources	If the PrivNIC resource is not online: `The PrivNIC resource` *`name`* `is not online.`	Checks whether the PrivNIC resources configured under VCS are online.
Compare the NICs used by PrivNIC with the NICs configured under LLT	For PrivNIC resources: `Mismatch between LLT links` *`llt nics`* `and PrivNIC links` *`private nics`*`.`	Checks whether the NICs used by the PrivNIC resource have the same interface (*private nics*) as those configured as LLT links (*llt nics*).
Cross check NICs used by PrivNIC with the NICs used by Oracle Clusterware.	For PrivNIC resources: `Mismatch between Oracle Clusterware links` *`crs nics`* `and PrivNIC links` *`private nics`*`.`	Checks whether the private interconnect used for Oracle Clusterware is the same as the NIC configured under the PrivNIC resource.

Oracle Clusterware health checks in SF Oracle RAC clusters

This section lists the health checks performed for Oracle Clusterware, the messages displayed for each check, and a brief description of the check.

Note: Warning messages indicate issues in the components or the general health of the cluster.

For recommendations on resolving the issues, see the troubleshooting chapter in this document.

Table A-4 lists the health checks performed for the Oracle Clusterware module.

Table A-4 List of health checks for the Oracle Clusterware module

List of health checks	Message	Description
Verify the working status of CRS	`Oracle Clusterware is not running.`	Checks whether Oracle Clusterware is up and running.
Verify the working status of CSSD resource	`No CSSD resource is configured under VCS.` `The CSSD resource` *`name`* `is not running.`	Checks whether the CSSD resource is configured under VCS and if the resource is online.
Compare the NICs used by CRS with the NICs configured under LLT	`Mismatch between LLT links` *`llt_nic1, llt_nic2`* `and Oracle Clusterware links` *`crs_nic1, crs_nic2`*`.`	Checks whether the private interconnects used by Oracle Clusterware are the same as the LLT links (*llt nics*).
Compare the NICs used by CRS with the NICs monitored by PrivNIC	`Mismatch between Oracle Clusterware links` *`crs nics`* `and PrivNIC links` *`private nics`*`.`	Checks whether the private interconnects (*crs nics*) used by Oracle Clusterware are monitored by the PrivNIC resource (*private nics*).
Compare the nodes configured for CRS with the nodes listed by llt commands.	`Mismatch between CRS nodes` *`crs nodes`* `and LLT nodes` *`llt nodes`*`.`	Checks whether the host names configured during the Oracle Clusterware installation match the list of host names displayed on running the LLT command:`lltstat -nvv`.

CVM, CFS, and ODM health checks in SF Oracle RAC clusters

This section lists the health checks performed for CVM, CFS, and ODM, the messages displayed for each check, and a brief description of the check.

Note: Warning messages indicate issues in the components or the general health of the cluster.

For recommendations on resolving the issues, see the troubleshooting chapter in this document.

Table A-5 lists the health checks performed for CVM, CFS, and ODM modules.

Table A-5 List of health checks for CVM, CFS, and ODM

List of health checks	Message	Description
Verify CVM status	`CVM is not running`	Checks whether CVM is running in the cluster.
Verify CFS status	`CFS is not running`	Checks whether CFS is running in the cluster.
Verify ODM status	`ODM is not running`	Checks whether ODM is running in the cluster.

Appendix B

Error messages

This appendix includes the following topics:

About error messages

Error messages can be generated by the following software modules:

- Veritas Volume Manager (VxVM)
- Veritas Fencing (VXFEN) driver

VxVM error messages

Table B-1 contains VxVM error messages that are related to I/O fencing.

Table B-1 VxVM error messages for I/O fencing

Message	Explanation
`vold_vcs_getnodeid(disk_path): failed to open the vxfen device. Please make sure that the vxfen driver is installed and configured.`	The vxfen driver is not configured. Follow the instructions to set up these disks and start I/O fencing. You can then clear the faulted resources and bring the service groups online.
`vold_pgr_register(disk_path): Probably incompatible vxfen driver.`	Incompatible versions of VxVM and the vxfen driver are installed on the system. Install the proper version of SF Oracle RAC.

VXFEN driver error messages

Table B-2 contains VXFEN driver error messages. In addition to VXFEN driver error messages, informational messages can also be displayed.

See "VXFEN driver informational message" on page 252.

See "Node ejection informational messages" on page 253.

Table B-2 VXFEN driver error messages

Message	Explanation
Unable to register with coordinator disk with serial number: xxxx	This message appears when the vxfen driver is unable to register with one of the coordinator disks. The serial number of the coordinator disk that failed is displayed.
Unable to register with a majority of the coordinator disks. Dropping out of cluster.	This message appears when the vxfen driver is unable to register with a majority of the coordinator disks. The problems with the coordinator disks must be cleared before fencing can be enabled. This message is preceded with the message "VXFEN: Unable to register with coordinator disk with serial number xxxx."
There exists the potential for a preexisting split-brain. The coordinator disks list no nodes which are in the current membership. However, they, also list nodes which are not in the current membership. I/O Fencing Disabled!	This message appears when there is a preexisting split-brain in the cluster. In this case, the configuration of vxfen driver fails. Clear the split-brain before configuring vxfen driver. See "Clearing preexisting split-brain condition" on page 191.
Unable to join running cluster since cluster is currently fencing a node out of the cluster	This message appears while configuring the vxfen driver, if there is a fencing race going on in the cluster. The vxfen driver can be configured by retrying after some time (after the cluster completes the fencing).

VXFEN driver informational message

The following informational message appears when a node is ejected from the cluster to prevent data corruption when a split-brain occurs.

```
VXFEN CRITICAL V-11-1-20 Local cluster node ejected from cluster
to prevent potential data corruption
```

Node ejection informational messages

Informational messages may appear on the console of one of the cluster nodes when a node is ejected from a disk or LUN.

These informational messages can be ignored.