

Storage Foundation for Sybase ASE CE 7.4.1 Administrator's Guide - Linux

Last updated: 2019-10-17

Legal Notice

Copyright © 2019 Veritas Technologies LLC. All rights reserved.

Veritas and the Veritas Logo are trademarks or registered trademarks of Veritas Technologies LLC or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.

This product may contain third-party software for which Veritas is required to provide attribution to the third-party ("Third-Party Programs"). Some of the Third-Party Programs are available under open source or free software licenses. The License Agreement accompanying the Software does not alter any rights or obligations you may have under those open source or free software licenses. Refer to the third-party legal notices document accompanying this Veritas product or available at:

<https://www.veritas.com/about/legal/license-agreements>

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

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

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

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:

infoscaledocs@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

Chapter 1	Overview of Storage Foundation for Sybase ASE CE	8
	About Storage Foundation for Sybase ASE CE	8
	Benefits of SF Sybase CE	9
	How SF Sybase CE works (high-level perspective)	10
	About SF Sybase CE components	13
	Communication infrastructure	13
	Cluster interconnect communication channel	15
	Low-level communication: port relationship between GAB and processes	17
	Cluster Volume Manager (CVM)	18
	Cluster File System (CFS)	20
	Cluster Server (VCS)	23
	About I/O fencing in SF Sybase CE environment	26
	Sybase ASE CE components	34
	About optional features in SF Sybase CE	36
	Typical configuration of SF Sybase CE clusters in secure mode	36
	Typical configuration of VOM-managed SF Sybase CE clusters	37
	About SF Sybase CE global cluster setup for disaster recovery	38
	How the agent makes Sybase highly available	38
	About Veritas InfoScale Operations Manager	39
	About Veritas Services and Operations Readiness Tools (SORT)	39
Chapter 2	Administering SF Sybase CE and its components	41
	Administering SF Sybase CE	41
	Setting the environment variables for SF Sybase CE	42
	Starting or stopping SF Sybase CE on each node	43
	Applying operating system updates on SF Sybase CE nodes	48
	Adding storage to an SF Sybase CE cluster	49
	Recovering from storage failure	50

Enhancing the performance of SF Sybase CE clusters	51
Verifying the nodes in an SF Sybase CE cluster	51
Administering VCS	51
Viewing available Veritas device drivers	52
Starting and stopping VCS	54
Environment variables to start and stop VCS modules	54
Adding and removing LLT links	58
Configuring aggregated interfaces under LLT	60
Displaying the cluster details and LLT version for LLT links	62
Configuring destination-based load balancing for LLT	63
Enabling and disabling intelligent resource monitoring for agents manually	63
Administering the AMF kernel driver	65
Administering I/O fencing	67
About administering I/O fencing	67
About the vxfsentsthdw utility	68
About the vxfsenadm utility	76
About the vxfsenclearpre utility	81
About the vxfsenswap utility	85
Enabling or disabling the preferred fencing policy	95
About I/O fencing log files	98
Administering CVM	98
Establishing CVM cluster membership manually	99
Changing the CVM master manually	99
Importing a shared disk group manually	102
Deporting a shared disk group manually	102
Verifying if CVM is running in an SF Sybase CE cluster	102
Verifying CVM membership state	103
Verifying the state of CVM shared disk groups	103
Verifying the activation mode	104
Administering CFS	104
Adding CFS file systems to a VCS configuration	104
Uses of cfsmount to mount and cfsunmount to unmount CFS file system	105
Resizing CFS file systems	105
Verifying the status of CFS file system nodes and their mount points	105
Administering the Sybase agent	106
Sybase agent functions	106
Monitoring options for the Sybase agent	109
Using the IPC Cleanup feature for the Sybase agent	109
Configuring the service group Sybase using the command line	111

Bringing the Sybase service group online	112
Taking the Sybase service group offline	112
Modifying the Sybase service group configuration	113
Viewing the agent log for Sybase	113
Chapter 3 Troubleshooting SF Sybase CE	114
About troubleshooting SF Sybase CE	114
Gathering information from an SF Sybase CE cluster for support analysis	114
SF Sybase CE log files	117
About SF Sybase CE kernel and driver messages	119
VCS message logging	119
Troubleshooting tips	126
Restarting the installer after a failed network connection	128
Installer cannot create UUID for the cluster	128
Troubleshooting I/O fencing	129
The vxfsntshdw utility fails when SCSI TEST UNIT READY command fails	129
Node is unable to join cluster while another node is being ejected	129
System panics to prevent potential data corruption	130
Cluster ID on the I/O fencing key of coordinator disk does not match the local cluster's ID	130
Fencing startup reports preexisting split-brain	131
Registered keys are lost on the coordinator disks	133
Replacing defective disks when the cluster is offline	133
Troubleshooting Cluster Volume Manager in SF Sybase CE clusters	136
Restoring communication between host and disks after cable disconnection	136
Shared disk group cannot be imported in SF Sybase CE cluster	137
Error importing shared disk groups in SF Sybase CE cluster	137
Unable to start CVM in SF Sybase CE cluster	137
CVM group is not online after adding a node to the SF Sybase CE cluster	138
CVMVolDg not online even though CVMCluster is online in SF Sybase CE cluster	138
Shared disks not visible in SF Sybase CE cluster	139
Troubleshooting interconnects	139
Network interfaces change their names after reboot	139
Example entries for mandatory devices	139

	Troubleshooting Sybase ASE CE	140
	Sybase private networks	140
	Sybase instances under VCS control	140
	Node does not reboot	140
	Sybase instance not starting	141
Chapter 4	Prevention and recovery strategies	142
	Prevention and recovery strategies	142
	Verification of GAB ports in SF Sybase CE cluster	142
	Examining GAB seed membership	143
	Manual GAB membership seeding	144
	Evaluating VCS I/O fencing ports	145
	Verifying normal functioning of VCS I/O fencing	146
	Managing SCSI-3 PR keys in SF Sybase CE cluster	147
	Identifying a faulty coordinator LUN	148
	Starting shared volumes manually	148
	Listing all the CVM shared disks	149
	I/O Fencing kernel logs	149
Chapter 5	Tunable parameters	150
	About GAB tunable parameters	150
	About GAB load-time or static tunable parameters	151
	About GAB run-time or dynamic tunable parameters	152
	About LLT tunable parameters	157
	About LLT timer tunable parameters	158
	About LLT flow control tunable parameters	162
	Setting LLT timer tunable parameters	165
	About VXFEN tunable parameters	166
	Configuring the VXFEN module parameters	168
Appendix A	Error messages	170
	About error messages	170
	VxVM error messages	170
	VXFEN driver error messages	171
	VXFEN driver informational message	171
	Node ejection informational messages	172

Overview of Storage Foundation for Sybase ASE CE

This chapter includes the following topics:

- [About Storage Foundation for Sybase ASE CE](#)
- [How SF Sybase CE works \(high-level perspective\)](#)
- [About SF Sybase CE components](#)
- [About optional features in SF Sybase CE](#)
- [How the agent makes Sybase highly available](#)
- [About Veritas InfoScale Operations Manager](#)
- [About Veritas Services and Operations Readiness Tools \(SORT\)](#)

About Storage Foundation for Sybase ASE CE

Storage Foundation for Sybase® Adaptive Server Enterprise Cluster Edition (SF Sybase CE) by Veritas leverages proprietary storage management and high availability technologies to enable robust, manageable, and scalable deployment of Sybase ASE CE on UNIX platforms. The solution uses 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.

SF Sybase CE integrates existing Veritas storage management and clustering technologies into a flexible solution which administrators can use to:

- Create a standard toward application and database management in data centers. SF Sybase CE provides flexible support for many types of applications and databases.
- Set up an infrastructure for Sybase ASE CE that simplifies database management while fully integrating with Sybase clustering solution.
- Apply existing expertise of Veritas technologies toward this product.

The solution stack comprises the Cluster Server (VCS), Veritas Cluster Volume Manager (CVM), Veritas Cluster File System (CFS), and Storage Foundation, which includes the base Veritas Volume Manager (VxVM) and Veritas File System (VxFS).

Benefits of SF Sybase CE

SF Sybase CE provides the following benefits:

- Use of a generic clustered file system (CFS) technology or a local file system (VxFS) technology for storing and managing Sybase ASE CE installation binaries.
- Support for file system-based management. SF Sybase CE provides a generic clustered file system technology for storing and managing Sybase ASE CE data files as well as other application data.
- Use of Cluster File System (CFS) for the Sybase ASE CE quorum device.
- Support for a standardized approach toward application and database management. A single-vendor solution for the complete SF Sybase CE software stack lets you devise a standardized approach toward application and database management. Further, administrators can apply existing expertise of Veritas technologies toward SF Sybase CE.
- Easy administration and monitoring of SF Sybase CE clusters using Veritas InfoScale Operations Manager.
- Enhanced scalability and availability with access to multiple Sybase ASE CE instances per database in a cluster.
- Prevention of data corruption in split-brain scenarios with robust SCSI-3 Persistent Reservation (PR) based I/O fencing.
- Support for sharing all types of files, in addition to Sybase ASE CE database files, across nodes.
- 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 (HBAs) and Storage Area Network (SAN) switches.
- 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 node fails, clients that are attached to the failed node can reconnect to a surviving node and resume access to the shared database. Recovery after failure in the SF Sybase CE environment is far quicker than recovery for a failover database.

- Support for block-level replication using VVR.

How SF Sybase CE works (high-level perspective)

Sybase ASE Cluster Edition is a shared disk cluster implementation of Sybase's flagship enterprise database. Sybase ASE is a highly reliable, scalable, and efficient database engine used in mission critical environments such as financial markets, telecommunications networks, and healthcare. Sybase ASE CE allows multiple "instances" of the Sybase ASE database engine running on different hardware "nodes" to simultaneously access and manage a common set of databases on disks. The primary goal of such a system is to provide exceptional availability with the added benefit of some scalability for certain use cases.

In traditional environments, only one instance accesses a database at a specific time. SF Sybase CE enables all nodes to concurrently run Sybase adaptive servers and execute transactions against the same database. This software coordinates access to the shared data for each node to provide consistency and integrity. Each node adds its processing power to the cluster as a whole and can increase overall throughput or performance.

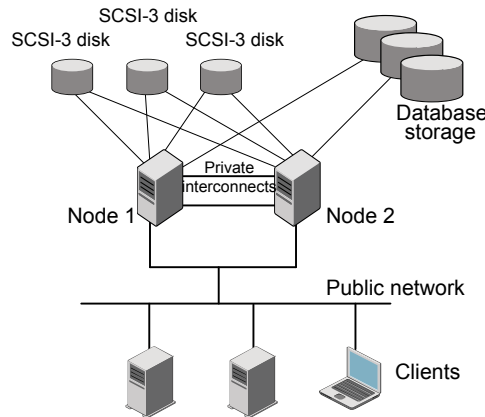
At a conceptual level, SF Sybase CE is a cluster that manages applications (instances), networking, and storage components using resources contained in service groups. SF Sybase CE clusters have the following properties:

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

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

- Nodes that form an application cluster and are connected to both the coordinator disks and databases
- Databases for storage and backup
- SCSI-3 Coordinator disks used for I/O Fencing

Figure 1-1 SF Sybase CE basic layout and components

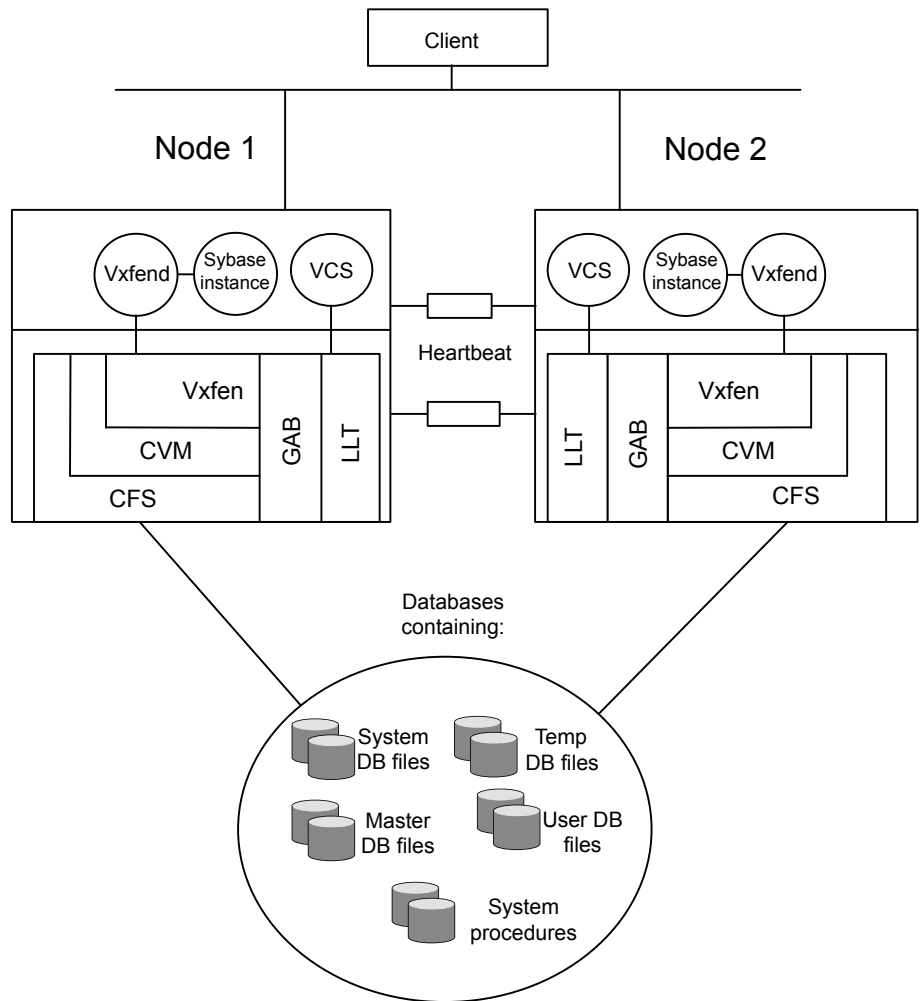


SF Sybase CE adds the following technologies to a cluster environment, which are engineered specifically to improve performance, availability, and manageability of Sybase ASE CE environments:

- Cluster File System (CFS) and cluster volume manager (CVM) technologies to manage multi-instance database access to shared storage.
- VCS for cluster management
- I/O Fencing to prevent split brain and protect data integrity
- DMP to provide increased availability and performance
- Veritas Cluster Membership Plug-in (VCMP) to provide interface between Sybase ASE CE cluster and the SF Sybase components
- The qrmutil interface to report the Sybase CE instance status

Figure 1-2 displays the technologies that make up the SF Sybase CE internal architecture.

Figure 1-2 SF Sybase CE architecture



SF Sybase CE 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 Sybase CE environment is far quicker than recovery for a failover database because another Sybase instance is already up and running.

About SF Sybase CE components

SF Sybase CE manages database instances running in parallel on multiple nodes using the following architecture and communication mechanisms to provide the infrastructure for Sybase ASE CE.

Table 1-1 SF Sybase CE 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 18.
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 20.
Cluster Server (VCS)	Uses technology from Cluster Server to manage Sybase ASE CE databases and infrastructure components. See “Cluster Server (VCS)” on page 23.
VXFEN	The VCS module prevents cluster corruption through the use of SCSI3 I/O fencing, where the vxfen mode is set to sybase.
VXFEND	The VXFEN daemon communicates directly with VCMP and relays membership modification messages.
VCMP	VCMP provides interface between Sybase cluster and the SF Sybase CE components.
QRMUTIL	QRMUTIL provides Sybase instance status.
Sybase agent	The VCS agent is responsible for bringing Sybase ASE online, taking it offline, and monitoring it.. It obtains status by checking for processes, performing SQL queries on a running database, and interacting with QRMUTIL.

See [“About Storage Foundation for Sybase ASE CE ”](#) on page 8.

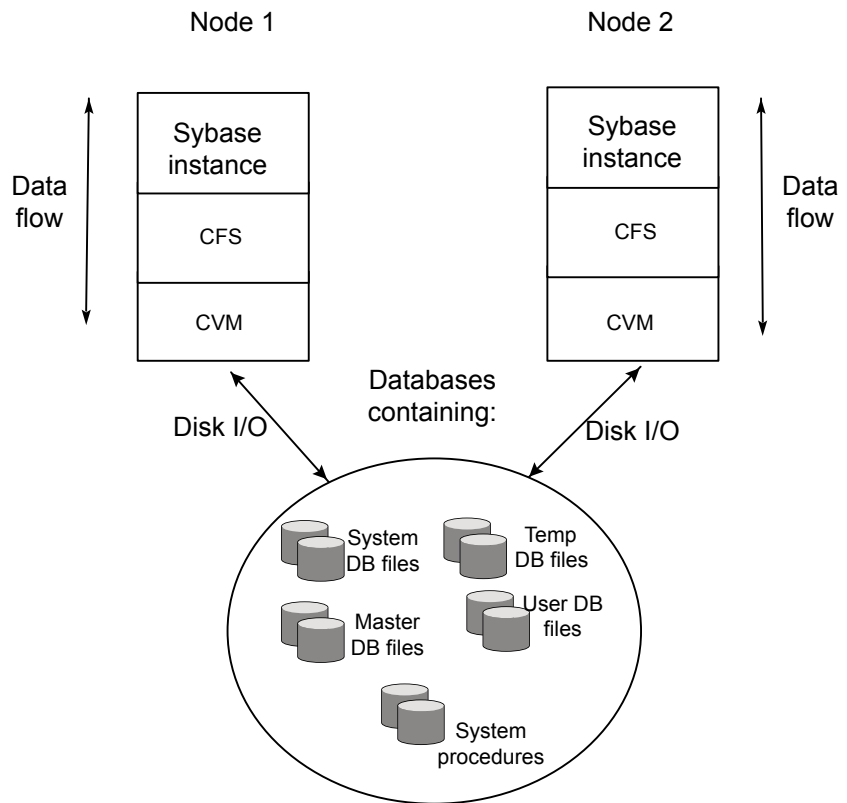
Communication infrastructure

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

Data flow

The CVM, CFS, and Sybase elements reflect the overall data flow, or data stack, from an instance running on a server to the shared storage. The various Sybase processes composing an instance read and write data to the storage through the I/O stack. Sybase writes and reads to CFS, which in turn accesses the storage through CVM.

Figure 1-3 Data stack



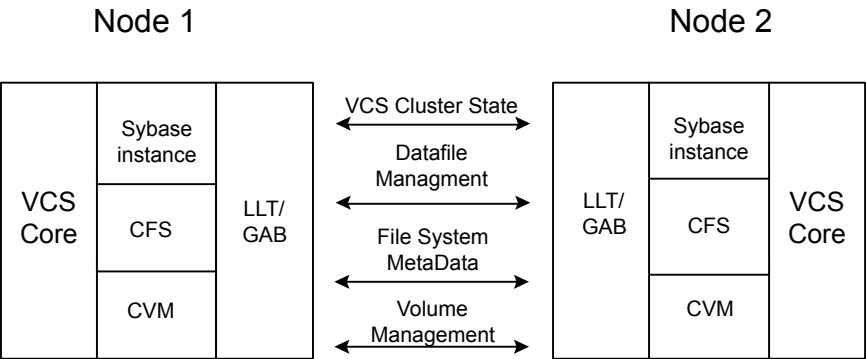
Communication requirements

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. Sybase instances must communicate to coordinate protection

of data blocks in the database. SF Sybase CE processes must communicate to coordinate data file protection and access across the cluster. CFS coordinates metadata and data 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 Sybase CE.

Low Latency Transport

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

The major functions of LLT are traffic distribution, heartbeats:

- Traffic distribution
LLT distributes (load-balances) internode communication across all available cluster interconnect links. All cluster communications are evenly distributed across as many as eight network links for performance and fault resilience. If a link fails, LLT redirects traffic to the remaining links.
- Heartbeats

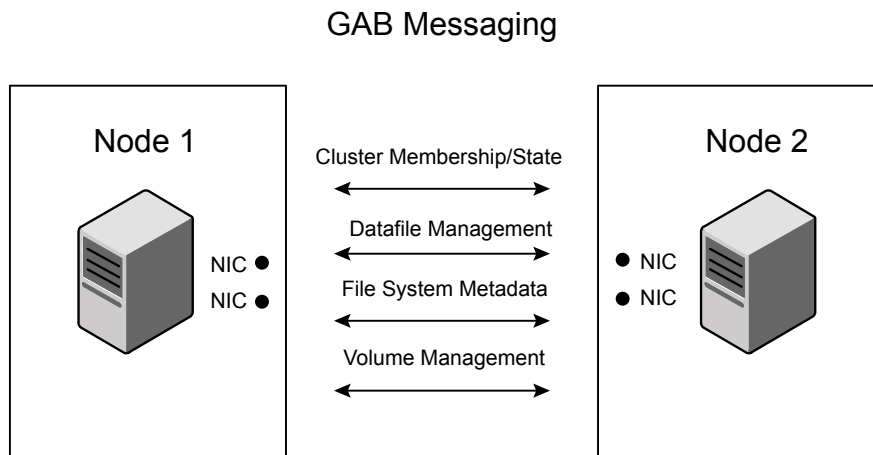
LLT is responsible for sending and receiving heartbeat traffic over network links. The Group Membership Services function of GAB uses heartbeats to determine cluster membership.

Group Membership Services/Atomic Broadcast

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

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

Figure 1-5 Cluster communication



Review the following information on cluster membership and cluster communication:

- **Cluster membership**

At a high level, all nodes configured by the installer can operate as a cluster; these nodes form a cluster membership. In SF Sybase CE, 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 Sybase CE, 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 Sybase CE uses its cluster membership capability to dynamically track the overall cluster topology.

SF Sybase CE 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.

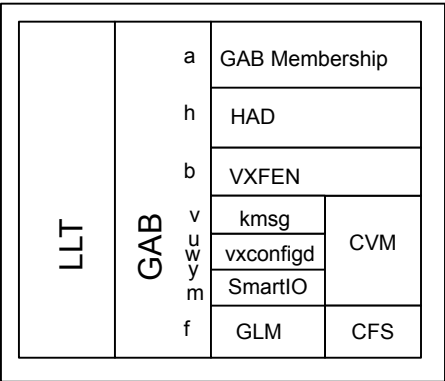
- GAB informs processes on the remaining nodes that the cluster membership has changed; this action initiates recovery actions specific to each module. For example, CVM must initiate volume recovery and CFS must perform a fast parallel file system check.
- When systems start receiving heartbeats from 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 Sybase CE 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 Sybase CE 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 1-6 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 1-6](#) on page 17.

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 1-6](#) on page 17.

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.

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.

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 Sybase CE 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 Sybase CE 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.

CFS uses port `f` for GLM lock and metadata communication.

CFS communication

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

CFS file system benefits

CFS adds such features as high availability, consistency and scalability, and centralized management to VxFS. Using CFS in an SF Sybase CE environment provides the following benefits:

- Increased manageability, including easy creation and expansion of files
without a file system, you must provide Sybase with fixed-size partitions. With CFS, you can grow file systems dynamically to meet future requirements. Use the `vxresize` command from CVM master and CFS primary to dynamically change the size of a CFS filesystem. For more information on `vxresize`, refer to the `vxresize(1)`, `fsadm_vxfs(1)` and `chfs(1)` manual pages.
- Less prone to user error
Raw partitions are not visible and administrators can compromise them by mistakenly putting file systems over the partitions.
- Data center consistency
If you have raw partitions, you are limited to a Sybase ASE CE-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.

For example, type:

```
# vxedit -g dname set group=sybase owner=sybase mode=660 \  
volume_name
```

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

Cluster Server (VCS)

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

In a typical SF Sybase CE configuration, the Sybase ASE CE 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 Sybase 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 Sybase CE provides specific agents for VCS to manage CVM, CFS, and Sybase components.

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 24.

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 25.

See [“Enabling and disabling intelligent resource monitoring for agents manually”](#) on page 63.

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, ext4, and XFS file system 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.

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 23.

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.

About I/O fencing in SF Sybase CE environment

I/O fencing is a mechanism to prevent uncoordinated access to the shared storage. This feature works even in the case of faulty cluster communications causing a split-brain condition. Veritas provides a technology called I/O fencing to remove the risk associated with split-brain. I/O fencing allows write access for members of the active cluster and blocks access to storage from non-members; even a node that is alive is unable to cause damage.

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

Fencing involves coordinator disks and data disks. Each component has a unique purpose and uses different physical disk devices. The fencing driver, known as vxfen, directs CVM as necessary to carry out actual fencing operations at the disk group level. Fencing uses GAB port b for its communication.

In addition to providing I/O fencing capabilities, the I/O fencing module VxFEN is also used to notify Sybase ASE of membership changes on the VCS cluster. When a node is booting, VxFEN will come up after LLT and GAB, process membership information, and reach regular running state. When VxFEN later launches vxferd, the I/O fencing daemon that is used for communication, this daemon first opens a UNIX socket and registers with VCMP, a thread of Sybase ASE. The vxferd daemon is responsible for the communication between VxFEN and VCMP. If the handshake

between vxfsend and VCMP is successful, vxfsend calls an ioctl into the VxFEN kernel module and awaits instructions. VxFEN proceeds to send the current cluster view from VCS perspective to Sybase ASE. When a connection between VxFEN and Sybase ASE has already been established, cluster membership change notification messages are delivered as soon as VxFEN completes any necessary actions (for example, after fencing out departing nodes or lost nodes).

About preferred fencing

The I/O fencing driver uses coordination points to prevent split-brain in a VCS cluster. By default, the fencing driver favors the subcluster with maximum number of nodes during the race for coordination points. With the preferred fencing feature, you can specify how the fencing driver must determine the surviving subcluster.

You can configure the preferred fencing policy using the cluster-level attribute PreferredFencingPolicy for the following:

- Enable system-based preferred fencing policy to give preference to high capacity systems.
- Enable group-based preferred fencing policy to give preference to service groups for high priority applications.
- Enable site-based preferred fencing policy to give preference to sites with higher priority.
- Disable preferred fencing policy to use the default node count-based race policy.

See [“Enabling or disabling the preferred fencing policy”](#) on page 95.

About preventing data corruption with I/O fencing

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

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

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

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

SF Sybase CE uses I/O fencing to remove the risk that is associated with split-brain. I/O fencing allows write access for members of the active cluster. It blocks access to storage from non-members.

About SCSI-3 Persistent Reservations

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

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

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

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

About I/O fencing operations

I/O fencing, provided by the kernel-based fencing module (vxfen), performs identically on node failures and communications failures. When the fencing module on a node is informed of a change in cluster membership by the GAB module, it immediately begins the fencing operation. The node tries to eject the key for departed

nodes from the coordinator disks using the preempt and abort command. When the node successfully ejects the departed nodes from the coordinator disks, it also ejects the departed nodes from the data disks. In a split-brain scenario, both sides of the split would race for control of the coordinator disks. The side winning the majority of the coordinator disks wins the race and fences the loser. The loser then panics and restarts the system.

About I/O fencing components

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

- Data disks—Store shared data
See [“About data disks”](#) on page 29.
- Coordination points—Act as a global lock during membership changes
See [“About coordination points”](#) on page 29.

About data disks

Data disks are standard disk devices for data storage and are either physical disks or RAID Logical Units (LUNs).

These disks must support SCSI-3 PR and must be part of standard VxVM or CVM disk groups. CVM is responsible for fencing data disks on a disk group basis. Disks that are added to a disk group and new paths that are discovered for a device are automatically fenced.

About coordination points

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

Disks that act as coordination points are called coordinator disks. Coordinator disks are three standard disks or LUNs set aside for I/O fencing during cluster reconfiguration. Coordinator disks do not serve any other storage purpose in the SF Sybase CE configuration.

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

See the *Storage Foundation Administrator's Guide*.

How I/O fencing works in different event scenarios

[Table 1-2](#) describes how I/O fencing works to prevent data corruption in different failure event scenarios. For each event, review the corrective operator actions.

Table 1-2 I/O fencing scenarios

Event	Node A: What happens?	Node B: What happens?	Operator action
Both private networks fail.	Node A races for majority of coordination points. If Node A wins race for coordination points, Node A ejects Node B from the shared disks and continues.	Node B races for majority of coordination points. If Node B loses the race for the coordination points, Node B panics and removes itself from the cluster.	When Node B is ejected from cluster, repair the private networks before attempting to bring Node B back.
Both private networks function again after event above.	Node A continues to work.	Node B has crashed. It cannot start the database since it is unable to write to the data disks.	Restart Node B after private networks are restored.
One private network fails.	Node A prints message about an IOFENCE on the console but continues.	Node B prints message about an IOFENCE on the console but continues.	Repair private network. After network is repaired, both nodes automatically use it.

Table 1-2 I/O fencing scenarios (continued)

Event	Node A: What happens?	Node B: What happens?	Operator action
Node A hangs.	<p>Node A is extremely busy for some reason or is in the kernel debugger.</p> <p>When Node A is no longer hung or in the kernel debugger, any queued writes to the data disks fail because Node A is ejected. When Node A receives message from GAB about being ejected, it panics and removes itself from the cluster.</p>	<p>Node B loses heartbeats with Node A, and races for a majority of coordination points.</p> <p>Node B wins race for coordination points and ejects Node A from shared data disks.</p>	<p>Repair or debug the node that hangs and reboot the node to rejoin the cluster.</p>

Table 1-2 I/O fencing scenarios (*continued*)

Event	Node A: What happens?	Node B: What happens?	Operator action
<p>Nodes A and B and private networks lose power. Coordination points and data disks retain power.</p> <p>Power returns to nodes and they restart, but private networks still have no power.</p>	<p>Node A restarts and I/O fencing driver (vxfen) detects Node B is registered with coordination points. The driver does not see Node B listed as member of cluster because private networks are down. This causes the I/O fencing device driver to prevent Node A from joining the cluster. Node A console displays:</p> <p>Potentially a preexisting split brain. Dropping out of the cluster. Refer to the user documentation for steps required to clear preexisting split brain.</p>	<p>Node B restarts and I/O fencing driver (vxfen) detects Node A is registered with coordination points. The driver does not see Node A listed as member of cluster because private networks are down. This causes the I/O fencing device driver to prevent Node B from joining the cluster. Node B console displays:</p> <p>Potentially a preexisting split brain. Dropping out of the cluster. Refer to the user documentation for steps required to clear preexisting split brain.</p>	<p>Resolve preexisting split-brain condition.</p> <p>See “Fencing startup reports preexisting split-brain” on page 131.</p>

Table 1-2 I/O fencing scenarios (*continued*)

Event	Node A: What happens?	Node B: What happens?	Operator action
Node A crashes while Node B is down. Node B comes up and Node A is still down.	Node A is crashed.	Node B restarts and detects Node A is registered with the coordination points. The driver does not see Node A listed as member of the cluster. The I/O fencing device driver prints message on console: Potentially a preexisting split brain. Dropping out of the cluster. Refer to the user documentation for steps required to clear preexisting split brain.	Resolve preexisting split-brain condition. See “Fencing startup reports preexisting split-brain” on page 131.
The disk array containing two of the three coordination points is powered off. No node leaves the cluster membership	Node A continues to operate as long as no nodes leave the cluster.	Node B continues to operate as long as no nodes leave the cluster.	Power on the failed disk array so that subsequent network partition does not cause cluster shutdown, or replace coordination points. See “Replacing I/O fencing coordinator disks when the cluster is online” on page 86.

Table 1-2 I/O fencing scenarios (*continued*)

Event	Node A: What happens?	Node B: What happens?	Operator action
The disk array containing two of the three coordination points is powered off. Node B gracefully leaves the cluster and the disk array is still powered off. Leaving gracefully implies a clean shutdown so that vxfs is properly unconfigured.	Node A continues to operate in the cluster.	Node B has left the cluster.	Power on the failed disk array so that subsequent network partition does not cause cluster shutdown, or replace coordination points. See “Replacing I/O fencing coordinator disks when the cluster is online” on page 86.
The disk array containing two of the three coordination points is powered off. Node B abruptly crashes or a network partition occurs between node A and node B, and the disk array is still powered off.	Node A races for a majority of coordination points. Node A fails because only one of the three coordination points is available. Node A panics and removes itself from the cluster.	Node B has left cluster due to crash or network partition.	Power on the failed disk array and restart I/O fencing driver to enable Node A to register with all coordination points, or replace coordination points. See “Replacing defective disks when the cluster is offline” on page 133.

Sybase ASE CE components

Sybase ASE consists of a single monolithic, user space process named `dataserver`. A single ASE instance may consist of multiple `dataserver` processes, each representing an ‘engine’ in a single instance. The engines communicate via shared memory. ASE’s internal threads run across these engines, allowing a single instance to scale to tens of thousands of concurrent users and dozens of processors on an SMP system.

Sybase ASE CE has various clustering components and a failure detection mechanism to enable multiple instances of the same database to simultaneously access it while providing protection against failures at various levels.

The following components are part of Sybase ASE CE:

- CMS (Cluster Membership Service)

Membership management is provided by CMS which is built into the dataserver binary. ASE only handles application level membership management. It is only concerned about applications, namely dataserver, running on the cluster nodes. ASE does not differentiate between a software level failure and a physical node failure.

- **Quorum Device**

ASE utilizes a single quorum device to assist with membership management. Quorum device serves as a membership voting area, but also acts as a configuration repository and a semaphore for numerous operations. All access to the quorum device is through a quorum management library which exposes a common API. The cluster definition is stored in the configuration section of the quorum device. This definition includes the instances in the cluster, the nodes they run on, interconnect address, etc. This is essential information to bootstrap each instance. The quorum API provides a disk based distributed locking mechanism. This distributed lock is implemented entirely in software and requires no network communication.

Quorum locks currently have three primary uses:

- Race prevention at boot time
- Configuration changes
- Split brain prevention

The quorum API also provides a mechanism to query the state of each instance without needing to connect to the database server.

- **CIPC**

Sybase has a built-in layer known as CIPC (Cluster Inter Process Communication) to provide message passing capabilities to the various subsystems within the dataserver. Cluster instances communicate via connection oriented UDP/IP, with CIPC providing reliability on top of UDP. Sybase recommends two private networks for the cluster interconnect.

The following mechanisms are used within ASE CE:

- **Heart-beating among instances**

ASE instances exchange periodic heartbeats over the cluster interconnect to signify instance health. The default period is 5 seconds, and this is dynamically configurable. There is also a dynamically configurable number of retries before which missing heartbeats translate into membership failure. Although heartbeat messages are sent explicitly, "proxy heartbeating" is also supported where any message exchange between instances during the heartbeat period can serve as a proxy for the true heartbeat message. This has improved reliability in stress situations.

The heartbeat interval can be bypassed for software failures - failures where the underlying hardware is intact. Sybase CE instances use UDP, the UDP driver on the remote node provides notification when the ASE process exists. This allows the remaining instances to immediately go into membership failure. In this situation the time from process exit to formation of the new cluster view may be under one second.

- **Monitoring the health of private interconnects**
A separate mechanism called linkswitch is used to monitor the health of the two interconnect links. Linkswitch is part of the larger CIPC module. When multiple links are configured, linkswitch will detect the loss of one of the links and provide traffic switching. It also detects when a down link comes back online.

Note: The above mechanism of cluster heart-beating, linkswitch, and connected UDP allow CMS to detect the failure of the ASE process, individual interconnects, and the overall physical node (although it is not always clear which of these failures has occurred).

- **Monitoring the accessibility to the disk sub-system**
A quorum heartbeat mechanism is used to determine when an instance has lost the ability to write to the disk subsystem. ASE periodically writes a heartbeat value to the quorum device. If this write fails ASE assumes that it has lost access to the disk subsystem and the instance terminates. The frequency of the heartbeat writes and the number of retries are both configurable. Note that this scheme assumes that the access to the quorum device utilizes the same fabric / SAN as the database devices.

About optional features in SF Sybase CE

SF Sybase CE supports the following activities using optional product features:

- [Typical configuration of SF Sybase CE clusters in secure mode](#)
- [Typical configuration of VOM-managed SF Sybase CE clusters](#)
- [About SF Sybase CE global cluster setup for disaster recovery](#)

Typical configuration of SF Sybase CE clusters in secure mode

Enabling secure mode for SF Sybase CE guarantees that all inter-system communication is encrypted and that security credentials of users are verified.

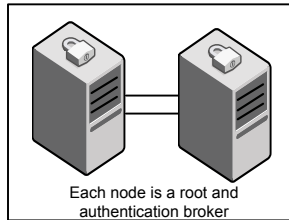
[Figure 1-7](#) illustrates typical configuration of SF Sybase CE clusters in secure mode.

For information about how to configure secure clusters, see your product installation guide.

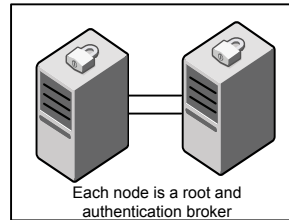
Figure 1-7 Typical configuration of SF Sybase CE clusters in secure mode

Multiple clusters

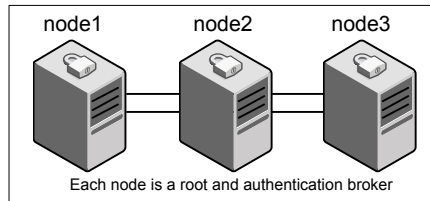
Cluster 1



Cluster 2



Single cluster



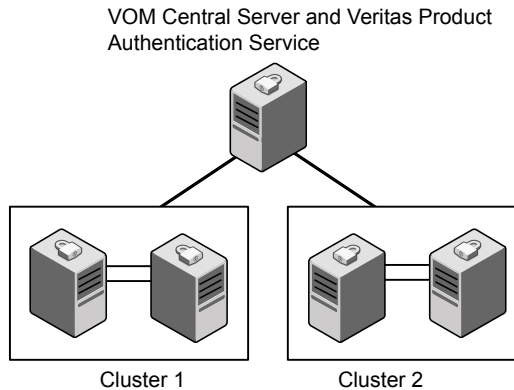
Typical configuration of VOM-managed SF Sybase CE clusters

Veritas InfoScale Operations Manager (VOM) provides a centralized management console for Storage Foundation and High Availability products.

See [“About Veritas InfoScale Operations Manager”](#) on page 39.

[Figure 1-8](#) illustrates a typical setup of SF Sybase CE clusters that are centrally managed using Veritas InfoScale Operations Manager.

Figure 1-8 Typical configuration of VOM-managed clusters



About SF Sybase CE global cluster setup for disaster recovery

SF Sybase CE leverages the global clustering feature of VCS to enable high availability and disaster recovery (HA/DR) for businesses that span wide geographical areas. Global clusters provide protection against outages caused by large-scale disasters such as major floods, hurricanes, and earthquakes. An entire cluster can be affected by such disasters. This type of clustering involves migrating applications between clusters over a considerable distance.

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

SF Sybase CE supports host-based replication using Veritas Volume Replicator (VVR). VVR replicates data to remote sites over any standard IP network. The host at the source location on which the application is running is known as the primary host. The host at the target location is known as the secondary host.

Hardware-based replication technologies are not supported at the time of publication.

How the agent makes Sybase highly available

The Cluster Server agent for Sybase continuously monitors the Sybase database processes to verify they function properly.

The agent for Sybase can perform different levels of monitoring and different actions which you can configure.

- Primary or Basic monitoring: In the basic monitoring mode, the agent monitors the configured Sybase server process, and fails over or restarts the group during an application failure.
- Detail monitoring: In the optional detail monitoring mode, the agent detects application failure if it cannot perform a transaction in the test table in the Sybase database server.
- Intelligent monitoring framework (IMF): The Sybase agent is IMF-aware and uses asynchronous monitoring framework (AMF) kernel driver for resource state change notifications.

For Sybase cluster edition, the agent uses the Sybase-provided utility, `qrmutil`, to know if the status of the instance is up or down. If `qrmutil` reports the status as failure pending, the agent reboots the node and the instance is automatically started again.

When the Sybase agent detects that the configured Sybase server is not running on a system, based on the value of `OnlineRetryLimit` attribute of the Sybase server group, the service group is restarted on the same system on which the group faulted.

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.

About Veritas Services and Operations Readiness Tools (SORT)

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

SORT can help you do the following:

- | | |
|---|--|
| Prepare for your next installation or upgrade | <ul style="list-style-type: none"> ■ List product installation and upgrade requirements, including operating system versions, memory, disk space, and architecture. ■ Analyze systems to determine if they are ready to install or upgrade Veritas products. ■ Download the latest patches, documentation, and high availability agents from a central repository. ■ Access up-to-date compatibility lists for hardware, software, databases, and operating systems. |
| Manage risks | <ul style="list-style-type: none"> ■ Get automatic email notifications about changes to patches, array-specific modules (ASLs/APMs/DDIs/DDIs), and high availability agents from a central repository. ■ Identify and mitigate system and environmental risks. ■ Display descriptions and solutions for hundreds of Veritas error codes. |
| Improve efficiency | <ul style="list-style-type: none"> ■ Find and download patches based on product version and platform. ■ List installed Veritas products and license keys. ■ Tune and optimize your environment. |

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

To access SORT, go to:

<https://sort.veritas.com>

Administering SF Sybase CE and its components

This chapter includes the following topics:

- [Administering SF Sybase CE](#)
- [Administering VCS](#)
- [Administering I/O fencing](#)
- [Administering CVM](#)
- [Administering CFS](#)
- [Administering the Sybase agent](#)

Administering SF Sybase CE

This section provides instructions for the following SF Sybase CE administration tasks:

- Setting the environment variables
See [“Setting the environment variables for SF Sybase CE”](#) on page 42.
- Starting or stopping SF Sybase CE on each node
See [“Starting or stopping SF Sybase CE on each node”](#) on page 43.
- Applying operating system updates on SF Sybase CE nodes
See [“Applying operating system updates on SF Sybase CE nodes”](#) on page 48.
- Adding storage to an SF Sybase CE cluster
See [“Adding storage to an SF Sybase CE cluster”](#) on page 49.
- Recovering from storage failure

See [“Recovering from storage failure”](#) on page 50.

- Enhancing the performance of SF Sybase CE clusters
See [“Enhancing the performance of SF Sybase CE clusters”](#) on page 51.
- Administering SmartIO
- Verifying the nodes in an SF Sybase CE cluster
See [“Verifying the nodes in an SF Sybase CE cluster”](#) on page 51.

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 Sybase CE, refer to the troubleshooting section for assistance.

See [“About troubleshooting SF Sybase CE”](#) on page 114.

Setting the environment variables for SF Sybase CE

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:\n/opt/VRTS/man`

For csh `# setenv MANPATH /usr/share/man:\n/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:\n/opt/VRTS/bin\n$PATH; export PATH`

Starting or stopping SF Sybase CE on each node

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

To start SF Sybase CE	Using installer: See “Starting SF Sybase CE using the script-based installer” on page 43. Manual: See “Starting SF Sybase CE manually on each node” on page 43.
To stop SF Sybase CE	Using installer: See “Stopping SF Sybase CE using the script-based installer” on page 45. Manual: See “Stopping SF Sybase CE manually on each node” on page 46.

Starting SF Sybase CE using the script-based installer

Run the installer with the `-start` option to start SF Sybase CE on each node.

Note: Start SF Sybase CE 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 Sybase CE using the installer

- 1 Log into one of the nodes in the cluster as the root user.
- 2 Start SF Sybase CE:

```
# /opt/VRTS/install/installer \  
-start system1 system2
```

Starting SF Sybase CE manually on each node

Perform the steps in the following procedures to start SF Sybase CE manually on each node.

To start SF Sybase CE 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 lltd
```

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

```
# /etc/init.d/lltd 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 vxfsd
```

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

```
# /etc/init.d/vxfsd 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 VCS, CVM, and CFS:

```
# hastart
```

7 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 f gen    564024 membership 01
Port h gen    56401a membership 01
Port m gen    56401c 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 Sybase CE using the script-based installer

Run the installer with the `-stop` option to stop SF Sybase CE on each node.

To stop SF Sybase CE using the installer

- 1 Log into one of the nodes in the cluster as the root user.
- 2 Unmount the VxFS file system, which is not under VCS control.

```
# mount |grep vxfs
# fuser -m /mount_point
# umount /mount_point
```

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

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

3 Stop VCS:

```
# hastop -all
```

4 Stop SF Sybase CE:

```
# /opt/VRTS/install/installer \
-stop system1 system2
```

Stopping SF Sybase CE manually on each node

Perform the steps in the following procedures to stop SF Sybase CE manually on each node.

To stop SF Sybase CE manually on each node

1 Stop the Sybase database.

If the Sybase ASE CE instance is managed by VCS, log in as the root user and take the service group offline.

```
# hagrps -offline sybase_group -sys node_name
```

2 Stop all applications that are not configured under VCS but dependent on Sybase ASE CE 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:

```
# hagrps -offline group_name -sys node_name
```

Verify that the VCS service groups are offline:

```
# hagrps -state group_name
```

When Sybase group under VCS control is offline, the vxfsend daemon also stops.

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 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 or shared volumes:

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

- 8 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
```

- 9 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
```

- 10 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
```

11 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
```

12 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 operating system updates on SF Sybase CE 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:

```
# hagrps -offline grp_name -sys node_name
```

- 3 Stop SF Sybase CE:

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

- 4 Stop the applications not configured under VCS, but dependent on Sybase ASE CE or the resources controlled by VCS. Use native application commands to stop the applications.

- 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 or shared volumes:

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

- 6 Unmount the VxFS file system:

```
# umount /mount_point
```

- 7 Upgrade the operating system.

See the operating system documentation.

- 8 If the node is not rebooted after the operating system upgrade, reboot the node:

```
# shutdown -r now
```

- 9 Repeat all the steps on each node in the cluster.

Adding storage to an SF Sybase CE cluster

You can add storage to an SF Sybase CE cluster in the following ways:

Add a disk to a disk group

Use the `vxvg` command to add a disk to a disk group.

See the `vxvg` (1M) manual page for information on various options.

See [“To add storage to an SF Sybase CE cluster by adding a disk to a disk group”](#) on page 50.

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 Sybase CE cluster by extending the volume space on a disk group”](#) on page 50.

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

To add storage to an SF Sybase CE cluster by adding a disk to a disk group

- ◆ Add a disk to the disk group:

```
# vxdbg -g dg_name adddisk disk_name
```

To add storage to an SF Sybase CE 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 `sybvol` in the disk group `dbdata_dg` can grow, given its attributes and free storage available:

```
# vxassist -g dbdata_dg 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 `sybvol` of size 10 GB in the disk group `dbdata_dg` to 30 GB:

```
# vxresize -g dbdata_dg \
sybvol 30g
```

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

```
# vxresize -g dbdata_dg \
sybvol +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.

Enhancing the performance of SF Sybase CE clusters

The main components of clustering that impact the performance of an SF Sybase CE 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.

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

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

Verifying the nodes in an SF Sybase CE cluster

[Table 2-1](#) lists the various options that you can use to periodically verify the nodes in your cluster.

Table 2-1 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 .

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 52.

- Starting and stopping VCS
See [“Starting and stopping VCS”](#) on page 54.
- Environment variables to start and stop VCS modules
See [“Environment variables to start and stop VCS modules”](#) on page 54.
- Adding and removing LLT links
See [“Adding and removing LLT links”](#) on page 58.
- Displaying the cluster details and LLT version for LLT links
See [“Displaying the cluster details and LLT version for LLT links”](#) on page 62.
- Configuring aggregated interfaces under LLT
See [“Configuring aggregated interfaces under LLT”](#) on page 60.
- Configuring destination-based load balancing for LLT
See [“Configuring destination-based load balancing for LLT”](#) on page 63.
- Enabling and disabling intelligent resource monitoring
See [“Enabling and disabling intelligent resource monitoring for agents manually”](#) on page 63.
- Administering the AMF kernel driver
See [“Administering the AMF kernel driver”](#) on page 65.

Viewing available Veritas device drivers

To view the available Veritas devices:

```
# cat /proc/devices
```

Character devices:

```
1 mem
2 pty
3 tty
4 /dev/vc/0
4 tty
4 ttyS
5 /dev/tty
5 /dev/console
5 /dev/ptmx
7 vcs
10 misc
13 input
21 sg
29 fb
128 ptm
136 pts
```

```
180 usb
189 usb_device
199 VxVM
200 VxSPEC
250 vxfen
251 vxglm
252 gab
253 llt
254 amf
```

Block devices:

```
1 ramdisk
2 fd
3 ide0
7 loop
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                282172  12
```

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

```
# lsmod |grep "^vx"
vxfen              321896  1
vxglm              256672  1
vxspec             21656  6
vxio               3228456  5 vxspec, [unsafe]
vxdmp              357624  40
vxportal           24320  3
vxfs               2503680  7 vxportal, fdd
```

Starting and stopping VCS

This section describes how to start and stop the VCS.

To start VCS

- ◆ On each node, start VCS:

```
# hstart
```

To stop VCS

- ◆ On each node, stop VCS:

```
# hstop -local
```

You can also use the command `hstop -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 Sybase CE, you must either enable or disable the startup and shutdown modes for all these modules.

Table 2-2 Start and stop environment variables for VCS

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

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

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

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

Environment variable	Definition and default value
VCS_STOP_TIMEOUT	<p>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.</p> <p>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.</p> <p>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 (<code>hastop -local -force</code>) 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.</p> <p>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.</p> <p>In <code>systemd</code> environments, the <code>TimeoutStopSec</code> attribute of the VCS service is set to 0 (infinity) to prevent the stop operation from timing out.</p> <p>This environment variable is defined in the <code>/etc/sysconfig/vcs</code> file.</p> <p>Note: Veritas recommends that you do not change the <code>TimeoutStopSec</code> attribute of the VCS service. If you want to configure a time-out value for the stop operation, use the <code>VCS_STOP_TIMEOUT</code> variable in the <code>/etc/sysconfig/vcs</code> file.</p> <p>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:</p> <p>Scenario 1, which may result in some unexpected behavior: If the value of <code>VCS_STOP_TIMEOUT</code> 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.</p> <p>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.</p> <p>Default value: 0 seconds (indicates no time-out)</p>

Adding and removing LLT links

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

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/llttab` 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	<p>Device name of the interface.</p> <p>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.</p> <p>For link types udp and udp6, the device is the udp and udp6 device name respectively.</p> <p>For link type rdma, the device name is udp.</p>
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:

```
# lltnconfig -t eth4 -d eth4 -s 0xcale -m 1500
```

- For UDP link type:

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

- For UDP6 link type:

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

- For RDMA link:

```
# lltnconfig -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 Run the following command to disable a network link that is configured under LLT.

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

- 2 Wait for the 16 seconds (LLT peerinact time).
- 3 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 lltd
```

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

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

See [“Starting or stopping SF Sybase CE on each node”](#) on page 43.

- 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 lltd
```

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

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

See [“Starting or stopping SF Sybase CE on each node”](#) on page 43.

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/llttab` file.

See [“To configure aggregated interfaces under LLT by editing the `/etc/llttab` file”](#) on page 61.

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 : 0xcale
promisc sap : 0
promisc mac : 0
cidsnoop : 1
=== Listening for LLT packets ===
cid nid vma j 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.

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 24.

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.

- 4 Save the VCS configuration.

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

- 5 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 23.

See [“Environment variables to start and stop VCS modules”](#) on page 54.

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 66.
 - 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 66.

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 67.
- About vxfentsthdw utility
See [“About the vxfentsthdw utility”](#) on page 68.
- About vxfenadm utility
See [“About the vxfenadm utility”](#) on page 76.
- About vxfenclearpre utility
See [“About the vxfenclearpre utility”](#) on page 81.
- About vxfenswap utility
See [“About the vxfenswap utility”](#) on page 85.
- About enabling or disabling the preferred fencing policy
See [“Enabling or disabling the preferred fencing policy”](#) on page 95.

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

See [“Troubleshooting I/O fencing”](#) on page 129.

See [“About administering I/O fencing”](#) on page 67.

About administering I/O fencing

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

vxfentsthdw	Tests SCSI-3 functionality of the disks for I/O fencing See “About the vxfentsthdw utility” on page 68.
-------------	--

<code>vxfenconfig</code>	Configures and unconfigures I/O fencing Lists the coordination points used by the <code>vxfen</code> driver.
<code>vxfenadm</code>	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 76.
<code>vxfenclearpre</code>	Removes SCSI-3 registrations and reservations from disks See “About the vxfenclearpre utility” on page 81.
<code>vxfenswap</code>	Replaces coordination points without stopping I/O fencing See “About the vxfenswap utility” on page 85.
<code>vxfendisk</code>	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 Sybase ASE CE Configuration Guide* to set up I/O fencing.

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

Refer also to the `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 `vxfcntlsthaw` 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 `vxfcntlsthaw -i diskpath` command to verify a disk's serial number. See [“Verifying that the nodes see the same disk”](#) on page 81.
- 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 diskpath_a is ready to be configured for
I/O Fencing on node system1
```

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 `vxfcntlsthaw` command options

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

Table 2-3 `vxfcntlsthaw` options

vxfcntlsthaw option	Description	When to use
<code>-n</code>	Utility uses rsh for communication.	Use when rsh is used for communication.

Table 2-3 vxfcntlshdw options (*continued*)

vxfcntlshdw option	Description	When to use
-r	Non-destructive testing. Testing of the disks for SCSI-3 persistent reservations occurs in a non-destructive way; that is, there is only testing for reads, not writes. 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 73.
-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 vxfcntlshdw 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 vxfcntlshdw 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 vxfcntlshdw” on page 71.
-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 vxfcntlshdw -m option” on page 73.

Table 2-3 vxfentsthdw options (*continued*)

vxfentsthdw option	Description	When to use
-f <i>filename</i>	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 75.
-g <i>disk_group</i>	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 75.
-o	Utility overrides 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 system1, the disks are seen as /dev/sdg, /dev/sdh, and /dev/sdi.
- From the node system2, 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, vxfencoordg, be accessible from two nodes.

To test the coordinator disk group using vxfcntlsthaw -c

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

```
# vxfcntlsthaw -c vxfencoorddg
```

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

```
Enter the first node of the cluster: system1
```

```
Enter the second node of the cluster: system2
```

- 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  
system1 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  
system2 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 Sybase ASE CE Configuration Guide* for instructions to initialize disks for I/O fencing and to set up coordinator disk groups.
If necessary, start the disk group.
See the *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 vxfcntlshdw”](#) on page 71.

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:

```
# vxfcntlshdw -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 vxfcntlshdw -m option

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

This procedure uses the `diskpath_a` disk in the steps.

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

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

```
The disk diskpath_a is ready to be configured for
I/O Fencing on node system1
```

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

To test disks using the `vxfcntlsthdw` script

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

```
# vxfcntlsthdw [-n]
```

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

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

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

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

```
Enter the first node of the cluster: system1
```

```
Enter the second node of the cluster: system2
```

- 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
system1 in the format:
```

```
for dmp: /dev/vx/rdmp/sdx
```

```
for raw: /dev/sdx
```

```
Make sure it's the same disk as seen by nodes system1 and system2
/dev/sdr
```

```
Enter the disk name to be checked for SCSI-3 PGR on node
system2 in the format:
```

```
for dmp: /dev/vx/rdmp/sdx
```

```
for raw: /dev/sdx
```

```
Make sure it's the same disk as seen by nodes system1 and system2
/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 diskpath_a have PASSED
The disk is now ready to be configured for I/O Fencing on node
system1
...
Removing test keys and temporary files, if any ...
.
.
```

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

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

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

To test the shared disks listed in a file

- 1 Create a text file `disks_test` to test two disks shared by systems `system1` and `system2` that might resemble:

```
system1 /dev/sdz system2 /dev/sdy
system1 /dev/sdu system2 /dev/sdw
```

where the first disk is listed in the first line and is seen by `system1` as `/dev/sdz` and by `system2` as `/dev/sdy`. The other disk, in the second line, is seen as `/dev/sdu` from `system1` and `/dev/sdw` from `system2`. Typically, the list of disks could be extensive.

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

```
# vxfcntlsthdw -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 `vxfcntlsthdw -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 `vxvg 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`:

```
# vxfsentsthdw -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 `diskpath_a` from the system with node ID 1, enter the following command:

```
# vxfenadm -s diskpath_a
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: system1
```

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: system2
```

- 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: system1
```

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: system2
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: system1
```

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: system2
```

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 Sybase ASE CE.
- 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 82.

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 `vxfcntlpre` utility, you can refer to the log file at, `/var/VRTSvcs/log/vxfen/vxfen.log` file.

Removing preexisting keys

If you encountered a split-brain condition, use the `vxfcntlpre` 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.

```
# hstart
```

About the vxfsnwap utility

The vxfsnwap 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.

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 `vxfsnwap(1M)` manual page.

See the *Storage Foundation for Sybase ASE CE Configuration 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 86.
See [“Replacing the coordinator disk group in a cluster that is online”](#) on page 89.
If you want to replace the coordinator disks when the cluster is offline, you cannot use the vxfsnwap utility. You must manually perform the steps that the utility does to replace the coordinator disks.
See [“Replacing defective disks when the cluster is offline”](#) on page 133.
- 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 94.

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.

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: vxfen_mode
```

```
Fencing SCSI3 Disk Policy: dmp
```

```
Cluster Members:
```

```
  * 0 (system1)
```

```
  1 (system2)
```

```
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, `vxfencoordg`) 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 vxencoorddg 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 Sybase ASE CE Configuration 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.

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

- 18** Deport the diskgroup.

```
# vxdbg 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: vxfen_mode
Fencing SCSI3 Disk Policy: dmp
Cluster Members:
    * 0 (system1)
    1 (system2)
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
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 71.

- 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 Sybase ASE CE Configuration Guide* for more information.

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

```
# vxfsnwap -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 `vxfsnwap` 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 `vxfsnwap` operation.

```
Do you want to clear the keys on the coordination points
and proceed with the vxfsnwap 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 vxfencoordg 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 `vxfsnwap` 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.

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: vxfen_mode
```

```
Fencing SCSI3 Disk Policy: dmp
```

```
Cluster Members:
```

```
  * 0 (system1)
```

```
  1 (system2)
```

```
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/disk_a  
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 vxfencoordg [-n]
```

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

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

```
Device Name: /dev/vx/rdmp/disk_a  
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.

See [“About preferred fencing”](#) on page 27.

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 system1 five times more weight compared to system2, run the following commands:

```
# hasys -modify system1 FencingWeight 50
# hasys -modify system2 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:

```
# hagrps -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.

Administering CVM

This section provides instructions for the following CVM administration tasks:

- Establishing CVM cluster membership manually
See [“Establishing CVM cluster membership manually”](#) on page 99.
- Changing CVM master manually
See [“Changing the CVM master manually”](#) on page 99.
- Importing a shared disk group manually
See [“Importing a shared disk group manually”](#) on page 102.
- Deporting a shared disk group manually

See [“Deporting a shared disk group manually”](#) on page 102.

- Verifying if CVM is running in an SF Sybase CE cluster
See [“Verifying if CVM is running in an SF Sybase CE cluster”](#) on page 102.
- Verifying CVM membership state
See [“Verifying CVM membership state”](#) on page 103.
- Verifying the state of CVM shared disk groups
See [“Verifying the state of CVM shared disk groups”](#) on page 103.
- Verifying the activation mode
See [“Verifying the activation mode”](#) on page 104.

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

See [“Troubleshooting Cluster Volume Manager in SF Sybase CE clusters”](#) on page 136.

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
system1	0	0	Joined: Master
system2	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 101.

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
system1        0          0         Joined: Slave
system2        1          1         Joined: Master

# vxdctl -c mode
mode: enabled: cluster active - MASTER
master: system2
```

In this example, the CVM master is system2.

- 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 system2 to system1:

```
# vxclustadm setmaster system1
```

- 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
system1        0          0         Joined: Master
system2        1          1         Joined: Slave

# vxdctl -c mode
mode: enabled: cluster active - MASTER
master: system1
```

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 system1
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:

```
# vxdbg -s import dg_name
```

Deporting a shared disk group manually

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

```
# vxdbg deport dg_name
```

Note that the deport of a shared disk group removes the SCSI-3 PGR keys on the disks.

Verifying if CVM is running in an SF Sybase CE cluster

You can use the following options to verify whether Cluster Volume Manager is up or not in an SF Sybase CE cluster.

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

```
# vxctl -c mode  
mode: enabled: cluster inactive  
# vxclustadm -v nodestate  
state: out of cluster
```

On the master node, the following output is displayed:

```
# vxctl -c mode
```

```
mode: enabled: cluster active - MASTER
master: system1
```

On the slave nodes, the following output is displayed:

```
# vxctl -c mode

mode: enabled: cluster active - SLAVE
master: system2
```

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

```
# vxclustadm nidmap
```

Name	CVM Nid	CM Nid	State
system1	0	0	Joined: Master
system2	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 `vxctl -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 Sybase CE cluster:

```
# vxdg list |grep shared
```

```
sybbin_dg enabled, shared 1052685125.1485.csha3
```

Verifying the activation mode

In an SF Sybase CE 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 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 104.
- Using cfsmount to mount CFS file systems
- Resizing CFS file systems
See [“Resizing CFS file systems”](#) on page 105.
- Verifying the status of CFS file systems
See [“Verifying the status of CFS file system nodes and their mount points”](#) on page 105.

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 quorum_dg quorum_vol /quorum sybasece \  
all=suid,rw
```

```
Mount Point is being added...
```

```
/quorum added to the cluster-configuration
```


Uses of cfsmount to mount and cfsumount to unmount CFS file system

To mount a CFS file system

- ◆ Mount a CFS file system:

```
# cfsmntadm add sdg vol1 /quorum all=
# cfsmount /quorum
Mounting...
[/dev/vx/rdisk/quorum_dg/quorum
mounted successfully at /quorum_vol on system1
[/dev/vx/rdisk/quorum_dg/quorum_vol]
mounted successfully at /quorum on system2
```

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 Sybase binary CFS file system (the Sybase binary volume is CFS mounted):

```
# vxresize -g sybbindg sybbin_vol +2G
```

where `sybbindg` is the CVM disk group, `sybbin_vol` is the volume, and `+2G` indicates the increase 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                : system2
Cluster Manager     : not-running
CVM state           : not-running
MOUNT POINT         SHARED VOLUME  DISK GROUP      STATUS
```

/quorum	quorum_vol	quorum_dg	NOT MOUNTED
/sybase	sybbin_vol	sybbindg	NOT MOUNTED
/sybdata	sybvol	dbdata_dg	NOT MOUNTED

Node	:	system1	
Cluster Manager	:	running	
CVM state	:	running	

MOUNT POINT	SHARED VOLUME	DISK GROUP	STATUS
/quorum	quorum_vol	quorum_dg	MOUNTED
/sybase	sybbin_vol	sybbindg	MOUNTED
/sybdata	sybvol	dbdata_dg	MOUNTED

Administering the Sybase agent

SF Sybase CE includes the VCS Sybase agent. The agent can perform different operations or functions on the database. These functions are online, offline, monitor, and clean.

Sybase agent functions

The agent for Sybase starts a Sybase ASE dataserer, monitors the server processes, shuts down the server, and restarts the server processes..

The Sybase agent is IMF-aware.

[Table 2-4](#) lists the Sybase agent for SQL server functions.

Table 2-4 Sybase agent functions

Agent function	Description
Online	<p>Starts the Sybase ASE dataserver by using the following command.</p> <pre>startserver -f \$SYBASE/\$SYBASE_ASE/install/RUN_\$Server</pre> <p>where \$Server is the instance_name, and \$SYBASE/\$SYBASE_ASE/install/RUN_\$Server is the default location of the Run server file. If you specify the value of the Run_ServerFile attribute, then the value that you specify is used instead of the default location.</p> <p>If the WaitForRecovery attribute is enabled, the agent waits either till recovery has been completed and all databases that can be made online are brought online. The agent queries the recovery status by connecting to the isql session. The OnlineTimeout attribute must be set to a sufficiently large value so that the recovery completes before the OnlineTimeout is reached.</p> <p>By default, the WaitForRecovery attribute is not enabled.</p> <p>If the interfaces file location is specified using the interfaces_File attribute, agent uses [-I interfaces file] option while connecting to the isql session.</p> <p>When DelayAfterOnline attribute is set, the monitor function is invoked after completion of online function, and after the number of seconds specified in DelayAfterOnline attribute have elapsed.</p>
Monitor	<p>In the basic monitoring mode, agent scans the process table for the dataserver process. In the detail monitoring mode, agent runs the script that is specified in MonScript as an option.</p> <p>The agent uses the Sybase provided utility, qrmutil, to know if the status of the instance is up or down. If qrmutil reports the status as failure pending, the agent reboots the node and the instance is automatically started again.</p> <p>See “Monitoring options for the Sybase agent” on page 109.</p>

Table 2-4 Sybase agent functions (*continued*)

Agent function	Description
Offline	<p>Stops the Sybase SQL server by using the <code>isql</code> command in the following manner.</p> <p>If interfaces file location is specified using the <code>interfaces_File</code> attribute, agent uses the specified file while connecting to <code>isql</code> session.</p> <p>The agent first executes the <code>shutdown with wait</code> command.</p> <p>Sybase agent uses the <code>timeout</code> option during shutdown of Sybase dataserver if this option is supported.</p> <p>For Sybase ASE Cluster edition the timeout option for shutdown command is supported from versions 15.5 ESD #1 onwards. If the timeout option is not supported for Sybase ASE Cluster edition, the offline script waits in a loop till the dataserver completely stops. The agent waits for up to the <code>OfflineTimeout</code> duration. If the process is still running, the offline script kills it.</p> <p>When <code>DelayAfterOffline</code> attribute is set, the monitor function is invoked after completion of offline function and after the number of seconds specified in <code>DelayAfterOffline</code> attribute have elapsed.</p>
<code>imf_init</code>	<p>Initializes the agent to interface with the AMF kernel driver, which is the IMF notification module for Sybase agent. This function runs when the agent starts up.</p>
<code>imf_register</code>	<p>Registers or unregisters resource entities with the AMF kernel module. This function runs for each resource after the resource goes into steady state (online or offline).</p>
<code>imf_getnotification</code>	<p>Gets notification about resource state changes. This function runs after the agent initializes with the AMF kernel module. This function continuously waits for notification and takes action on the resource upon notification.</p>
Clean	<p>Forcefully stops the Sybase SQL server by using the <code>isql</code> command in the following manner.</p> <p>The agent first executes the <code>shutdown with wait</code> command.</p> <p>For Sybase ASE Cluster edition, if the <code>shutdown with wait</code> command does not stop the dataserver, the agent directly proceeds to kill the dataserver process.</p>

Monitoring options for the Sybase agent

The VCS agent for Sybase provides two levels of application monitoring: basic and detail.

In the basic monitoring mode, the agent for Sybase monitors the Sybase dataserver processes to verify whether they are running.

For Sybase cluster edition, the agent uses `qrmutil` utility that Sybase provides to get the status of the Sybase instance. If the state returned by `qrmutil` utility is 'failure pending', the agent panics the node. When the Sybase agent detects that the configured Sybase server is not running on a system, based on the value of the `OnlineRetryLimit` attribute of the Sybase service group, the service group is restarted on the same system on which the group faulted.

For example:

```
# qrmutil --quorum_dev=/quorum/quorum.dat --monitor=ase1
Executing 'monitor' command for instance 'ase1'
Instance 'ase1' has a failure pending.
# echo $?
99
```

In this example instance 'ase1' has a failure pending state. The agent will panic the node running the instance 'ase1'. The node will automatically rejoin the cluster after reboot.

In the detail monitoring mode, the agent performs a transaction on a test table in the database to ensure that Sybase server is functioning properly. The test table should be created by the user, and the table is specified in the attribute `Table` for the Sybase agent. The agent uses this test table for internal purposes. Veritas recommends that you do not perform any other transaction on the test table.

Using the IPC Cleanup feature for the Sybase agent

When the Adaptive Server starts, it creates shared memory files in `$SYBASE` to store information about the shared memory segments that it uses. Adaptive Server start-up parameter `-M` can be used to change the location of directory that stores shared memory files. The start-up parameter `-M` should be updated in `RUN_`\$Server file.

If the Sybase home directory is unmounted, the Sybase clean script cannot access the shared memory files and does not clean the IPC resources that are allocated by the Sybase processes. Hence, the agent requires shared memory files to be present in the following directory on local system

```
/var/VRTSvcs/log/tmp/sybase_shm/$Server.
```

In the \$SYBASE/\$SYBASE_ASE/install directory, edit the RUN_\$Server file. Change the location of the directory that stores shared memory files to

/var/VRTSvcS/log/tmp/sybase_shm/\$Server using the -M option.

For example, the file RUN_Sybase_Server resembles the following before the change:

```
/home/sybase/ASE-15_0/bin/dataserver \  
  
-sSybase_Server \  
  
-d/home/sybase/data/master.dat \  
  
-e/home/sybase/ASE-15_0/install/Sybase_Server.log \  
  
-c/home/sybase/ASE-15_0/Sybase_Server.cfg \  
  
-M/home/sybase/ASE-15_0 \  
  
--quorum_dev=/qrmmt/qfile
```

After the replacement, the file resembles:

```
/home/sybase/ASE-15_0/bin/dataserver \  
  
-sSybase_Server \  
  
-d/home/sybase/data/master.dat \  
  
-e/home/sybase/ASE-15_0/install/Sybase_Server.log \  
  
-c/home/sybase/ASE-15_0/Sybase_Server.cfg \  
  
-M/var/tmp/sybase_shm/Sybase_Server \  
  

```

Here Sybase_Server is the name of the Adaptive server.

Note: Make sure you create the /var/VRTSvcS/log/tmp/sybase_shm/Sybase_Server directory with proper permissions.

Configuring the service group Sybase using the command line

The VCS agent for Sybase contains a sample configuration file that can be used as reference to directly modify your present configuration file. This method requires you to restart VCS before the configuration takes effect.

To configure a service group Sybase from the command line

- 1 Log in to a cluster system as superuser.
- 2 Make sure the Sybase type definition is imported into VCS engine.
- 3 Edit the main.cf file at `/etc/VRTSvcs/conf/config/main.cf`. For reference, use the sample files at `/etc/VRTSagents/ha/conf/Sybase`.
 - Create binmnt group to configure CFS mounts for Sybase binaries.
 - Create sybasece service group.
 - Create Sybase resources.
 - Edit the default attributes to match the parameters in your configuration. For added security, you must always provide a secure value for passwords.
 - Assign dependencies to the newly created resources. Refer to the sample file at `/etc/VRTSagents/ha/conf/Sybase/`.
See the *Cluster Server User's Guide* for more information on assigning dependencies.
- 4 Save and close the file.
- 5 Verify the syntax of the file `/etc/VRTSvcs/conf/config/main.cf`

```
# cd /etc/VRTSvcs/conf/config
# hacf -verify .
```
- 6 Start VCS on the local node.

```
# hastart
```
- 7 Start VCS on the other nodes.
- 8 If the system is listed in AutoStartList attribute of the Sybase service group, verify that all Sybase service group resources are brought online.

```
# hagrps -state
```

- 9 Take the service group offline and verify that all the resources are stopped.

```
# hagr -offline service_group -sys system_name  
  
# hagr -state
```

- 10 Bring the service group online again and verify that all the resources are available.

```
# hagr -online service_group -sys system_name  
  
# hagr -state
```

- 11 On all systems, look at the following log files for any errors or status.

```
/var/VRTSvcs/log/engine_A.log  
/var/VRTSvcs/log/Sybase_A.log
```

Bringing the Sybase service group online

Perform the following steps to bring a service group online. Note that in the initial few cycles of bringing a service group online, the memory usage by the agent can spike.

To bring a service group online

- 1 From Cluster Explorer, click the **Service Groups** tab in the configuration tree.
- 2 Right-click the service group and click **Enable Resources** to enable all the resources in this group.
- 3 Right-click the service group, hover over **Enable**, and select either the node or all the nodes where you want to enable the service group.
- 4 Save and close the configuration. Click **File > Save Configuration**, then **Close Configuration**.
- 5 Right-click the service group, pause over Online, and select the system where you want to bring the service group online.

Taking the Sybase service group offline

Perform the following procedure from Cluster Manager (Java Console) to take the service group offline.

To take a service group offline

- 1** In the Cluster Explorer configuration tree with the Service Groups tab selected, right-click the service group that you want to take offline.
- 2** Choose **Offline**, and select the appropriate system from the pop-up menu.

Modifying the Sybase service group configuration

You can dynamically modify the Sybase agent using several methods, including the Cluster Manager (Java Console), Veritas InfoScale Operations Manager, and the command line.

Viewing the agent log for Sybase

The VCS agent for Sybase logs messages to the following files:

`/var/VRTSvcs/log/engine_A.log`

`/var/VRTSvcs/log/Sybase_A.log`

Troubleshooting SF Sybase CE

This chapter includes the following topics:

- [About troubleshooting SF Sybase CE](#)
- [Restarting the installer after a failed network connection](#)
- [Installer cannot create UUID for the cluster](#)
- [Troubleshooting I/O fencing](#)
- [Troubleshooting Cluster Volume Manager in SF Sybase CE clusters](#)
- [Troubleshooting interconnects](#)
- [Troubleshooting Sybase ASE CE](#)

About troubleshooting SF Sybase CE

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 Sybase CE cluster for support analysis

Use troubleshooting scripts to gather information about the configuration and status of your cluster and its modules. The scripts identify Package information, debugging messages, console messages, and information about disk groups and volumes. Forwarding the output of these scripts to Veritas Tech Support can assist with analyzing and solving any problems.

- Gathering configuration information using SORT Data Collector
 See “[Gathering configuration information using SORT Data Collector](#)” on page 115.
- Gathering VCS information for support analysis
 See “[Gathering VCS information for support analysis](#)” on page 115.
- Gathering LLT and GAB information for support analysis
 See “[Gathering LLT and GAB information for support analysis](#)” on page 116.
- Gathering IMF information for support analysis
 See “[Gathering IMF information for support analysis](#)” on page 116.

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 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 120.

- 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 Sybase CE 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
VCS engine log file	<code>/var/VRTSvcs/log/engine_A.log</code>	Contains all actions performed by the high availability daemon <code>had</code> . Note: Verify if there are any CVM errors logged in this file, since they may prove to be critical errors.
CVM log files	<code>/var/adm/vx/cmdlog</code> <code>/var/adm/vx/ddl.log</code> <code>/var/adm/vx/translog</code> <code>/var/adm/vx/dmpevents.log</code> <code>/var/VRTSvcs/log/engine_A.log</code>	The <code>cmdlog</code> file contains the list of CVM commands. For more information on collecting important CVM logs: See “Collecting important CVM logs” on page 118.
VCS agent log files	<code>/var/VRTSvcs/log/agenttype_A.log</code> where <i>agenttype</i> is the type of the VCS agent. For example, the log files for the CFS agent can be located at: <code>/var/VRTSvcs/log/CFSMount_A.log</code> For example, the log files for the Sybase agent can be located at: <code>/var/VRTSvcs/log/Sybase_A.log</code>	Contains messages and errors related to the agent functions. For more information, see the <i>Storage Foundation Administrator's Guide</i> .

Table 3-1 List of log files (*continued*)

Log file	Location	Description
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: <pre># /opt/VRTSvcs/vxfen/bin/\ vxfendebug -p</pre>	Contains messages, errors, or diagnostic information for I/O fencing.
Vxfend log	/var/VRTSvcs/log/vxfen/vxfend_A.log	Contains messages, errors, or diagnostic information for the vxfend process.

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

- 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``
- Stop the cluster.

```
# hstop -all
```
- Start the cluster.

```
# hstart
```


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 Sybase CE kernel and driver messages

SF Sybase CE 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-16-20018-301
Sybase:ase:monitor:Open for dataserver 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 151.

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

Entity	Debug logs used
HAD	DBG_AGENT (for agent-related debug logs) DBG_ALERTS (for alert debug logs) DBG_CTEAM (for GCO debug logs) DBG_GAB, DBG_GABIO (for GAB debug messages) DBG_GC (for displaying global counter with each log message) DBG_INTERNAL (for internal messages) DBG_IPM (for Inter Process Messaging) DBG_JOIN (for Join logic) DBG_LIC (for licensing-related messages) DBG_NTEVENT (for NT Event logs) DBG_POLICY (for engine policy) DBG_RSM (for RSM debug messages) DBG_TRACE (for trace messages) DBG_SECURITY (for security-related messages) DBG_LOCK (for debugging lock primitives) DBG_THREAD (for debugging thread primitives) DBG_HOSTMON (for HostMonitor debug logs)

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 Sybase resource type.

```
# hatype -modify Sybase 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"
# hstart
```

```
# export VCS_DEBUG_LOG_TAGS="DBG_AGINFO DBG_AGDEBUG DBG_AGTRACE"
# hstart
```

```
# export VCS_DEBUG_LOG_TAGS="DBG_IPM"
# hagrps -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 Sybase 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 CFSSMount 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

Troubleshooting tips

The following files and command output may be required to determine the source of a problem:

- [Sybase installation error log](#)
- [Veritas log files](#)
- [OS system log](#)
- [GAB port membership](#)

Sybase installation error log

This file contains errors that occurred during installation. It clarifies the nature of the error and exactly when it occurred during the installation.

To check the Sybase installation error log

- 1 Access the following file:

```
$SYBASE/$SYBASE_ASE/install/dataserver_name*.log
```

- 2 Verify if there are any installation errors logged in this file, since they may prove to be critical errors.
- 3 If there are any installation problems, send this file to Tech Support. It is required for debugging the issue.

Veritas log files

The Veritas log file contains all actions performed by `HAD`.

To check the Veritas log files

- 1 Access the following file:

```
/var/VRTSvcs/log/engine_A.log
```

- 2 Verify if there are any `CVM` errors logged in this file, since they may prove to be critical errors.
- 3 Check the `vxconfigd` log file at:

```
/var/adm/vx/vxconfigd.log
```

There are additional log files pertaining to the agents for SF Sybase CE components such as `CVM` and `CFS` in the `/var/VRTSvcs/log` directory, which are also helpful in diagnosing issues.

To check the agent log files for CVM:

```
# /var/VRTSvcS/log/CVMVolDg_A.log
```

To check the agent log files for CFS:

```
# /var/VRTSvcS/log/CFSMount_A.log
```

To check the agent log files for Sybase:

```
# /var/VRTSvcS/log/Sybase_A.log
```

OS system log

OS syslog files can provide valuable information for diagnosing problems. The system log can be checked in the following file:

```
/var/log/messages
```

GAB port membership

To check GAB port membership

Enter the following command:

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

Port b must exist on the local system.

The output resembles this information:

```
GAB Port Memberships
=====
Port a gen 4a1c0001 membership 01
Port b gen ada40d01 membership 01
Port f gen f1990002 membership 01
Port h gen d8850002 membership 01
Port m gen d8860002 membership 01
Port u gen de4f0203 membership 01
Port v gen 1fc60002 membership 01
Port w gen 15ba0002 membership 01
Port y gen 73f449 membership 01
```

[Table 3-2](#) defines each GAB port's function.

For illustration of different GAB ports,

Table 3-2 GAB port function

Port	Function
a	This port is used for GAB internally.
b	This port is used for I/O fencing communications.
f	CFS uses this port for GLM lock and metadata communication.
h	VCS uses this port. VCS communicates the status of resources running on each system to all systems in the cluster.
m	CVM uses this port for SmartIO VxVM cache coherency.
u	CVM uses this port to ship commands from slave node to master node.
v	CVM uses this port for kernel-to-kernel communication.
w	<code>vxconfigd</code> configuration daemon (module for CVM) uses this port for messaging.
y	CVM uses this port for I/O shipping.

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 Sybase CE configuration, upgrade, or when you add a node to the cluster using the installer.

Workaround: To start SF Sybase CE, 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 I/O fencing

The following sections discuss troubleshooting the I/O fencing problems. Review the symptoms and recommended solutions.

The vxfcntlsthdw utility fails when SCSI TEST UNIT READY command fails

While running the `vxfcntlsthdw` 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.

See [“How I/O fencing works in different event scenarios”](#) on page 30.

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 system1:

```
system1> # lltstat -C
57069

system1> # cat /etc/vxfentab
/dev/vx/rmp/disk_7
/dev/vx/rmp/disk_8
/dev/vx/rmp/disk_9

system1> # vxfenadm -s /dev/vx/rmp/disk_7
Reading SCSI Registration Keys...
Device Name: /dev/vx/rmp/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 `vxfcntlpre` command before you use them as coordinator disks in the local cluster.

See [“About the vxfcntlpre utility”](#) on page 81.

Fencing startup reports preexisting split-brain

The vxfs 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.

See [“Clearing preexisting split-brain condition”](#) on page 132.

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 131.

See [“About the vxfcntlclearpre utility”](#) on page 81.

[Table 3-3](#) describes how to resolve a preexisting split-brain condition depending on the scenario you have encountered:

Table 3-3 Recommended solution to clear pre-existing split-brain condition

Scenario	Solution
Actual potential split-brain condition—system 2 is up and system 1 is ejected	<ol style="list-style-type: none"> 1 Determine if system1 is up or not. 2 If system 1 is up and running, shut it down and repair the private network links to remove the split-brain condition. 3 Restart system 1.
Apparent potential split-brain condition—system 2 is down and system 1 is ejected	<ol style="list-style-type: none"> 1 Physically verify that system 2 is down. Verify the systems currently registered with the coordination points. Use the following command for coordinator disks: <pre># vxfenadm -s all -f /etc/vxfentab</pre> 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 <code>vxfcntlclearpre</code> command. The command removes SCSI-3 registrations and reservations. See “About the vxfcntlclearpre utility” on page 81. 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 94.

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 85.

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 vx fencescoorddg 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:

```
# hstop -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 I/O fencing on each node:

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

```
# systemctl stop vx fen
```

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

```
# /etc/init.d/vx fen stop
```

This removes any registration keys on the disks.

4 Import the coordinator disk group. The file /etc/vxfendg includes the name of the disk group (typically, vx fencescoorddg) that contains the coordinator disks, so use the command:

```
# vx dg -t fC 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.

5 To remove disks from the disk group, use the VxVM disk administrator utility, vx diskadm.

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
```

6 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 5, then create the disk group again and add the new disk to it.

See the *Storage Foundation for Sybase ASE CE Configuration Guide* for detailed instructions.

- If the diskgroup already exists, then add the new disk to it.

```
# vxdg -g vxfencoorddg -o coordinator adddisk disk_name
```

7 Test the recreated disk group for SCSI-3 persistent reservations compliance.

See [“Testing the coordinator disk group using the -c option of vxfsenthdw”](#) on page 71.

8 After replacing disks in a coordinator disk group, deport the disk group:

```
# vxdg deport `cat /etc/vxfendg`
```

9 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
```

- 10 Verify that the I/O fencing module has started and is enabled.

```
# gabconfig -a
```

Make sure that port b membership exists in the output for all nodes in the cluster.

```
# vxfenadm -d
```

Make sure that I/O fencing mode is not disabled in the output.

- 11 If necessary, restart VCS on each node:

```
# hstart
```

Troubleshooting Cluster Volume Manager in SF Sybase CE 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 Sybase CE cluster

If you see a message resembling:

```
vxvm:vxconfigd:ERROR:vold_vcs_getnodeid(/dev/vx/rmp/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
system1       1          0           Joined: Master
system2       0          1           Joined: Slave
```

This above output shows that CVM is healthy, with system `system1` 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 Sybase CE 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 82.

Unable to start CVM in SF Sybase CE 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 82.

CVM group is not online after adding a node to the SF Sybase CE cluster

The possible causes for the CVM group being offline after adding a node to the cluster are as follows:

- Other resources configured in the cvm group as critical resources are not online.

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 Sybase CE 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 Sybase CE 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 interconnects

This section discusses troubleshooting interconnect problems.

Network interfaces change their names after reboot

On SUSE systems, network interfaces change their names after reboot even with `HOTPLUG_PCI_QUEUE_NIC_EVENTS=yes` and `MANDATORY_DEVICES="..."` set.

Workaround: Use `PERSISTENT_NAME= ethX` where X is the interface number for all interfaces.

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 Sybase ASE CE

This section discusses troubleshooting Syabase ASE CE.

Sybase private networks

Sybase private networks should be on LLT links.

Sybase instances under VCS control

Sybase instances should be configured under VCS control.

Node does not reboot

Cluster membership mode should be set to "vcs".

Problem: a node does not reboot after a dataserver is killed.

Resolution: check if membership-mode is set to vcs in qrmutil.

```
# qrmutil
--quorum-dev=/quorum/quorum.data --display=config |grep mode
```

Sybase instance not starting

Problem: a Sybase instance that is not starting and is stuck in "VCMP is waiting for vxfend message."

Resolution: restart vxfend:

```
# hares -online vxfend -sys system1
```

Prevention and recovery strategies

This chapter includes the following topics:

- [Prevention and recovery strategies](#)

Prevention and recovery strategies

The following topics are useful diagnostic tools and strategies for preventing and recovering from the various problems that can occur in the SF Sybase CE environment.

Verification of GAB ports in SF Sybase CE cluster

The following ports need to be up on all the nodes of SF Sybase CE cluster:

- port a (GAB)
- port b (I/O fencing)
- port f (CFS)
- port h (VCS)
- port m (CVM - for SmartIO VxVM cache coherency)
- 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 f gen 1ea84702 membership 01
Port h gen cf430b02 membership 01
Port m gen cf430b03 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 Sybase CE cluster, see the *Storage Foundation for Sybase ASE CE 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 Sybase CE 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 lltd
```

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

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

If LLT is configured correctly on each node, the console output displays:

```
LLT INFO V-14-1-10009 LLT Protocol available
```

On a two node cluster, for example `system1` and `system2`, checks you can run to make sure LLT is properly configured:

```
# /sbin/lltdconfig
```

System displays the state of LLT.

```
# cat /etc/llthosts
```

```
0 system1
```

```
1 system2
```

Check the `llttab` on both nodes:

```
# cat /etc/llttab
```

```
set-node system1
```

```
set-cluster Cluster id
```

```
link private_nic1 eth-00:15:17:48:b4:98 - ether - -
link private_nic2 eth-00:15:17:48:b4:99 - ether - -
```

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
```

Check to make sure that GAB is properly configured:

```
# gabconfig -a |grep 'Port b'
```

```
Port b gen      614604 membership 01
```

```
# cat /etc/gabtab
```

```
/sbin/gabconfig -c -n2 <here it 2 as number of nodes are 2)
```

Verifying normal functioning of VCS I/O fencing

It is mandatory to have VCS I/O fencing enabled in SF Sybase CE 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 Sybase CE 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 Sybase CE/SF CFS/SF HA environment, VxVM/CVM registers the keys on data disks, the format of which is ASCII "A" prefixed to the LLT ID of the system followed by the characters "PGRxxxx" where 'xxxx' = i such that the disk group is the ith shared group to be imported.

For example: node 0 uses APGR0001 (for the first imported shared group).

In addition to the registration keys, VCS/CVM also installs a reservation key on the data LUN. There is one reservation key per cluster as only one node can reserve the LUN.

See ["About SCSI-3 Persistent Reservations"](#) on page 28.

The following command lists the keys on a data disk group:

```
# vxdbg list |grep data
```

```
dbdata_dg enabled,shared,cds 1201715530.28.pushover
```

Select the data disk belonging to dbdata_dg:

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

```
sdh auto:cdsdisk    sdh          sybdata_101  online shared
sdj auto:cdsdisk    sdj          sybdata_101  online shared
sdk auto:cdsdisk    sdk          sybdata_101  online shared
```

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 76.

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 `vxfstshdw` (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 vxfstshdw utility”](#) on page 68.

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."

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
```

I/O Fencing kernel logs

I/O Fencing kernel logs contain useful information to troubleshoot intricate I/O fencing issues. The logs can be collected using the following command:

```
# /opt/VRTSvcs/vxfen/bin/vxfendebug -p
```

Tunable parameters

This chapter includes the following topics:

- [About GAB tunable parameters](#)
- [About LLT tunable parameters](#)
- [About VXFEN tunable parameters](#)

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 151.

See [“About GAB run-time or dynamic tunable parameters”](#) on page 152.

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
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 <code>gabconfig -Q</code> option. Use the <code>gabconfig</code> 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 152.	Default: 120000 msec (2 minutes) Range: 160000-240000 (in msec)

Table 5-1 GAB static tunable parameters (*continued*)

GAB parameter	Description	Values (default and range)
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 152.	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
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 <code>gabconfig -K</code> option. Use the <code>gabconfig</code> 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	<p>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 <code>gabtab</code> file, then the cluster will not form. For example: if you type <code>gabconfig -c -n4</code>, then the cluster will not form until all four nodes join the cluster. If this option is enabled using the <code>gabconfig -x</code> command then the node will join the cluster even if the other nodes in the cluster are not yet part of the membership.</p> <p>Use the following command to set the number of nodes that can form the cluster:</p> <pre>gabconfig -n count</pre> <p>Use the following command to enable control port seed. Node can form the cluster without waiting for other nodes for membership:</p> <pre>gabconfig -x</pre>
Halt on process death	<p>Default: Disabled</p> <p>This option controls GAB's ability to halt (panic) the system on user process death. If <code>_had</code> and <code>_hashadow</code> are killed using <code>kill -9</code>, 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.</p> <p>Use the following command to enable halt system on process death:</p> <pre>gabconfig -p</pre> <p>Use the following command to disable halt system on process death:</p> <pre>gabconfig -P</pre>

Table 5-2 GAB dynamic tunable parameters (*continued*)

GAB parameter	Description and command
Missed heartbeat halt	<p>Default: Disabled</p> <p>If this option is enabled then the system will panic on missing the first heartbeat from the VCS engine or the <code>vxconfigd</code> daemon in a CVM environment. The default option is to disable the immediate panic.</p> <p>This GAB option controls whether GAB can panic the node or not when the VCS engine or the <code>vxconfigd</code> 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 (<code>kill_ntries</code> - 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 <code>isolate_time</code> (default value 2 minutes). If the process is still alive, then GAB will PANIC the system.</p> <p>If this option is enabled GAB will immediately HALT the system in case of missed heartbeat from client.</p> <p>Use the following command to enable system halt when process heartbeat fails:</p> <pre>gabconfig -b</pre> <p>Use the following command to disable system halt when process heartbeat fails:</p> <pre>gabconfig -B</pre>

Table 5-2 GAB dynamic tunable parameters (*continued*)

GAB parameter	Description and command
Halt on rejoin	<p>Default: Disabled</p> <p>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.</p> <p>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.</p> <p>Use the following command to enable system halt on rejoin:</p> <pre>gabconfig -j</pre> <p>Use the following command to disable system halt on rejoin:</p> <pre>gabconfig -J</pre>
Keep on killing	<p>Default: Disabled</p> <p>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.</p> <p>Repeat attempts to kill process if it does not die</p> <pre>gabconfig -k</pre>

Table 5-2 GAB dynamic tunable parameters (*continued*)

GAB parameter	Description and command
Quorum flag	<p>Default: Disabled</p> <p>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</p> <p>Enable iofence quorum</p> <pre>gabconfig -q</pre> <p>Disable iofence quorum</p> <pre>gabconfig -d</pre>
GAB queue limit	<p>Default: Send queue limit: 128</p> <p>Default: Recv queue limit: 128</p> <p>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.</p> <p>Set the send queue limit to specified value</p> <pre>gabconfig -Q sendq:value</pre> <p>Set the receive queue limit to specified value</p> <pre>gabconfig -Q rcvq:value</pre>
IOFENCE timeout	<p>Default: 15000(ms)</p> <p>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.</p> <p>Set the iofence timeout value to specified value in milliseconds.</p> <pre>gabconfig -f value</pre>

Table 5-2 GAB dynamic tunable parameters (*continued*)

GAB parameter	Description and command
Stable timeout	<p>Default: 5000(ms)</p> <p>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.</p> <p>Set the stable timeout to specified value</p> <pre>gabconfig -t stable</pre>
Isolate timeout	<p>Default: 120000(ms)</p> <p>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</p> <pre>gabconfig -S isolate_time:value</pre>
Kill_ntries	<p>Default: 5</p> <p>This tunable specifies the number of attempts GAB will make to kill the process by sending SIGABRT signal.</p> <pre>gabconfig -S kill_ntries:value</pre>
Driver state	<p>This parameter shows whether GAB is configured. GAB may not have seeded and formed any membership yet.</p>
Partition arbitration	<p>This parameter shows whether GAB is asked to specifically ignore jeopardy.</p> <p>See the <code>gabconfig</code> (1M) manual page for details on the <code>-s</code> flag.</p>

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.

The tunable parameters are classified into two categories:

- LLT timer tunable parameters
See [“About LLT timer tunable parameters”](#) on page 158.

- LLT flow control tunable parameters
See [“About LLT flow control tunable parameters”](#) on page 162.
- See [“Setting LLT timer tunable parameters”](#) on page 165.

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	<ul style="list-style-type: none"> ■ 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 <code>peertrouble</code> timer value.

Table 5-3 LLT timer tunable parameters (*continued*)

LLT parameter	Description	Default	When to change	Dependency with other LLT tunable parameters
peerinact	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 <code>lltstat -nvv -r</code> 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	<ul style="list-style-type: none"> ■ 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	<ul style="list-style-type: none"> ■ 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	<p>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.</p> <p>If this timer tunable is set to 0, the out of timer context heartbeating mechanism is disabled.</p>	200	<p>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.</p> <p>In some circumstances, when the private networks links are very slow or nodes in the cluster are very busy, increase the value</p>	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	<p>Do not change this value for performance reasons. Lowering the value can result in unnecessary retransmissions/negative acknowledgement traffic.</p> <p>You can increase the value of oos if the round trip time is large in the cluster (for example, campus cluster).</p>	Not applicable
retrans	LLT retransmits a packet if it does not receive its acknowledgement for this timer interval value.	10	<p>Do not change this value. Lowering the value can result in unnecessary retransmissions.</p> <p>You can increase the value of retrans if the round trip time is large in the cluster (for example, campus cluster).</p>	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
rpothighwater	When the number of packets in the receive queue for a port reaches highwater, LLT is flow controlled.	200	<p>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.</p> <p>Lowering the value can result in unnecessary flow controlling the client on peer node.</p>	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 rpothighwater 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	<p>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.</p> <ul style="list-style-type: none"> ■ 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 <p>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.</p>	<p>This flow control value should not be higher than the difference between the highwater flow control value and the lowwater flow control value.</p> <p>The value of this parameter (window) should be aligned with the value of the bandwidth delay product.</p>

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/llttab` 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/llttab` 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/llttab` file:

```
set-flow flowcontrol tunable:value
```

See the `lltconfig(1M)` and `llttab(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	<p>Size of debug log in bytes</p> <ul style="list-style-type: none"> Values <ul style="list-style-type: none"> Default: 524288(512KB) Minimum: 65536(64KB) Maximum: 1048576(1MB)
vxfen_max_delay	<p>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.</p> <p>This value must be greater than the vxfen_min_delay value.</p> <ul style="list-style-type: none"> Values <ul style="list-style-type: none"> Default: 60 Minimum: 1 Maximum: 60
vxfen_min_delay	<p>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.</p> <p>This value must be smaller than or equal to the vxfen_max_delay value.</p> <ul style="list-style-type: none"> Values <ul style="list-style-type: none"> Default: 1 Minimum: 1 Maximum: 60
vxfen_vxfnd_tmt	<p>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.</p> <ul style="list-style-type: none"> Values <ul style="list-style-type: none"> 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	<p>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 <code>vxfenmode</code> file and use the <code>vxfenadm</code> command to check the value. Depending on the <code>vxfen_mode</code>, the GAB delay is calculated as follows:</p> <ul style="list-style-type: none"> For sybase mode: $1000 * (\text{panic_timeout_offst} + \text{vxfen_max_delay})$ For customized mode: $1000 * (\text{panic_timeout_offst} + \max(\text{vxfen_vxwnd_tmt}, \text{vxfen_loser_exit_delay}))$ 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 sybasece service groups on the node.

```
# hagrps -offline sybasece -sys system1

# hagrps -offline binmnt -sys system1
```

- 2 Stop all the applications that are not configured under VCS. Use native application commands to stop the application.

- 3 Stop VCS on all the nodes. Run the following command on each node:

```
# hstop -local
```

- 4 Stop the VxPEN 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
```

- 5 Edit the /etc/sysconfig/vxfen file.

For example, change the entry from:

```
vxfen_min_delay=1
```

to:

```
vxfen_min_delay=30
```

- 6 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
```

- 7 Bring the service groups online.

```
# hagr -online binmnt -sys system1
```

```
# hagr -online sybasece -sys system1
```

- 8 Start all the applications that are not configured under VCS. Use native application commands to start the applications.

- 9 Start VCS.

```
# hstart
```

Error messages

This appendix includes the following topics:

- [About error messages](#)
- [VxVM error messages](#)
- [VXFEN driver error messages](#)

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 A-1](#) contains VxVM error messages that are related to I/O fencing.

Table A-1 VxVM error messages for I/O fencing

Message	Explanation
<code>vold_vcs_getnodeid(disk_path) :</code> 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.
<code>vold_pgr_register(disk_path) :</code> Probably incompatible vxfen driver.	Incompatible versions of VxVM and the vxfen driver are installed on the system. Install the proper version of SF Sybase CE.

VXFEN driver error messages

Table A-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 171.

See [“Node ejection informational messages”](#) on page 172.

Table A-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.	<p>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.</p> <p>This message is preceded with the message "VXFEN: Unable to register with coordinator disk with serial number xxxx."</p>
<p>There exists the potential for a preexisting split-brain.</p> <p>The coordinator disks list no nodes which are in the current membership. However, they, also list nodes which are not in the current membership.</p> <p>I/O Fencing Disabled!</p>	<p>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.</p> <p>See “Clearing preexisting split-brain condition” on page 132.</p>
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.