Veritas Storage Foundation Sybase ASE CE Administrator's Guide

Solaris

Veritas Storage Foundation Sybase ASE CE Administrator's Guide

Documentation version

PN:

Legal Notice

Symantec Corporation
350 Ellis Street

Mountain View, CA 94043

http://www.symantec.com

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

Technical Support

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

Symantec's maintenance offerings include the following:

- A range of support options that give you the flexibility to select the right amount of service for any size organization
- Telephone and Web-based support that provides rapid response and up-to-the-minute information
- Upgrade assurance that delivers automatic software upgrade protection
- Global support that is available 24 hours a day, 7 days a week
- Advanced features, including Account Management Services

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

www.symantec.com/techsupp/

Contacting Technical Support

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

www.symantec.com/techsupp/

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

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

- Product release level
- Hardware information
- Available memory, disk space, and NIC information
- Operating system

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

Licensing and registration

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

www.symantec.com/techsupp/

Customer service

Customer service information is available at the following URL:

www.symantec.com/techsupp/

Customer Service is available to assist with the following types of issues:

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

Maintenance agreement resources

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

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

Additional enterprise services

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

Enterprise services that are available include the following:

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

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

www.symantec.com

Select your country or language from the site index.

Contents

Chapter 1

Overview of Veritas Storage Foundation for Sybase ASE CE

This chapter includes the following topics:

About Veritas Storage Foundation for Sybase ASE CE

Veritas Storage Foundation™ for Sybase® Adaptive Server Enterprise Cluster Edition (SF Sybase CE) by Symantec 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 Symantec 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 Symantec technologies toward this product.

The solution stack comprises the Veritas Cluster Server (VCS), Veritas Cluster Volume Manager (CVM), Veritas Cluster File System (CFS), and Veritas 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:

- Support for file system-based management. SF Sybase CE provides a generic clustered file system technology for storing and managing Sybase data files as well as other application data.
- Use of SFCFS for the Sybase 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 from a single web console.
- 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 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 HBAs and 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)

ASE Cluster Edition is a shared disk cluster implementation of Sybase's flagship enterprise database. ASE is a highly reliable, scalable, and efficient database engine used in mission critical environments such as financial markets, telecommunications networks, and healthcare. ASE CE allows multiple "instances" of the 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.

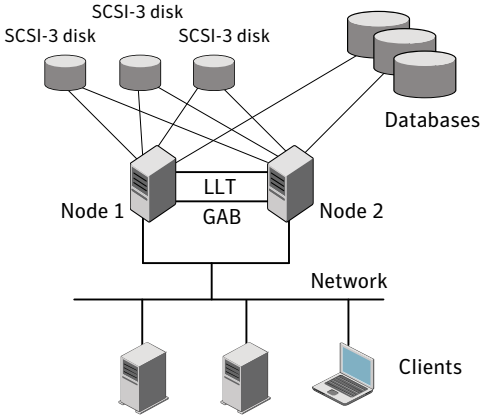
At a conceptual level, SF Sybase CE is a cluster that manages applications (instances), networking, and storage components using resources contained in service groups. SF Sybase CE clusters have the following properties:

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

Figure 1-1 below displays the basic layout and individual components required for a SF Sybase CE installation. This basic layout includes the following components:

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

Figure 1-1 SF Sybase CE basic layout and components

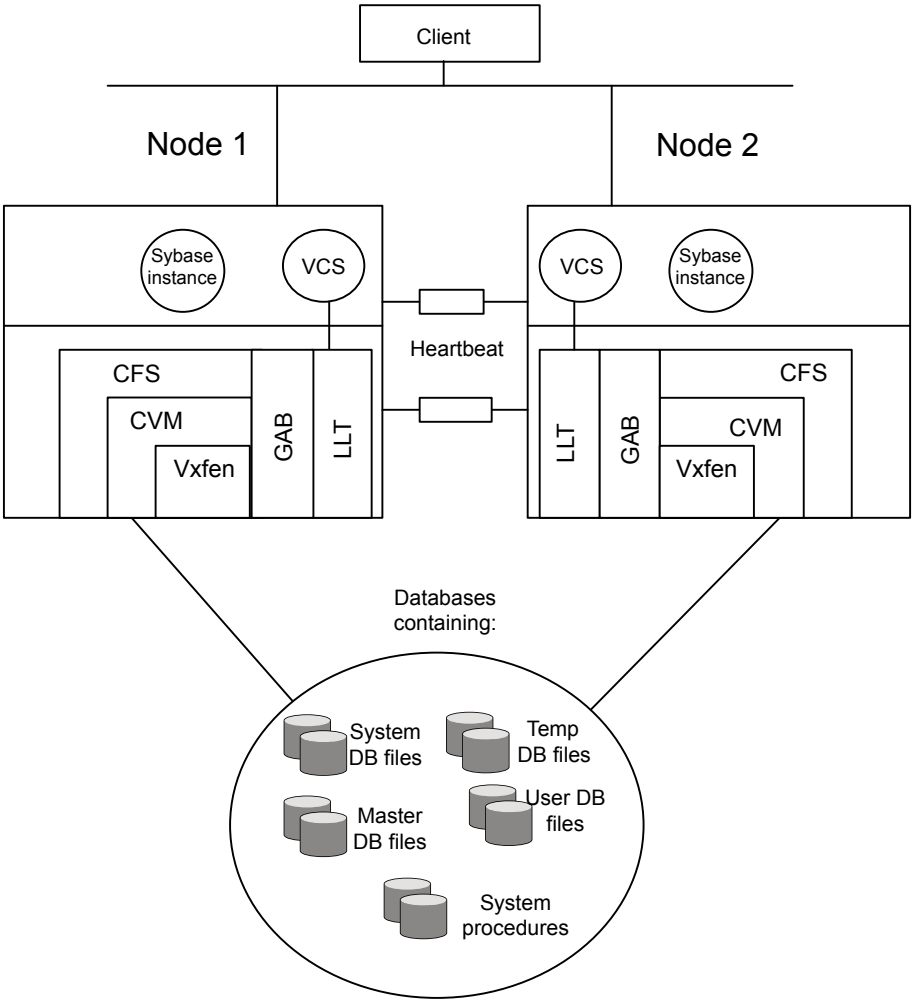


SF Sybase CE adds the following technologies to a failover cluster environment, which are engineered specifically to improve performance, availability, and manageability of Sybase ASE CE environments:

- 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.

How the agent makes Sybase highly available

The agent for Sybase can perform different levels of monitoring and different actions which you can configure. In the basic monitoring mode, the agent detects an application failure if a configured Sybase server process is not running.

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.

In the optional detail monitoring mode, the agent detects application failure if it cannot perform a transaction in the user-provided table in the Sybase database server.

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 22.
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 24.
Cluster Server (VCS)	Uses technology from Veritas Cluster Server to manage Sybase ASE CE databases and infrastructure components. See "Veritas Cluster Server" on page 26.
VXFEN	The VCS module prevents cluster corruption through the use of SCSI3 I/O fencing.
VXFEND	The VXFEN daemon communicates directly with VCMP and relays membership modification messages.
VCMP	VCMP provides interface between Sybase ASE CE cluster and the SF Sybase components.
QRMUTIL	QRMUTIL provides Sybase CE instance status.

Table 1-1 SF Sybase CE component products *(continued)*

Component product	Description
Sybase agent	The VCS agent is responsible for onlining, offlining, and monitoring Sybase ASE. It obtains status by checking for processes, performing SQL queries on a running database, and interacting with QRMUTIL.

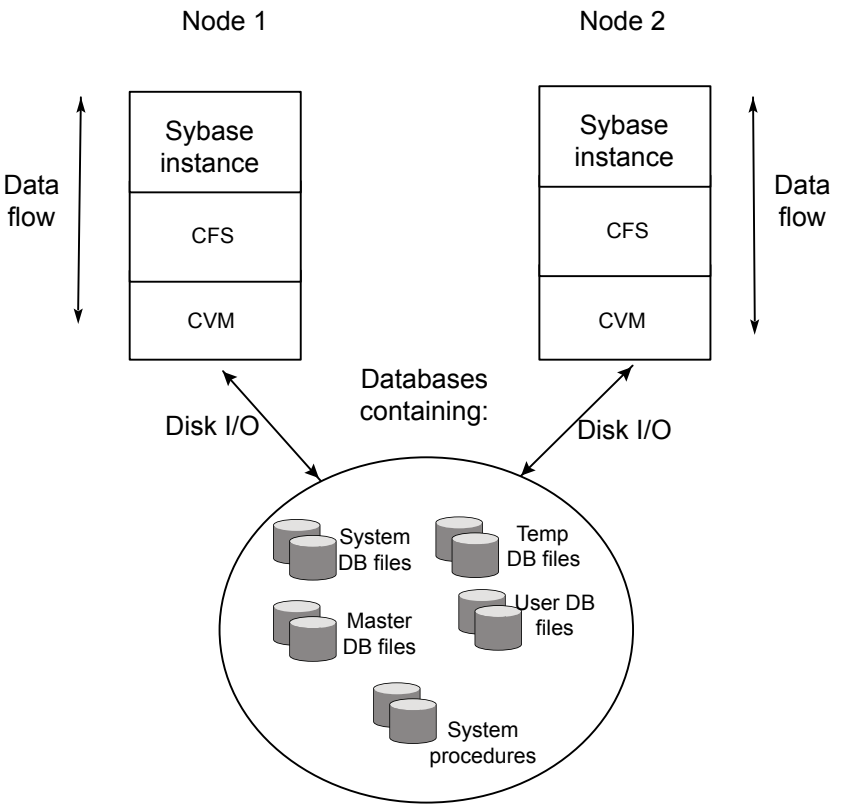
Communication infrastructure

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

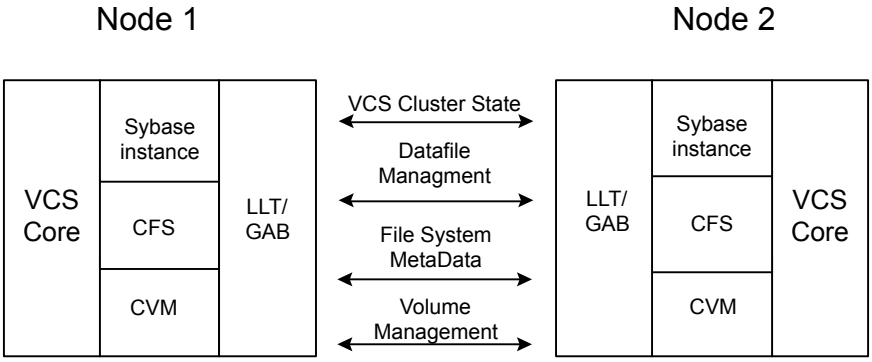


Communication requirements

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:

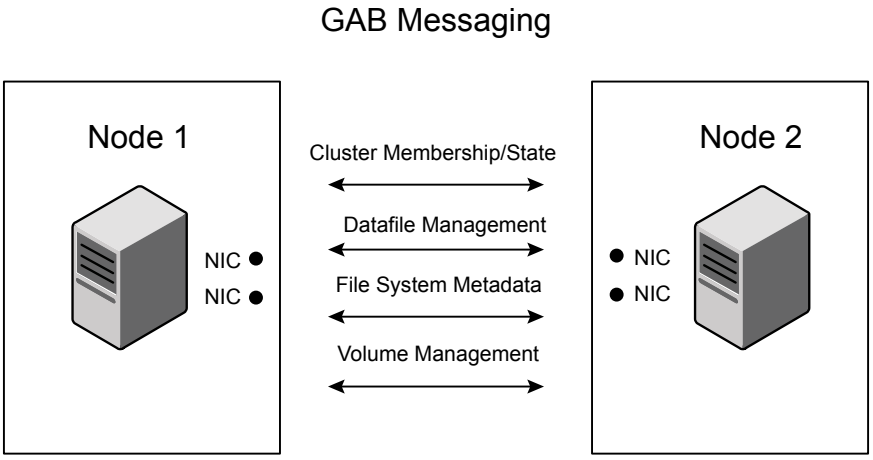
- Traffic distribution
  LLT distributes (load-balances) internode communication across all available cluster interconnect links. All cluster communications are evenly distributed across as many as eight network links for performance and fault resilience. If a link fails, LLT redirects traffic to the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

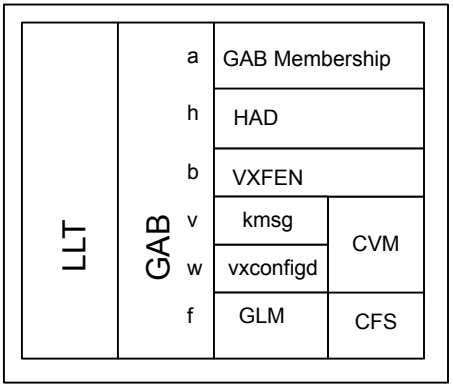
- Cluster communications
  GAB provides reliable cluster communication between SF Sybase CE modules. GAB provides guaranteed delivery of point-to-point messages and broadcast messages to all nodes. Point-to-point messaging involves sending and acknowledging the message. Atomic-broadcast messaging ensures 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 wanting to communicate with a peer process on other nodes registers with GAB on a specific port. This registration enables communication and notification of membership changes. For example, the VCS engine (HAD) registers on port h. HAD receives messages from peer had processes on port h. HAD also receives notification when a node fails or when a peer process on port h becomes unregistered.

Some 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



For additional information about the different GAB ports:

See "GAB port membership" on page 86.

Cluster Volume Manager (CVM)

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

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

CVM architecture

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

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

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

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

See Figure 1-6 on page 21.

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

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

CVM communication

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

See Figure 1-6 on page 21.

CVM communication involves the following GAB ports:

- Port w
  Most CVM communication uses port w for vxconfigd communications. During any change in volume configuration, such as volume creation, plex attachment or detachment, and volume resizing, vxconfigd on the master node uses port w to share this information with slave nodes.
  When all slaves use port w to acknowledge the new configuration as the next active configuration, the master updates this record to the disk headers in the VxVM private region for the disk group as the next configuration.
- Port v
  CVM uses port v for kernel-to-kernel communication. During specific configuration events, certain actions require coordination across all nodes. An example of synchronizing events is a resize operation. CVM must ensure all nodes see the new or old size, but never a mix of size among members.
  CVM also uses this port to obtain cluster membership from GAB and determine the status of other CVM members in the cluster.

CVM recovery

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

Configuration differences with VxVM

CVM configuration differs from VxVM configuration in the following areas:

- Configuration commands occur on the master node.
- Disk groups are created (could be private) and imported as shared disk groups.

- Disk groups are activated per node.
- Shared disk groups are automatically imported when CVM starts.

Cluster File System (CFS)

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

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

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

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 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.

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

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

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

CFS recovery

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

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

applications running at the time of the crash is the responsibility of the applications.

Comparing raw volumes and CFS for data files

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

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

  ```
  # vxedit -g dgname set group=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.

Veritas Cluster Server

Veritas 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 agents.

VCS communication

SF Sybase CE 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.

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.

I/O fencing

I/O fencing is a mechanism to prevent uncoordinated access to the shared storage. This feature works even in the case of faulty cluster communications causing a split-brain condition. Symantec 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 vxfen module 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 VCS later launches VXFEND, 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. If the handshake between VXFEND and VCMP is successful, VXFEND 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).

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 is 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 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. It even blocks a node that is alive is unable to cause damage.

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 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 optional features in SF Sybase CE

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

- About secure SF Sybase CE cluster setup

- About multiple SF Sybase CE cluster management setup using VCS Management Console
- About SF Sybase CE global cluster setup for disaster recovery

About secure SF Sybase CE cluster setup

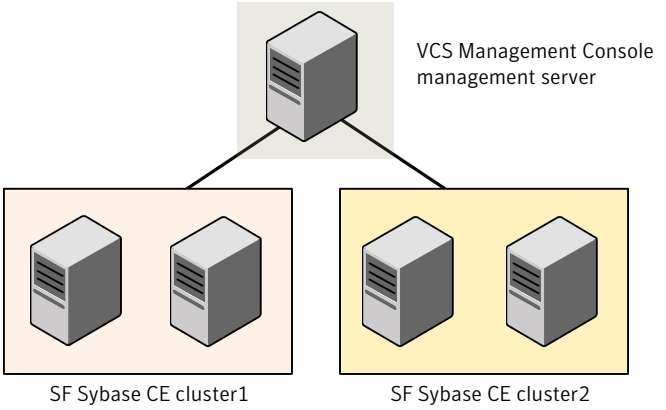
You can set up Authentication Service for the cluster during the installation or after installation. Before you install the authentication service, refer to the *Symantec Product Authentication Service Installation Guide* at the following location on the Veritas software disc:

authentication_service/docs/vxat_install.pdf

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

Figure 1-7 illustrates a sample secure cluster setup.

Figure 1-7 SF Sybase CE secure cluster setup

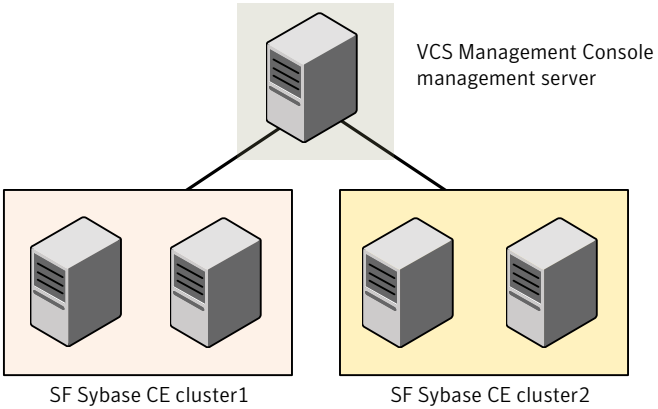


About multiple SF Sybase CE cluster management setup using VCS Management Console

Veritas Cluster Server Management Console is a high availability management solution that enables you to monitor and administer multiple clusters from a single Web console. SF Sybase CE nodes must be discovered by the cluster management console server before you can manage the nodes using the server.

Figure 1-8 illustrates centralized management of SF Sybase CE clusters using Veritas Cluster Server Management Console.

Figure 1-8 Centralized management of SF Sybase CE cluster



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. For updated information, see the TechNote for late-breaking and new information on updates, patches, and software issues regarding this release:

http://entsupport.symantec.com/325604

Chapter 2

Administering SF Sybase CE and its components

This chapter includes the following topics:

Administering SF Sybase CE

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

- Setting the PATH variable
- Setting the MANPATH variable
- Stopping and starting LLT and GAB
- Stopping SF Sybase CE manually on a single node
- Starting SF Sybase CE manually on a single node

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

See "About troubleshooting SF Sybase CE" on page 83.

Setting the MANPATH variable

To set the MANPATH variable for the root user:

For sh, ksh, bash:

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

For csh:

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

Setting the PATH variable

To set the PATH variable for the root user:

For sh, ksh, bash:

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

For csh:

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

To set the PATH variable for the Sybase user:

For sh, ksh, bash:

```
PATH=/opt/sybase/SDCADMIN-15_0/bin:/opt/sybase/ASE-15_0/jobscheduler/bin:
/opt/sybase/ASEP/bin:/opt/sybase/DBISQL/bin:/opt/sybase/UAF-2_5/bin:
/opt/sybase/OCS-15_0/bin:/opt/sybase/ASE-15_0/bin:/opt/sybase/ASE-15_0/install:/usr/bin:
```

For csh:

```
setenv PATH
PATH=/opt/sybase/SDCADMIN-15_0/bin:/opt/sybase/ASE-15_0/jobscheduler/bin:
/opt/sybase/ASEP/bin:/opt/sybase/DBISQL/bin:/opt/sybase/UAF-2_5/bin:
/opt/sybase/OCS-15_0/bin:/opt/sybase/ASE-15_0/bin:/opt/sybase/ASE-15_0/install:/usr/bin:
```

Stopping SF Sybase CE manually on a single node

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

In this procedure, the node is system1.

To stop SF Sybase CE manually on a single node

1. Log in as superuser to the node.

   ```
   su - sybase source $SYBASE_HOME/SYBASE.sh
   ```

2. Stop the Sybase instance.

   ```
   # hares -offline ase -sys system1
   # hares -offline vxfend -sys system1
   ```

3. If any process makes use of CFS mounts and is not configured under VCS, it should be stopped before running "hastop -local" to prevent a hang during "hastop -local".

4. Stop the resource group which contains the quorum, data diskgroups, volumes, and mount points.

   ```
   # hagrp -offline sybasece -sys system1
   ```

5. Stop the binary diskgroups, volumes and mount points used for the Sybase binaries.

   ```
   # hagrp -offline binmnt -sys system1
   ```

6. Stop VCS.

   ```
   # hastop -local
   ```

7. To stop other nodes on the cluster, repeat the steps above for the other nodes in the cluster.

Starting SF Sybase CE manually on a single node

This section describes the procedure for starting SF Sybase CE on a single node within a cluster. This procedure may be required for cluster or node maintenance, cluster or node testing, or for any other user-specific requirement .

In this procedure, the node is system1.

To start SF Sybase CE manually on a single node

1 Log in as superuser to the node.

```
su - sybase source $SYBASE_HOME/SYBASE.sh
```

2 Start VCS.

```
# hastart
```

3 To start other nodes on the cluster, repeat the steps above for the other nodes in the cluster.

Stopping and starting LLT and GAB

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

To stop LLT and GAB

- Run the following in the order below:

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

To start LLT and GAB

- Run the following in the order below:

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

Administering VCS

This section provides instructions for the following VCS administration tasks:

- Viewing available Veritas devices and drivers
- Loading Veritas drivers into memory
- Verifying VCS configuration
- Starting and stopping VCS

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

See "About troubleshooting SF Sybase CE" on page 83.

Viewing available Veritas devices and drivers

To view the available Veritas devices:

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

Sample output:

```
Character devices:
199 VxVM
200 VxSPEC
251 vxfen
252 gab
253 llt
254 vxglm

Block devices:
199 VxVM
201 VxDMP
```

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

For example:

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

```
# modinfo|grep gab -i

gab306 78a86000 49fcd 295 1 gab (GAB device 5.0MP3)
```

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

```
# modinfo|grep vx

vx32 12c98da 3e06e 296 1 vxdmp (VxVM 5.0MP3: DMP Driver)34
    78052000 218520 297 1 vxio (VxVM 5.0MP3 I/O driver)
36 7824a3a8 d10 299 1 vxspec (VxVM 5.0MP3 control/status driv)301 787e316c
    ceb 292 1 vxportal (VxFS 5.0_REV-5.0MP3A25_sol port)
302 7883a000 1d0e47 8 1 vxfs (VxFS 5.0_REV-5.0MP3A25_sol SunO)305 78a2e000
    5636d 294 1 vxfen (VRTS Fence 5.0.1)
308 7857c000 21ecc 300 1 vxglm (VxGLM 5.0MP3 (SunOS 5.9))
```

Loading Veritas drivers into memory

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

To load the VxFS driver into memory:

```
# add_drv vxfs
# modload drv/vxfs
```

Verifying VCS configuration

To verify the VCS configuration:

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

Starting and stopping VCS

To start VCS on each node:

```
# hastart
```

To stop VCS on each node:

```
# hastop -local
```

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

Administering CVM

This section provides instructions for the following CVM administration tasks:

- Listing all the CVM shared disks
- Establishing CVM cluster membership manually
- Manually importing a shared disk group
- Manually deporting a shared disk group
- Manually starting shared volumes
- Evaluating the state of CVM ports
- Verifying if CVM is running in an SF Sybase CE cluster
- Verifying CVM membership state
- Verifying the state of CVM shared disk groups
- Verifying the activation mode
- CVM log files

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

See "Troubleshooting CVM" on page 92.

Listing all the CVM shared disks

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

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

Establishing CVM cluster membership manually

In most cases you do not have to start CVM manually; it normally starts when VCS is started.

Run the following command to start CVM manually:

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

vxclustadm: initialization completed
```

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

To verify whether CVM is started properly:

```
# vxclustadm nidmap
Name          CVM Nid     CM Nid      State
system1        0                0      Joined: Master
system2        1                1      Joined: Slave
```

Manually importing a shared disk group

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

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

Manually deporting a shared disk group

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

```
# vxdg deport diskgroupname
```

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

Manually starting shared volumes

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

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

To verify that the volumes are started, run the following command:

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

Evaluating the state of CVM ports

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

```
# gabconfig -a | egrep "Port [vw]"
```

Verifying if CVM is running in an SF Sybase CE cluster

You can use the following options to verify whether CVM 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:

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

On the master node, the following output is displayed:

```
# vxdctl -c mode

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

On the slave nodes, the following output is displayed:

```
# vxdctl -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 CVM can be verified as follows:

```
# vxclustadm -v nodestate

state: joining
        nodeId=0
        masterId=0
        neighborId=0
        members=0x1
        joiners=0x0
        leavers=0x0
        reconfig_seqnum=0x0
        reconfig: vxconfigd in join
```

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

The `vxdctl -c mode` command indicates whether a node is a CVM master or 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

sybbindg_101 enabled,shared 1052685125.1485.csha3
```

Verifying the activation mode

In an SF Sybase CE cluster, the activation of shared disk group should be set to “shared-write” on each of the cluster nodes.

To verify whether the “shared-write”activation is set:

```
# vxdg list diskgroupname |grep activation

local-activation: shared-write
```

If "shared-write" activation is not set, run the following command:

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

CVM log files

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

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

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

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

Administering CFS

This section describes some of the major aspects of cluster file system administration.

This section provides instructions for the following CFS administration tasks:

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

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

Adding CFS file systems to VCS configuration

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

```
# cfsmntadm add quorum_101 quorumvol /quorum sybasece \
all=suid,rw
```

```
Mount Point is being added...
  /quorum added to the cluster-configuration
```

Using cfsmount to mount CFS file systems

To mount a CFS file system using cfsmount:

```
# cfsmount /quorum
Mounting...
[/dev/vx/rdsk/quorum_101/quorum
mounted successfully at /quorumvol on system1
[/dev/vx/rdsk/quorum_101/quorumvol]
mounted successfully at /quorum on system2
```

Resizing CFS file systems

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

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

where sybbindg is the CVM disk group,sybbinvol is the volume, and +2G indicates the increase in volume by 2 Gigabytes.

Verifying the status of CFS file systems

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

```
# cfscluster status

  Node             :  system2
  Cluster Manager  :  not-running
  CVM state        :  not-running
  MOUNT POINT    SHARED VOLUME  DISK GROUP        STATUS

  /quorum       quorumvol       quorum_101        NOT MOUNTED
  /sybase        sybbinvol      sybbindg           NOT MOUNTED
  /sybdata       sybvol         sybdata_101        NOT MOUNTED

  Node             :  system1
  Cluster Manager  :  running
  CVM state        :  running
  MOUNT POINT    SHARED VOLUME  DISK GROUP        STATUS
```

```
  /quorum       quorumvol       quorum_101        MOUNTED
/sybase         sybbinvol      sybbindg           MOUNTED
/sybdata        sybvol         sybdata_101        MOUNTED
```

Verifying CFS port

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

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

CFS agent log files

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

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

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

Storage Foundation Cluster File System commands

Table 2-1 describes the SFCFS commands.

Table 2-1 SFCFS commands

Commands	Description
cfscluster	Cluster configuration command
cfsmntadm	Adds, deletes, modifies, and sets policy on cluster mounted file systems
cfsdgadm	adds or deletes shared disk groups to/from a cluster configuration
cfsmount	mounts a cluster file system on a shared volume
cfsumount	unmounts a cluster file system on a shared volume

mount

The `mount` command with the `-o cluster` option lets you access shared file systems.

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

mount and fsclusteradm commands

The `mount` and `fsclustadm` commands are important for configuring cluster file systems.

fsclustadm

The `fsclustadm` command reports various attributes of a cluster file system. Using `fsclustadm` you can show and set the primary node in a cluster, translate node IDs to host names and vice versa, list all nodes that currently have a cluster mount of the specified file system mount point, and determine whether a mount is a local or cluster mount. The `fsclustadm` command operates from any node in a cluster on which the file system is mounted, and can control the location of the primary for a specified mount point.

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

fsadm

The `fsadm` command can be invoked from the primary or secondary node.

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

Run commands safely in a cluster environment

Any UNIX command that can write to a raw device must be used carefully in a shared environment to prevent data from being corrupted. For shared VxVM volumes, SFCFS provides protection by reserving the volumes in a cluster to prevent VxFS commands, such as `fsck` and `mkfs`, from inadvertently damaging a mounted file system from another node in a cluster. However, commands such as `dd` execute without any reservation, and can damage a file system mounted from another node. Before running this kind of command on a file system, be sure the file system is not mounted on a cluster. You can run the `mount` command to see if a file system is a shared or local mount.

Time synchronization for Cluster File Systems

SFCFS requires that the system clocks on all nodes are synchronized using some external component such as the Network Time Protocol (NTP) daemon. If the nodes are not in sync, timestamps for creation (`ctime`) and modification (`mtime`) may not be consistent with the sequence in which operations actually happened.

The fstab file

In the `/etc/vfstab` file, do not specify any cluster file systems to mount-at-boot because mounts initiated from `vfstab` occur before cluster configuration begins. For cluster mounts, use the VCS configuration file to determine which file systems to enable following a reboot.

Distribute the load on a cluster

Distributing the workload in a cluster provides performance and failover advantages.

For example, if you have eight file systems and four nodes, designating two file systems per node as the primary would be beneficial. Primaryship is determined by which node first mounts the file system. You can also use the `fsclustadm` to designate a SFCFS primary. The `fsclustadm setprimary` command can also define the order in which primaryship is assumed if the current primary fails. After setup, the policy is in effect as long as one or more nodes in the cluster have the file system mounted.

GUIs

Use the Veritas Enterprise Administrator (VEA) for various VxFS functions such as making and mounting file systems, on both local and cluster file systems.

With SFCFS HA, you can use the VCS Cluster Manager GUI to configure and monitor SFCFS. The VCS GUI provides log files for debugging LLT and GAB events.

Administering I/O fencing

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

- About vxfenadm utility
- About vxfenclearpre utility
- About vxfenswap utility

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

See "Troubleshooting I/O fencing" on page 87.

About I/O fencing

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

The fencing operation determines the following:

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

This decision prevents possible data corruption. The installsfsybasece installs the SF Sybase CE I/O fencing driver, VRTSvxfen. If you want to protect data on shared disks, you must configure I/O fencing after you install and configure SF Sybase CE.

See *Veritas Storage Foundation for Sybase ASE CE Installation and Configuration Guide*.

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

See "About preventing data corruption with I/O fencing" on page 30.

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
Coordination points	Act as a global lock during membership changes

About data disks

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

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

About coordination points

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

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

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

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

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

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

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

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

About I/O fencing utilities

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

vxfentsthdw	Tests hardware for I/O fencing Path: /opt/VRTSvcs/vxfen/bin/vxfentsthdw See "About vxfentsthdw utility" on page 51.
vxfenconfig	Configures and unconfigures I/O fencing Checks the list of coordinator disks used by the vxfen driver. Path: /sbin/vxfenconfig
vxfenadm	Displays information on I/O fencing operations and manages SCSI-3 disk registrations and reservations for I/O fencing Path: /sbin/vxfenadm See "About vxfenadm utility" on page 60.
vxfenclearpre	Removes SCSI-3 registrations and reservations from disks Path: /opt/VRTSvcs/vxfen/bin/vxfenclearpre See "About vxfenclearpre utility" on page 62.
vxfenswap	Replaces coordinator disks without stopping I/O fencing Path: /opt/VRTSvcs/vxfen/bin/vxfenswap See "About vxfenswap utility" on page 64.
vxfendisk	Generates the list of paths of disks in the diskgroup. This utility requires that Veritas Volume Manager is installed and configured. Path: /opt/VRTSvcs/vxfen/bin/vxfendisk

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

About 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 *Veritas Storage Foundation for Sybase ASE CE Installation and Configuration Guide* to set up I/O fencing.

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

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

About general guidelines for using vxfentsthdw utility

Review the following guidelines to use the vxfentsthdw utility:

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

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

- The two nodes must have ssh (default) or rsh communication. If you use rsh, launch the vxfentsthdw utility with the `-n` option.

  After completing the testing process, you can remove permissions for communication and restore public network connections.

- To ensure both systems are connected to the same disk during the testing, you can use the `vxfenadm -i` *`diskpath`* command to verify a disk's serial number.

  See "Verifying that the nodes see the same disk" on page 62.

- For disk arrays with many disks, use the `-m` option to sample a few disks before creating a disk group and using the `-g` option to test them all.

- When testing many disks with the `-f` or the `-g` option, you can review results by redirecting the command output to a file.

- The utility indicates a disk can be used for I/O fencing with a message resembling:

  ```
  The disk /dev/rdsk/c1t1d0s2 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.

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

About the vxfentsthdw command options

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

Table 2-2 vxfentsthdw options

vxfentsthdw option	Description	When to use
-n	Utility uses rsh for communication.	Use when rsh is used for communication.
-r	Non-destructive testing. Testing of the disks for SCSI-3 persistent reservations occurs in a non-destructive way; that is, there is only testing for reads, not writes. May be used with -m, -f, or -g options.	Use during non-destructive testing. See "Performing non-destructive testing on the disks using the -r option" on page 56.
-t	Testing of the return value of SCSI TEST UNIT (TUR) command under SCSI-3 reservations. A warning is printed on failure of TUR testing.	When you want to perform TUR testing.
-d	Use DMP devices. May be used with -c or -g options.	With the -d option, the script picks the DMP paths for disks in the diskgroup. By default, the script uses the -w option to pick up the OS paths for disks in the disk group.
-w	Use raw devices. May be used with -c or -g options.	By default, the script picks up the raw paths for disks in the diskgroup. If you want the script to use the raw paths for disks in the diskgroup, use the -d option.
-c	Utility tests the coordinator disk group prompting for systems and devices, and reporting success or failure.	For testing disks in coordinator disk group. See "Testing the coordinator disk group using vxfentsthdw -c option" on page 54.
-m	Utility runs manually, in interactive mode, prompting for systems and devices, and reporting success or failure. May be used with -r and -t options. -m is the default option.	For testing a few disks or for sampling disks in larger arrays. See "Testing the shared disks using the vxfentsthdw -m option" on page 56.

Table 2-2 vxfentsthdw options *(continued)*

vxfentsthdw option	Description	When to use
-f *filename*	Utility tests system/device combinations listed in a text file. May be used with -r and -t options.	For testing several disks. See "Testing the shared disks listed in a file using the vxfentsthdw -f option" on page 58.
-g *disk_group*	Utility tests all disk devices in a specified disk group. May be used with -r and -t options.	For testing many disks and arrays of disks. Disk groups may be temporarily created for testing purposes and destroyed (ungrouped) after testing. See "Testing all the disks in a disk group using the vxfentsthdw -g option" on page 59.

Testing the coordinator disk group using vxfentsthdw -c option

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 /dev/rdsk/c1t1d0s2, /dev/rdsk/c2t1d0s2, and /dev/rdsk/c3t1d0s2.
- From the node system2, the same disks are seen as /dev/rdsk/c4t1d0s2, /dev/rdsk/c5t1d0s2, and /dev/rdsk/c6t1d0s2.

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

To test the coordinator disk group using vxfentsthdw -c

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

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

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

```
Enter the first node of the cluster: 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/rdsk/c1t1d0s2 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/rdsk/c4t1d0s2 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 *Veritas Volume Manager Administrator's Guide*.

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

See the *Veritas Storage Foundation for Sybase ASE CE Installation and Configuration Guide* for instructions to initialize disks for I/O fencing and to set up coordinator disk groups.

If necessary, start the disk group.

See the *Veritas Volume Manager Administrator's Guide* for instructions to start the disk group.

3 Retest the disk group.

See "Testing the coordinator disk group using vxfentsthdw -c option" on page 54.

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

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

To perform non-destructive testing on disks

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

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

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

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

Testing the shared disks using the vxfentsthdw -m option

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

This procedure uses the /dev/rdsk/c1t1d0s2 disk in the steps.

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

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

```
The disk /dev/rdsk/c1t1d0s2 is ready to be configured for
I/O Fencing on node system1
```

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

To test disks using vxfentsthdw script

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

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

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

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

   Do you still want to continue : [y/n] (default: n) y
   Enter the first node of the cluster: 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/cxtxdxsx
        for raw: /dev/rdsk/cxtxdxsx
/dev/rdsk/c2t13d0s2
Make sure it's the same disk as seen by nodes system1 and system2

Enter the disk name to be checked for SCSI-3 PGR on node
system2 in the format:
        for dmp: /dev/vx/rdmp/cxtxdxsx
        for raw: /dev/rdsk/cxtxdxsx
Make sure it's the same disk as seen by nodes system1 and system2
/dev/rdsk/c2t13d0s2
```

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

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

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

```
ALL tests on the disk /dev/rdsk/c1t1d0s2 have PASSED
The disk is now ready to be configured for I/O Fencing on node
system1
```

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

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

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

To test the shared disks listed in a file

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

```
system1 /dev/rdsk/c2t2d1s2 system2 /dev/rdsk/c3t2d1s2
system1 /dev/rdsk/c2t2d1s2 system2 /dev/rdsk/c3t2d1s2
```

where the first disk is listed in the first line and is seen by system1 as /dev/rdsk/c2t2d1s2 and by system2 as /dev/rdsk/c3t2d1s2. The other disk, in the second line, is seen as /dev/rdsk/c2t2d2s2 from system1 and /dev/rdsk/c3t2d2s2 from system2. Typically, the list of disks could be extensive.

2 To test the disks, enter the following command:

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

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

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

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

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

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

For example:

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

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

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

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

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

To test all the disks in a diskgroup

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

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

   The utility reports the test results one disk at a time.
3. To redirect the test results to a text file for review, enter the following command:

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

Testing a disk with existing keys

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

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

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

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

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

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

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

About vxfenadm utility

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

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

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

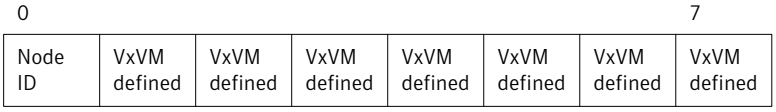
Refer to the `vxfenadm`(1m) manual page for a complete list of the command options.

About the registration key format

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

Figure 2-1 depicts the format of the key.

Figure 2-1 I/O fencing key format



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

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

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

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

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

Verifying that the nodes see the same disk

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

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

To verify that the nodes see the same disks

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

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

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

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

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

```
# vxfenadm -i /dev/rdsk/c2t1d0s2

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

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

About vxfenclearpre utility

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

See "Removing preexisting keys" on page 63.

Removing preexisting keys

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

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

To clear keys after split brain

1 Stop VCS 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:

```
# /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:

```
# cd /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...

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

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

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

7 Restart all nodes in the cluster.

About vxfenswap utility

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

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

See *Veritas Storage Foundation for Sybase ASE CE Installation and Configuration Guide* for details on the coordinator disk requirements.

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

- The disk becomes defective or inoperable and you want to switch to a new diskgroup.
  See "Replacing I/O fencing coordinator disks when the cluster is online" on page 65.
  See "Replacing the coordinator diskgroup in a cluster that is online" on page 68.
  If you want to replace the coordinator disks when the cluster is offline, you cannot use the vxfenswap utility. You must manually perform the steps that the utility does to replace the coordinator disks.
  See "Replacing defective disks when the cluster is offline" on page 90.
- You want to switch the disk interface between raw devices and DMP devices.
  See "Changing the disk interaction policy in a cluster that is online" on page 69.
- The keys that are registered on the coordinator disks are lost.
  In such a case, the cluster might panic when a split-brain occurs. You can replace the coordinator disks with the same disks using the 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 70.

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

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

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

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

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

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

2 Make sure that the cluster is online.

```
# vxfenadm -d
I/O Fencing Cluster Information:
================================
Fencing Protocol Version: 201
Fencing Mode: SCSI3
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 Import the coordinator disk group.

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

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

where:

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

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

-C specifies that any import locks are removed.

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

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

- Verify whether the coordinator attribute is set to on.

```
# vxdg list vxfencoorddg |  grep flags: | grep coordinator
```

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

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

- Destroy the disk group.

```
# vxdg destroy vxfencoorddg
```

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

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

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

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

See the *Veritas Storage Foundation for Sybase ASE CE Installation and Configuration Guide* for more information.

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

Do one of the following:

- If you use ssh for communication:

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

- If you use rsh for communication:

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

The utility performs the following tasks:

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

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

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

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

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

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

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

The vxfenswap utility rolls back the disk replacement operation.

You must manually restore the old diskgroup using the VxVM disk administrator utility.

Replacing the coordinator diskgroup in a cluster that is online

You can also replace the coordinator diskgroup using the vxfenswap utility.

To replace the coordinator diskgroup

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

2 Make sure that the cluster is online.

```
# vxfenadm -d
I/O Fencing Cluster Information:
================================
Fencing Protocol Version: 201
Fencing Mode: SCSI3
Fencing SCSI3 Disk Policy: dmp
Cluster Members:
  * 0 (system1)
  1 (system2)
RFSM State Information:
  node 0 in state 8 (running)
  node 1 in state 8 (running)
```

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

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

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

```
# vxdisk -o alldgs list

DEVICE    TYPE    DISK  GROUP    STATUS
c4t0d1    auto:cdsdisk    -  (vxfendg)    online
c4t0d2    auto:cdsdisk    -  (vxfendg)    online
c4t0d3    auto:cdsdisk    -  (vxfendg)    online
c4t0d4    auto:cdsdisk    -  (vxfencoorddg)    online
c4t0d5    auto:cdsdisk    -  (vxfencoorddg)    online
c4t0d6    auto:cdsdisk    -  (vxfencoorddg)    online
```

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

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

6 Verify that the coordinator disk group has changed.

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

Changing the disk interaction policy in a cluster that is online

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

To change the disk interaction policy

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

2 Make sure that the cluster is online.

```
# vxfenadm -d
I/O Fencing Cluster Information:
================================
Fencing Protocol Version: 201
Fencing Mode: SCSI3
Fencing SCSI3 Disk Policy: dmp
Cluster Members:
  * 0 (system1)
  1 (system2)
RFSM State Information:
  node 0 in state 8 (running)
  node 1 in state 8 (running)
```

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

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

4 From any node, start the vxfenswap utility:

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

5 Verify the change in the disk policy.

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

Refreshing lost keys on coordinator disks

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

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

To refresh lost keys on coordinator disks

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

2 Make sure that the cluster is online.

```
# vxfenadm -d
I/O Fencing Cluster Information:
================================
Fencing Protocol Version: 201
Fencing Mode: SCSI3
Fencing SCSI3 Disk Policy: dmp
Cluster Members:
  * 0 (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 -g all -f /etc/vxfentab
Device Name: /dev/vx/rdmp/c1t1d0s2
Total Number of Keys: 0
No keys...
...
```

4 On any node, run the following command to start the vxfenswap utility:

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

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

```
# vxfenadm -g all -f /etc/vxfentab
Device Name: /dev/vx/rdmp/c1t1d0s2
Total Number of Keys: 4
...
```

About VXFEN tunable parameters

The section describes the VXFEN tunable parameters and how to reconfigure the VXFEN module.

Table 2-3 describes the tunable parameters for the VXFEN driver.

Table 2-3 VXFEN tunable parameters

vxfen Parameter	Description and Values: Default, Minimum, and Maximum
vxfen_debug_sz	Size of debug log in bytes ■ Values Default: 65536 Minimum: 65536 Maximum: 256K
vxfen_max_delay	Specifies the maximum number of seconds that the smaller sub-cluster waits before racing with larger sub-clusters for control of the coordinator disks when a split brain occurs. This value must be greater than the vxfen_max_delay value. ■ Values Default: 60 Minimum: 1 Maximum: 600
vxfen_min_delay	Specifies the minimum number of seconds that the smaller sub-cluster waits before racing with larger sub-clusters for control of the coordinator disks when a split brain occurs. This value must be smaller than the vxfen_max_delay value. ■ Values Default: 1 Minimum: 1 Maximum: 600

In the event of a network partition, the smaller sub-cluster delays before racing for the coordinator disks. The time delayed allows a larger sub-cluster to win the race for the coordinator disks. The vxfen_max_delay and vxfen_min_delay parameters define the delay in seconds.

Configuring the VXFEN module parameters

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 Solaris node, edit the file /kernel/drv/vxfen.conf to change the value of the vxfen driver tunable global parameters, vxfen_max_delay and vxfen_min_delay.

Note: You must restart the VXFEN module to put any parameter change into effect.

To configure the VxFEN parameters and reconfigure the VxFEN module

1 Edit the file /kernel/drv/vxfen.conf to change the vxfen_min_delay value to 30.

The following VXFEN example displays the content of the default file /kernel/drv/vxfen.conf before changing the vxfen_min_delay parameter:

```
#
# VXFEN configuration file
#
name="vxfen" parent="pseudo" instance=0 dbg_log_size=65536
vxfen_max_delay=60 vxfen_min_delay=1;
```

After editing the file to change the vxfen_min_delay value to 30, the default file /kernel/drv/vxfen.conf contains the following values:.

```
#
# VXFEN configuration file
#
name="vxfen" parent="pseudo" instance=0 dbg_log_size=65536
vxfen_max_delay=60 vxfen_min_delay=30;
```

After reviewing the edits that you made to the default file, close and save the file.

2 Shut down all sybasece service groups on the node.

```
# hagrp -offline sybasece -sys system1

# hagrp -offline binmnt -sys system1
```

3 Unconfigure the VXVEN module:

```
# /sbin/vxfenconfig -U
```

4 Determine the VXFEN module ID:

```
# /usr/sbin/modinfo | grep -i vxfen
```

The module ID is the number in the first column of the output.

5 Unload the VXFEN module, using the module ID you determined:

```
# /usr/sbin/modunload -i
   module_ID
```

6 For a system running Solaris 10, run the `update_drv` command to re-read the /kernel/drv/vxfen.conf file.

```
# /usr/sbin/update_drv vxfen
```

Note: The `modunload` command has often been used on driver modules to force the system to reread the associated driver configuration file. While this procedure and command works in Solaris 9, this behavior may fail in later releases. The supported method for rereading the driver configuration file for systems running Solaris 10 is through the `update_drv` command. For additional information, refer to update_drv(1M).

---

7 Configure the VXFEN module:

```
# /sbin/vxfenconfig -c
```

8 Bring the service groups online.

```
# hagrp -online binmnt -sys system1

# hagrp -online sybasece -sys system1
```

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 SQL server, monitors the server processes, and shuts down the server.

Table 2-4 lists the Sybase agent for SQL server functions.

Table 2-4 Sybase agent for SQL server functions

Agent operation	Description
Online	Starts the Sybase SQL server by using the following command. `startserver -f $SYBASE/$SYBASE_ASE/install/RUN_$Server` where $Server is the instance_name
Monitor	In the basic monitoring mode, the agent scans process table for the dataserver process. In detail monitoring mode, the agent runs the script that is specified in Monscript as an option. 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.
Offline	Stops the Sybase SQL server by using the `isql` command in the following manner. The agent first executes the `shutdown with wait` command. If it does not bring down the Sybase SQL server, the offline script issues `kill -15` to the dataserver process.
Clean	Forcefully stops the Sybase SQL server by using the `isql` command in the following manner. The agent first executes the `shutdown with wait` command. If it does not bring down the Sybase SQL server, the offline script issues `kill -15` to dataserver process, and as the last option, the agent sends `kill -9` signal to dataserver process.

Monitoring options for the Sybase agent

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

In the basic monitoring mode, the agent for Sybase monitors the dataserver process to verify whether it is running.

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.

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 instance ‘ase1’. The node will automatically rejoin the cluster after reboot.

In the detail monitoring mode, the agent performs a transaction on a table (provided by the user) in the database to ensure that Sybase functions properly. The agent uses this table for internal purposes. Symantec recommends that you do not perform any other transaction on this 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 umounted, 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/tmp/sybase_shm/$Server.

In the $SYBASE/$SYBASE_ASE/install directory, edit the RUN_$Server file. Change the location of the directory that stores shared memory files to /var/tmp/sybase_shm/$Server using the -M option.

For example, the file RUN_$Server resembles the following before the change:

```
/sybase/ASE-15_0/bin/dataserver \

--instance=ase1 \

-d/sybdata/master.dat \

-e/sybase/ASE-15_0/install/ase1.log \

-c/sybase/ASE-15_0/ase1.cfg \

-M/sybase/ASE-15_0 \
```

After the replacement, the file resembles:

```
/sybase/ASE-15_0/bin/dataserver \

--instance=ase1 \

-d/sybdata/master.dat \

-e/sybase/ASE-15_0/install/ase1.log \

-c/sybase/ASE-15_0/ase1.cfg \

-M/var/tmp/sybase_shm/ase1 \
```

Here ase1 is the Adaptive server name.

Note: Make sure you create the /var/tmp/sybase_shm/$Server directory with proper permissions.

Configuring the service group from Cluster Manager (Java console)

A template for the Sybase resource groups is automatically installed with the Veritas agent for Sybase. Using the VCS cluster Manager (Java console), you can view the template, which displays the Sybase service group, its resources and their attributes. You can dynamically modify the attributes' values as necessary for your configuration.

To configure a service group from the Java console

1. Make sure that the Sybase type definition file SybaseTypes.cf is imported in your configuration.
2. Launch the Cluster Configuration wizard using any of the following ways:
   - From the Cluster Explorer menu, select **Tools > Configuration Wizard**.
   - If no service groups exist on the system, Cluster Explorer prompts you to launch the Cluster Configuration wizard. Click **Yes** when prompted.

   The Loading Templates Information window appears, and launches the wizard.
3. Review the information in the Welcome dialog box and click **Next**.
4. Specify the name of the service group and the target systems on which the service group is configured.
   - Enter the name of the service group.

- From the **Available Systems** box, select the systems on which to configure the service group.
- Click the right arrow to move the selected systems to the **Systems for Service Group** box. To remove a system from the box, select the system and click the left arrow.
- Specify system priority for the service group. System priority is numbered sequentially, with the lowest assigned number denoting the highest priority.
- Select the **Service Group Type** as Parallel and click **Next**.

5 On the **Would you like to use a template to configure the service group?** dialog box, click **Next** to configure the service group using a template.

6 Select the **SybaseGroup** template to configure a Sybase service group.

If applicable, a window opens notifying that names of some resources within the new service group are already in use. Resolve the name clashes, if any and click **Next**.

7 Click **Next** to create the service group that is based on the selected template.

A progress indicator displays the percentage of the commands that are executed to create the service group. The actual commands are displayed at the top of the indicator.

8 After the service group is created, click **Next** to edit the attributes for the resources.

The left pane in the dialog box lists all the resources for the Sybase service group. Select a resource from the left pane to list the attributes on the right pane. The attributes in bold denote mandatory attributes. You can modify the attribute values as given in the procedure that follows .

9 Click **Finish** to accept the default values and complete the configuration.

To edit the Sybase service group resource attributes

1 Select the resource from the list on the left pane. The resource attributes appear in the right pane.

2 Select the attribute to be modified and click the edit icon in the **Edit** column.

3 In the Edit Attribute dialog box, enter the attribute values. To modify the scope of the attribute, choose the **Global** or **Local** option.

4 Click **OK**.

5 Repeat the procedure for each resource and click **Finish**. Edit the attributes for all the resources according to your configuration.

6 Follow the wizard instructions to complete the configuration. Click **Finish** to quit the wizard.

Caution: For added security, you must always provide a secure value for passwords.

Configuring the service group using the command line

The Veritas 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 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. Use the file /etc/VRTSagents/ha/conf/Sybase/sample_sybasece_main.cf for reference.

- Create a Sybase service group.
- Create the Sybase resource.
- 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 /etc/VRTSagents/ha/conf/Sybase/sample_sybasece_main.cf. See the *Veritas 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`.

```
# hacf -verify config
```

6 Start VCS on local node.

```
# hastart
```

7 Start VCS on other nodes.

8 Verify that all Sybase service group resources are brought online.

```
# hagrp -state
```

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

```
# hagrp -offline service_group -sys system_name

# hagrp -state
```

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

```
# hagrp -online service_group -sys system_name

# hagrp -state
```

11 On all systems, review 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. Note that in the initial few cycles of taking a service group offline, the agent's memory usage can spike.

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), Cluster Manager (Web Console), Veritas Cluster Management Console, and the command line.

See the *Veritas Cluster Server User's Guide* for more information.

Viewing the agent log for Sybase

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

/var/VRTSvcs/log/engine_A.log

/var/VRTSvcs/log/Sybase_A.log

Chapter 3

Troubleshooting SF Sybase CE

This chapter includes the following topics:

About troubleshooting SF Sybase CE

SF Sybase CE contains several component products, and as a result can be affected by any issue with component products. The first step in case of trouble should be to identify the source of the problem. It is rare to encounter problems in SF Sybase CE itself; more commonly the problem can be traced to setup issues or problems in component products.

Use the information in this chapter to diagnose the source of problems. Indications may point to SF Sybase CE set up or configuration issues, in which case solutions are provided wherever possible. In cases where indications point to a component product or to Sybase as the source of a problem, it may be necessary to refer to the appropriate documentation to resolve it.

Running scripts for engineering support analysis

Troubleshooting scripts gather information about the configuration and status of your cluster and its modules. The scripts identify package information,

debugging messages, console messages, and information about disk groups and volumes. Forwarding the output of these scripts to Symantec Tech Support can assist with analyzing and solving any problems.

getsfsybasece

The getsfsybasece script gathers information about the SF Sybase CE modules. Two files contain output from the script: /tmp/sybaselogs.*sys_name*.tar.Z and /tmp/sybase.out.

To use the getsfsybasece script, on each system enter:

```
# /opt/VRTSvcs/bin/getsfsybasece -local
```

getcomms

The getcomms script gathers information about the GAB and LLT modules. The file /tmp/commslog.*time_stamp*.tar contains the script's output.

To use the getcomms script, on each system enter:

```
# /opt/VRTSgab/getcomms -local
```

hagetcf

The hagetcf script gathers information about the VCS cluster and the status of resources. The output from this script is placed in a tar file, /tmp/vcsconf.*sys_name*.tar.gz, on each cluster system.

To use the hagetcf script, on each system enter:

```
# /opt/VRTSvcs/bin/hagetcf
```

Troubleshooting tips

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

- Sybase installation error log
- Veritas log file
- 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/cluster_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 file

The Veritas log file contains all actions performed by HAD.

To check the Symantec log files

1 Access the following file:

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

2 Verify if there are any CVM errors logged in this file, since they may prove to be critical errors.

3 You can access 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 filescan provide valuable information for diagnosing problems. The system log can be checked in the following file:

```
/var/adm/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 v gen 1fc60002 membership 01
Port w gen 15ba0002 membership 01
```

Table 3-1 defines each GAB port's function.

For illustration of different GAB ports, See Figure 1-6 on page 21.

Table 3-1 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.
v	CVM uses this port for kernel-to-kernel communication.

Table 3-1 GAB port function *(continued)*

Port	Function
w	`vxconfigd` configuration daemon (module for CVM) uses this port for messaging.

Troubleshooting I/O fencing

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

The vxfentsthdw utility fails when SCSI TEST UNIT READY command fails

If you see a message that resembles as follows:

```
Issuing SCSI TEST UNIT READY to disk reserved by other node
FAILED.
Contact the storage provider to have the hardware configuration
fixed.
```

The disk array does not support returning success for a `SCSI TEST UNIT READY` command when another host has the disk reserved using SCSI-3 persistent reservations. This happens with the Hitachi Data Systems 99XX arrays if bit 186 of the system mode option is not enabled.

Node is unable to join cluster while another node is being ejected

A cluster that is currently fencing out (ejecting) a node from the cluster prevents a new node from joining the cluster until the fencing operation is completed. The following are example messages that appear on the console for the new node:

```
...VxFEN ERROR V-11-1-25 ... Unable to join running cluster
since cluster is currently fencing
a node out of the cluster.
```

If you see these messages when the new node is booting, the vxfen startup script on the node makes up to five attempts to join the cluster.

To manually join the node to the cluster when I/O fencing attempts fail

- If the vxfen script fails in the attempts to allow the node to join the cluster, restart vxfen driver with the command:

  ```
  # /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.
```

See "How vxfen driver checks for pre-existing split-brain condition" on page 88.

How vxfen driver checks for pre-existing split-brain condition

The vxfen driver functions to prevent an ejected node from rejoining the cluster after the failure of the private network links and before the private network links are repaired.

For example, suppose the cluster of system 1 and system 2 is functioning normally when the private network links are broken. Also suppose system 1 is the ejected system. When system 1 restarts before the private network links are restored, its membership configuration does not show system 2; however, when it attempts to register with the coordinator disks, it discovers system 2 is registered with them. Given this conflicting information about system 2, system 1 does not join the cluster and returns an error from vxfenconfig that resembles:

```
vxfenconfig: ERROR: There exists the potential for a preexisting
split-brain. The coordinator disks list no nodes which are in
the current membership. However, they also list nodes which are
not in the current membership.

I/O Fencing Disabled!
```

Also, the following information is displayed on the console:

```
<date> <system name> vxfen: WARNING: Potentially a preexisting
<date> <system name> split-brain.
<date> <system name> Dropping out of cluster.
<date> <system name> Refer to user documentation for steps
<date> <system name> required to clear preexisting split-brain.
```

```
<date> <system name>
<date> <system name> I/O Fencing DISABLED!
<date> <system name>
<date> <system name> gab: GAB:20032: Port b closed
```

However, the same error can occur when the private network links are working and both systems go down, system 1 restarts, and system 2 fails to come back up. From the view of the cluster from system 1, system 2 may still have the registrations on the coordinator disks.

To resolve actual and apparent potential split brain conditions

- Depending on the split brain condition that you encountered, do the following:

Actual potential split brain condition—system 2 is up and system 1 is ejected	1 Determine if system1 is up or not. 2 If system 1 is up and running, shut it down and repair the private network links to remove the split brain condition. 3 Restart system 1.
Apparent potential split brain condition—system 2 is down and system 1 is ejected	1 Physically verify that system 2 is down. Verify the systems currently registered with the coordinator disks. Use the following command: `# vxfenadm -g all -f /etc/vxfentab` The output of this command identifies the keys registered with the coordinator disks. 2 Clear the keys on the coordinator disks as well as the data disks using the `vxfenclearpre` command. See "Clearing keys after split brain using vxfenclearpre command" on page 89. 3 Make any necessary repairs to system 2. 4 Restart system 2.

Clearing keys after split brain using vxfenclearpre command

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

See "About vxfenclearpre utility" on page 62.

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 70.

Replacing defective disks when the cluster is offline

If the disk becomes defective or inoperable and you want to switch to a new diskgroup in a cluster that is offline, then perform the following procedure.

In a cluster that is online, you can replace the disks using the vxfenswap utility.

See "About vxfenswap utility" on page 64.

Review the following information to replace coordinator disk in the coordinator disk group, or to destroy a coordinator disk group.

Note the following about the procedure:

- When you add a disk, add the disk to the disk group vxfencoorddg and retest the group for support of SCSI-3 persistent reservations.
- You can destroy the coordinator disk group such that no registration keys remain on the disks. The disks can then be used elsewhere.

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

1. Log in as superuser on one of the cluster nodes.
2. If VCS is running, shut it down:

   ```
   # hastop -all
   ```

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

   ```
   # gabconfig -a
   ```

3. Stop I/O fencing on each node:

   ```
   # /etc/init.d/vxfen 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, vxfencoorddg) that contains the coordinator disks, so use the command:

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

where:

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

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

-C specifies that any import locks are removed.

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

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

- Verify whether the coordinator attribute is set to on.

```
# vxdg list vxfencoorddg |  grep flags: | grep coordinator
```

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

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

- Destroy the disk group.

```
# vxdg destroy vxfencoorddg
```

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

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

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

See "Testing the coordinator disk group using vxfentsthdw -c option" on page 54.

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:

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

10 If necessary, restart VCS on each node:

```
# hastart
```

The vxfenswap utility faults when echo or cat is used in .bashrc file

The vxfenswap utility faults when you use echo or cat to print messages in the .bashrc file for the nodes.

To recover the vxfenswap utility fault

- Verify whether the rcp or scp functions properly.

  If the vxfenswap operation is unsuccessful, use the `vxfenswap -cancel` command if required to roll back any changes that the utility made.

  See "About vxfenswap utility" on page 64.

Troubleshooting CVM

This section discusses troubleshooting CVM problems.

Shared disk group cannot be imported

If you see a message resembling:

```
vxvm:vxconfigd:ERROR:vold_pgr_register(/dev/vx/rdmp/disk_name):
local_node_id<0
Please make sure that CVM and vxfen are configured
and operating correctly
```

First, make sure that CVM is running. You can see the CVM nodes in the cluster by running the vxclustadm nidmap command.

```
# vxclustadm nidmap
Name          CVM Nid     CM Nid      State
system1         1           0           Joined: Master
system2         0           1           Joined: Slave
```

This above ouput 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:

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

Port b must exist on the local system.

Error importing shared disk groups

The following message may appear when importing shared disk group:

```
VxVM vxdg ERROR V-5-1-587 Disk group disk group name: import
failed: No valid disk found containing disk group
```

You may need to remove keys written to the disk.

For information about removing keys written to the disk:

See "Removing preexisting keys" on page 63.

Unable to start CVM

If you cannot start CVM, check the consistency between the `/etc/llthosts` and `main.cf` files for node IDs.

You may need to remove keys written to the disk.

For information about removing keys written to the disk:

See "Removing preexisting keys" on page 63.

CVMVolDg not online even though CVMCluster is online

When the CVMCluster resource goes online, the shared disk groups are automatically imported. If the disk group import fails for some reason, the CVMVolDg resources fault. Clearing and offlining the CVMVolDg type resources does not fix the problem.

To resolve the resource issue

1. Fix the problem causing the import of the shared disk group to fail.
2. Offline the service group containing the resource of type CVMVolDg as well as the service group containing the CVMCluster resource type.

3 Bring the service group containing the CVMCluster resource online.

4 Bring the service group containing the CVMVolDg resource online.

VxVM error messages

Table 3-2 contains VxVM error messages that are related to I/O fencing.

Table 3-2 VxVM error messages for I/O fencing

Message	Explanation
vold_pgr_register(disk_path): failed to open the vxfen device. Please make sure that the vxfen driver is installed and configured.	The vxfen driver is not configured. Follow the instructions to set up these disks and start I/O fencing. You can then clear the faulted resources and bring the service groups online.
vold_pgr_register(disk_path): Probably incompatible vxfen driver.	Incompatible versions of VxVM and the vxfen driver are installed on the system. Install the proper version of SF Sybase CE.

Troubleshooting interconnects

This section discusses troubleshooting interconnect 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, issue the following `vxdctl` command to force the VxVM configuration daemon `vxconfigd` to rescan the disks:

```
# vxdctl enable
```

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_device=/quorum/quorum.dat --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
```

Chapter 4

Prevention and recovery strategies

This chapter includes the following topics:

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 6 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 v (CVM kernel messaging)
- port w (CVM vxconfigd)

The following command can be used to verify the state of GAB ports:

```
# gabconfig -a
```

GAB Port Memberships

```
Port a gen 7e6e7e05 membership 01
Port b gen 58039502 membership 01
Port f gen 1ea84702 membership 01
Port h gen cf430b02 membership 01
Port v gen db411702 membership 01
Port w gen cf430b02 membership 01
```

The data indicates that all the GAB ports are up on the cluster having nodes 0 and 1.

For more information on the GAB ports in SF Sybase CE cluster, see the *Veritas Storage Foundation for Sybase ASE CE Installation and Configuration Guide*.

Examining GAB seed membership

The number of systems that participate in the cluster is specified as an argument to the `gabconfig` command in /etc/gabtab. In the following example, two nodes are expected to form a cluster:

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

```
/sbin/gabconfig -c -n2
```

GAB waits until the specified number of nodes becomes available to automatically create the port “a” membership. Port “a” indicates GAB membership for an SF 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.

Manual GAB membership seeding

It is possible that one of the nodes does not come up when all the nodes in the cluster are restarted, due to the "minimum seed requirement" safety that is enforced by GAB. Human intervention is needed to safely determine that the other node is in fact not participating in its own mini-cluster.

The following should be carefully validated before manual seeding, to prevent introducing split-brain and subsequent data corruption:

- Verify that none of the other nodes in the cluster have a port “a” membership
- Verify that none of the other nodes have any shared disk groups imported
- Determine why any node that is still running does not have a port “a” membership

Run the following command to manually seed GAB membership:

```
# gabconfig -cx
```

Refer to gabconfig (1M) for more details.

Evaluating VCS I/O fencing ports

I/O Fencing (VxFEN) uses a dedicated port that GAB provides for communication across nodes in the cluster. You can see this port as port ‘b’ when `gabconfig -a` runs on any node in the cluster. The entry corresponding to port ‘b’ in this membership indicates the existing members in the cluster as viewed by I/O Fencing.

GAB uses port "a" for maintaining the cluster membership and must be active for I/O Fencing to start.

To check whether fencing is enabled in a cluster, the '-d' option can be used with vxfenadm (1M) to display the I/O Fencing mode on each cluster node. Port "b" membership should be present in the output of `gabconfig -a` and the output should list all the nodes in the cluster.

If the GAB ports that are needed for I/O fencing are not up, that is, if port "a" is not visible in the output of `gabconfig -a` command, LLT and GAB must be started on the node.

The following commands can be used to start LLT and GAB respectively:

To start LLT on each node:

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

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

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

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

```
# gabconfig -a |grep 'Port a'
Port a gen   614605 membership 01

# cat /etc/llthosts
0 system1
1 system2
```

Check the llttab on both nodes:

```
# cat /etc/llttab
set-node system1
set-cluster Cluster id
link qfe0 eth-00:15:17:48:b4:98 - ether - -
link qfe1 eth-00:15:17:48:b4:99 - ether - -
```

To start GAB, on each node:

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

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

```
GAB INFO V-15-1-20021 GAB available

GAB INFO V-15-1-20026 Port a registration waiting for seed port
membership
```

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 both nodes

  ```
  # gabconfig -a
  ```

- Registered keys present on the coordinator disks

  ```
  # vxfenadm -g 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 30.

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

```
# vxdg list |grep data

sybdata_101 enabled,shared,cds 1201715530.28.pushover
```

Select the data disk belonging to sybdata_101:

```
# vxdisk -o alldgs list |grep sybdata_101

c1t2d0s2  auto:cdsdisk    c1t2d0s2        sybdata_101  online shared
c1t2d1s2 auto:cdsdisk    c1t2d1s2       sybdata_101  online shared
c1t2d2s2 auto:cdsdisk    c1t2d2s2       sybdata_101  online shared
```

The following command lists the PR keys:

```
# vxdisk -o listreserve list c1t2d0s2

.....

.....

Multipathing information:
numpaths:   1
hdisk6  state=enabled
Reservations:
BPGR0000 (type: Write Exclusive Registrants Only, scope: LUN(0x0))
2 registered pgr keys
BPGR0004
APGR0004
```

Alternatively, the PR keys can be listed using `vxfenadm` command:

```
# echo "/dev/vx/dmp/c1t2d0s2" > /tmp/disk71

# vxfenadm -g all -f /tmp/disk71

Device Name: /dev/vx/dmp/c1t2d0s2
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 vxfenadm utility" on page 60.

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 -g all -f /etc/vxfentab
```

There should be a key for each node in the operating cluster on each of the coordinator disks for normal cluster operation.

Identifying a faulty coordinator LUN

The utility vxfentsthdw (1M) provided with I/O Fencing can be used to identify faulty coordinator LUNs. This utility must be run from any two nodes in the cluster. The coordinator LUN, which needs to be checked, should be supplied to the utility.

See "About vxfentsthdw utility" on page 51.

Collecting I/O Fencing kernel logs

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

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

Collecting important CVM logs

You need to stop and restart the cluster to collect detailed CVM logs.

- Stop the cluster.
  `# hastop -all`
- 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=`$VXCLUSTADM -m vcs -t $TRANSPORT startnode 2> $CVM_ERR_FILE``
    Modified string: **`clust_run=`$VXCLUSTADM -m vcs -t $TRANSPORT -T startnode 2> $CVM_ERR_FILE``**
  - Enable logging on vxconfigd daemon.
    `# vxdctl debug 9 /var/adm/vx/vxconfigd_debug.out`

- Start the cluster

  ```
  # hastart
  ```

The debug information that is enabled is accumulated in the system console log and in the text file /var/adm/vx/vxconfigd_debug.out

The CVM kernel message dump can be collected on a live node as follows:

```
# /etc/vx/diag.d/kmsgdump -k 2000 >/var/adm/vx/kmsgdump.out
```

Appendix A

SFCFS architecture

This appendix includes the following topics:

Storage Foundation Cluster File System benefits and applications

This section describes the SFCFS benefits and applications.

This section includes the following topics:

How Storage Foundation Cluster File System works

SFCFS simplifies or eliminates system administration tasks that result from the following hardware limitations:

- The SFCFS single file system image administrative model simplifies administration by making all file system management operations and resizing and reorganization (defragmentation) can be performed from any node.
- Because all servers in a cluster have access to SFCFS cluster-shareable file systems, keeping data consistent across multiple servers is automatic. All cluster nodes have access to the same data, and all data is accessible by all servers using single server file system semantics.
- Because all files can be accessed by all servers, applications can be allocated to servers to balance load or meet other operational requirements. Similarly,

failover becomes more flexible because it is not constrained by data accessibility.

- Because each SFCFS file system can be on any node in the cluster, the file system recovery portion of failover time in an *n*-node cluster can be reduced by a factor of *n* by distributing the file systems uniformly across cluster nodes.
- Enterprise RAID subsystems can be used more effectively because all of their capacity can be mounted by all servers, and allocated by using administrative operations instead of hardware reconfigurations.
- Larger volumes with wider striping improve application I/O load balancing. Not only is the I/O load of each server spread across storage resources, but with SFCFS shared file systems, the loads of all servers are balanced against each other.
- Extending clusters by adding servers is easier because each new server's storage configuration does not need to be set up—new servers simply adopt the cluster-wide volume and file system configuration.

When to use Storage Foundation Cluster File System

You should use SFCFS for any application that requires the sharing of files, such as for home directories and boot server files, Web pages, and for cluster-ready applications. SFCFS is also applicable when you want highly available standby data, in predominantly read-only environments where you just need to access data, or when you do not want to rely on NFS for file sharing.

Almost all applications can benefit from SFCFS. Applications that are not "cluster-aware" can operate on and access data from anywhere in a cluster. If multiple cluster applications running on different servers are accessing data in a cluster file system, overall system I/O performance improves due to the load balancing effect of having one cluster file system on a separate underlying volume. This is automatic; no tuning or other administrative action is required.

Many applications consist of multiple concurrent threads of execution that could run on different servers if they had a way to coordinate their data accesses. SFCFS provides this coordination. Such applications can be made cluster-aware allowing their instances to co-operate to balance client and data access load, and thereby scale beyond the capacity of any single server. In such applications, SFCFS provides shared data access, enabling application-level load balancing across cluster nodes.

SFCFS provides the following features:

- For single-host applications that must be continuously available, SFCFS can reduce application failover time because it provides an already-running file system environment in which an application can restart after a server failure.

- For parallel applications, such as distributed database management systems and Web servers, SFCFS provides shared data to all application instances concurrently. SFCFS also allows these applications to grow by the addition of servers, and improves their availability by enabling them to redistribute load in the event of server failure simply by reassigning network addresses.
- For workflow applications, such as video production, in which very large files are passed from station to station, the SFCFS eliminates time consuming and error prone data copying by making files available at all stations.
- For backup, the SFCFS can reduce the impact on operations by running on a separate server, accessing data in cluster-shareable file systems.

The following are examples of applications and how they might work with SFCFS:

- Using Storage Foundation Cluster File System on file servers
  Two or more servers connected in a cluster configuration (that is, connected to the same clients and the same storage) serve separate file systems. If one of the servers fails, the other recognizes the failure, recovers, assumes the primaryship, and begins responding to clients using the failed server's IP addresses.
- Using Storage Foundation Cluster File System on web servers
  Web servers are particularly suitable to shared clustering because their application is typically read-only. Moreover, with a client load balancing front end, a Web server cluster's capacity can be expanded by adding a server and another copy of the site. A SFCFS-based cluster greatly simplifies scaling and administration for this type of application.

When the Storage Foundation Cluster File System primary fails

If the server on which the SFCFS primary is running fails, the remaining cluster nodes elect a new primary. The new primary reads the file system intent log and completes any metadata updates that were in process at the time of the failure. Application I/O from other nodes may block during this process and cause a delay. When the file system is again consistent, application processing resumes.

Because nodes using a cluster file system in secondary node do not update file system metadata directly, failure of a secondary node does not require metadata repair. SFCFS recovery from secondary node failure is therefore faster than from primary node failure.

About Storage Foundation Cluster File System and the Group Lock Manager

SFCFS uses the Veritas Group Lock Manager (GLM) to reproduce UNIX single-host file system semantics in clusters. UNIX file systems make writes appear atomic. This means when an application writes a stream of data to a file, a subsequent application reading from the same area of the file retrieves the new data, even if it has been cached by the file system and not yet written to disk. Applications cannot retrieve stale data or partial results from a previous write.

To reproduce single-host write semantics, system caches must be kept coherent, and each must instantly reflect updates to cached data, regardless of the node from which they originate.

About asymmetric mounts

A VxFS file system mounted with the `mount -o cluster` option is a cluster, or shared mount, as opposed to a non-shared or local mount. A file system mounted in shared mode must be on a VxVM shared volume in a cluster environment. A local mount cannot be remounted in shared mode and a shared mount cannot be remounted in local mode. File systems in a cluster can be mounted with different read/write options. These are called asymmetric mounts.

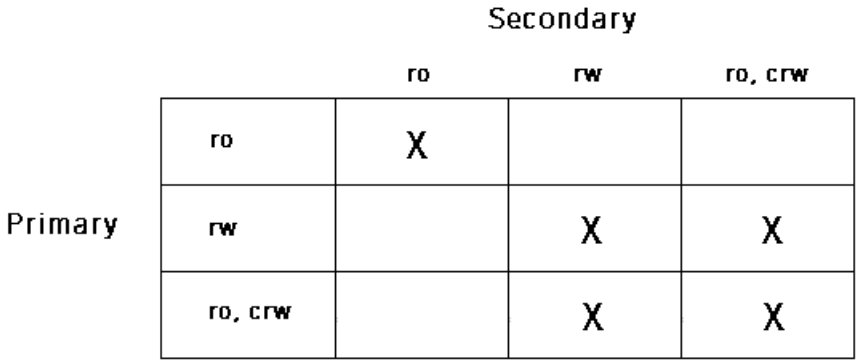
Asymmetric mounts allow shared file systems to be mounted with different read/write capabilities. One node in the cluster can mount read/write, while other nodes mount read-only.

You can specify the cluster read-write (`crw`) option when you first mount the file system, or the options can be altered when doing a remount (`mount -o remount`).

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

About asymmetric mounts describes the first column in the following table shows the mode in which the primary is mounted:

Figure A-1 Primary and secondary mounts



The check marks indicate the mode secondary mounts can use.

Mounting the primary with only the `-o cluster,ro` option prevents the secondaries from mounting in a different mode; that is, read-write.

Note: `rw` implies read-write capability throughout the cluster.

Parallel I/O

Some distributed applications read and write to the same file concurrently from one or more nodes in the cluster; for example, any distributed application where one thread appends to a file and there are one or more threads reading from various regions in the file. Several high-performance compute (HPC) applications can also benefit from this feature, where concurrent I/O is performed on the same file. Applications do not require any changes to use parallel I/O feature.

Traditionally, the entire file is locked to perform I/O to a small region. To support parallel I/O, SFCFS locks ranges in a file that correspond to an I/O request. The granularity of the locked range is a page. Two I/O requests conflict if at least one is a write request, and the I/O range of the request overlaps the I/O range of the other.

The parallel I/O feature enables I/O to a file by multiple threads concurrently, as long as the requests do not conflict. Threads issuing concurrent I/O requests could be executing on the same node, or on a different node in the cluster.

An I/O request that requires allocation is not executed concurrently with other I/O requests. Note that when a writer is extending the file and readers are lagging behind, block allocation is not necessarily done for each extending write.

If the file size can be predetermined, the file can be preallocated to avoid block allocations during I/O. This improves the concurrency of applications performing parallel I/O to the file. Parallel I/O also avoids unnecessary page cache flushes and invalidations using range locking, without compromising the cache coherency across the cluster.

For applications that update the same file from multiple nodes, the `-nomtime` mount option provides further concurrency. Modification and change times of the file are not synchronized across the cluster, which eliminates the overhead of increased I/O and locking. The timestamp seen for these files from a node may not have the time updates that happened in the last 60 seconds.

Storage Foundation Cluster File System namespace

The mount point name must remain the same for all nodes mounting the same cluster file system. This is required for the VCS mount agents (online, offline, and monitoring) to work correctly.

Storage Foundation Cluster File System backup strategies

The same backup strategies used for standard VxFS can be used with SFCFS because the APIs and commands for accessing the namespace are the same. File System checkpoints provide an on-disk, point-in-time copy of the file system. Because performance characteristics of a checkpointed file system are better in certain I/O patterns, they are recommended over file system snapshots (described below) for obtaining a frozen image of the cluster file system.

File System snapshots are another method of a file system on-disk frozen image. The frozen image is non-persistent, in contrast to the checkpoint feature. A snapshot can be accessed as a read-only mounted file system to perform efficient online backups of the file system. Snapshots implement “copy-on-write” semantics that incrementally copy data blocks when they are overwritten on the snapped file system. Snapshots for cluster file systems extend the same copy-on-write mechanism for the I/O originating from any cluster node.

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

The following are several characteristics of a cluster snapshot:

- A snapshot for a cluster mounted file system can be mounted on any node in a cluster. The file system can be a primary, secondary, or secondary-only. A stable image of the file system is provided for writes from any node.
- Multiple snapshots of a cluster file system can be mounted on the same or different cluster node.
- A snapshot is accessible only on the node mounting a snapshot. The snapshot device cannot be mounted on two nodes simultaneously.
- The device for mounting a snapshot can be a local disk or a shared volume. A shared volume is used exclusively by a snapshot mount and is not usable from other nodes as long as the snapshot is active on that device.
- On the node mounting a snapshot, the snapped file system cannot be unmounted while the snapshot is mounted.
- A SFCFS snapshot ceases to exist if it is unmounted or the node mounting the snapshot fails. However, a snapshot is not affected if a node leaves or joins the cluster.
- A snapshot of a read-only mounted file system cannot be taken. It is possible to mount snapshot of a cluster file system only if the snapped cluster file system is mounted with the `crw` option.

In addition to file-level frozen images, there are volume-level alternatives available for shared volumes using mirror split and rejoin. Features such as Fast Mirror Resync and Space Optimized snapshot are also available.

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

Synchronize time on Cluster File Systems

SFCFS requires that the system clocks on all nodes are synchronized using some external component such as the Network Time Protocol (NTP) daemon. If the nodes are not in sync, timestamps for creation (`ctime`) and modification (`mtime`) may not be consistent with the sequence in which operations actually happened.

Distribute a load on a cluster

You can use the `fsclustadm` to designate a SFCFS primary. The `fsclustadm setprimary` mount point can be used to change the primary. This change to the primary is not persistent across unmounts or reboots. The change is in effect as long as one or more nodes in the cluster have the file system mounted. The primary selection policy can also be defined by a VCS attribute associated with the SFCFS mount resource.

For example, if you have eight file systems and four nodes, designating two file systems per node as the primary is beneficial. The first node that mounts a file system becomes the primary for that file system.

File system tuneables

Tuneable parameters are updated at the time of mount using the `tunefstab` file or `vxtunefs` command. The file system `tunefs` parameters are set to be identical on all nodes by propagating the parameters to each cluster node. When the file system is mounted on the node, the `tunefs` parameters of the primary node are used. The `tunefstab` file on the node is used if this is the first node to mount the file system. Symantec recommends that this file be identical on each node.

Split-brain and jeopardy handling

A split-brain occurs when the cluster membership view differs among the cluster nodes, increasing the chance of data corruption. Membership change also occurs when all private-link cluster interconnects fail simultaneously, or when a node is unable to respond to heartbeat messages.With I/O fencing, the potential for data corruption is eliminated. I/O fencing requires disks that support SCSI-3 PGR.

Jeopardy state

In the absence of I/O fencing, SFCFS installation requires two heartbeat links. When a node is down to a single heartbeat connection, SFCFS can no longer discriminate between loss of a system and loss of the final network connection. This state is defined as jeopardy.

SFCFS employs jeopardy to prevent data corruption following a split-brain.

In certain following scenarios, the possibility of data corruption remains:

- All links go down simultaneously.
- A node hangs and is unable to respond to heartbeat messages.

To eliminate the chance of data corruption in these scenarios, I/O fencing is required. With I/O fencing, the jeopardy state does not require special handling by the SFCFS stack.

Jeopardy handling

For installations that do not support SCSI-3 PGR, potential split-brain conditions are safeguarded by jeopardy handling. If any cluster node fails following a jeopardy state notification, the cluster file system mounted on the failed nodes is disabled.

If a node fails after the jeopardy state notification, all cluster nodes also leave the shared disk group membership.

Recover from jeopardy

The disabled file system can be restored by a force unmount and the resource can be brought online without rebooting, which also brings the shared disk group resource online. Note that if the jeopardy condition is not fixed, the nodes are susceptible to leaving the cluster again on subsequent node failure.

See the *Veritas Cluster Server User's Guide*.

Fencing

With the use of I/O enabled fencing, all remaining cases with the potential to corrupt data (for which jeopardy handling cannot protect) are addressed.

See "About preventing data corruption with I/O fencing" on page 30.

Single network link and reliability

Certain environments may prefer using a single private link or a public network for connecting nodes in a cluster, despite the loss of redundancy for dealing with network failures. The benefits of this approach include simpler hardware topology and lower costs; however, there is obviously a tradeoff with high availability.

For the above environments, SFCFS provides the option of a single private link, or using the public network as the private link if I/O fencing is present. I/O fencing is used to handle split-brain scenarios. The option for single network is given during installation.

See "About preventing data corruption with I/O fencing" on page 30.

Configuring low priority link

LLT can be configured to use a low-priority network link as a backup to normal heartbeat channels. Low-priority links are typically configured on the customer's public or administrative network. This typically results in a completely different network infrastructure than the cluster private interconnect, and reduces the chance of a single point of failure bringing down all links. The low-priority link is not used for cluster membership traffic until it is the only remaining link. In normal operation, the low-priority link carries only heartbeat traffic for cluster membership and link state maintenance. The frequency of heartbeats drops 50 percent to reduce network overhead. When the low-priority link is the only remaining network link, LLT also switches over all cluster status traffic. Following

repair of any configured private link, LLT returns cluster status traffic to the high-priority link.

LLT links can be added or removed while clients are connected. Shutting down GAB or the high-availability daemon, had, is not required.

To add a link

- To add a link, type the following command:

```
# lltconfig -d device -t tag
```

To remove a link

- To remove a link, type the following command:

```
# lltconfig -u tag
```

Changes take effect immediately and are lost on the next reboot. For changes to span reboots you must also update `/etc/lltab`.

Note: LLT clients do not recognize the difference unless only one link is available and GAB declares jeopardy.

I/O error handling policy

I/O errors can occur for several reasons, including failures of Fibre Channel link, host-bus adapters, and disks. SFCFS disables the file system on the node encountering I/O errors. The file system remains available from other nodes.

After the hardware error is fixed (for example, the Fibre Channel link is reestablished), the file system can be force unmounted and the mount resource can be brought online from the disabled node to reinstate the file system.

Appendix B

File System and Volume Manager functionality

This appendix includes the following topics:

About Veritas File System features supported in cluster file systems

The Veritas Storage Foundation Cluster File System is based on the Veritas File System (VxFS).

Most of the major features of VxFS local file systems are available on cluster file systems, including the following features:

- Extent-based space management that maps files up to a terabyte in size
- Fast recovery from system crashes using the intent log to track recent file system metadata updates
- Online administration that allows file systems to be extended and defragmented while they are in use

The list of supported features and commands that operate on SFCFS. Every VxFS manual page has a section on Storage Foundation Cluster File System Issues with information on whether the command functions on a cluster-mounted file system and indicates any difference in behavior from local mounted file systems.

Veritas File System features in cluster file systems

Table B-1 describes the VxFS supported features and commands for SFCFS.

Table B-1 Veritas File System features in cluster file systems

Features	Description
Storage Checkpoints	Storage Checkpoints are supported on cluster file systems, but are licensed only with other Veritas products.
Snapshots	Snapshots are supported on cluster file systems.
Quotas	Quotas are supported on cluster file systems.
NFS mounts	You can mount cluster file systems to NFS.
Nested Mounts	You can use a directory on a cluster mounted file system as a mount point for a local file system or another cluster file system.
Freeze and thaw	Synchronizing operations, which require freezing and thawing file systems, are done on a cluster-wide basis.
Memory mapping	Shared memory mapping established by the `mmap`() function is supported on SFCFS. See the `mmap`(2) manual page.
Disk layout versions	SFCFS supports only disk layout Version 6 and 7. Cluster mounted file systems can be upgraded, a local mounted file system can be upgraded, unmounted, and mounted again as part of a cluster. Use the `fstyp -v` special_device command to ascertain the disk layout version of a VxFS file system. Use the `vxupgrade` command to update the disk layout version.
Locking	Advisory file and record locking are supported on SFCFS. For the F_GETLK command, if there is a process holding a conflicting lock, the `l_pid` field returns the process ID of the process holding the conflicting lock. The nodeid-to-node name translation can be done by examining the `/etc/llthosts` file or with the `fsclustadm` command. Mandatory locking, and deadlock detection supported by traditional `fcntl` locks, are not supported on SFCFS. See the `fcntl`(2) manual page.

Veritas File System features not in cluster file systems

Table B-2 describes functionality as not supported and may not be expressly prevented from operating on cluster file systems, but the actual behavior is indeterminate.

It is not advisable to use unsupported functionality on SFCFS, or to alternate mounting file systems with these options as local and cluster mounts.

Table B-2 Veritas File System features not in cluster file systems

Unsupported features	Comments
qlog	Quick log is not supported.
Swap files	Swap files are not supported on cluster mounted file system.
mknod	The `mknod` command cannot be used to create devices on a cluster mounted file system.
Cache advisories	Cache advisories are set with the mount command on individual file systems, but are not propagated to other nodes of a cluster.
Cached Quick I/O	This Quick I/O for Databases feature that caches data in the file system cache is not supported.
Commands that depend on file access times	File access times may appear different across nodes because the `atime` file attribute is not closely synchronized in a cluster file system. So utilities that depend on checking access times may not function reliably.

About Veritas Volume Manager cluster functionality

Veritas Volume Manager cluster functionality (CVM) allows up to 32 nodes in a cluster to simultaneously access and manage a set of disks under VxVM control (VM disks). The same logical view of the disk configuration and any changes are available on each node. When the cluster functionality is enabled, all cluster nodes can share VxVM objects. Features provided by the base volume manager, such as mirroring, fast mirror resync and dirty region logging are also supported in the cluster environment.

Note: RAID-5 volumes are not supported on a shared disk group.

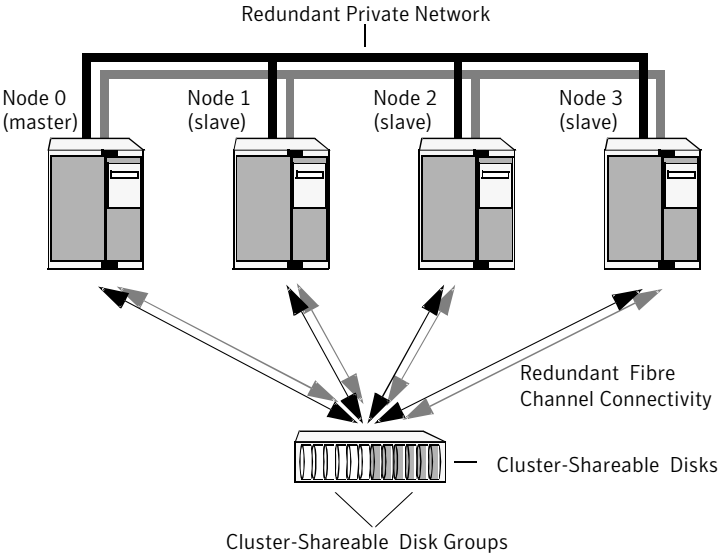
To implement cluster functionality, VxVM works together with the cluster monitor daemon provided by the host operating system or by VCS. The cluster monitor informs VxVM of changes in cluster membership. Each node starts up independently and has its own cluster monitor, plus its own copies of the operating system and CVM. When a node joins a cluster it gains access to shared disks. When a node leaves a cluster, it no longer has access to shared disks. A node joins a cluster when the cluster monitor is started on that node.

Figure B-1 illustrates a simple cluster arrangement consisting of four nodes with similar or identical hardware characteristics (CPUs, RAM and host adapters), and configured with identical software (including the operating system).

The nodes are fully connected by a private network and they are also separately connected to shared external storage (either disk arrays or JBODs: just a bunch of disks) via Fibre Channel. Each node has two independent paths to these disks, which are configured in one or more cluster-shareable disk groups.

The private network allows the nodes to share information about system resources and about each other's state. Using the private network, any node can recognize which nodes are currently active, which are joining or leaving the cluster, and which have failed. The private network requires at least two communication channels to provide redundancy against one of the channels failing. If only one channel were used, its failure would be indistinguishable from node failure—a condition known as network partitioning.

Figure B-1 Example of a four node cluster



To the cluster monitor, all nodes are the same. VxVM objects configured within shared disk groups can potentially be accessed by all nodes that join the cluster. However, the cluster functionality of VxVM requires one node to act as the master node; all other nodes in the cluster are slave nodes. Any node is capable of being the master node, which is responsible for coordinating certain VxVM activities.

Note: You must run commands that configure or reconfigure VxVM objects on the master node. Tasks that must be initiated from the master node include setting up shared disk groups and creating and reconfiguring volumes.

VxVM designates the first node to join a cluster the master node. If the master node leaves the cluster, one of the slave nodes is chosen to be the new master. In the preceding example, node 0 is the master node and nodes 1, 2 and 3 are slave nodes.

Shared disk groups overview

This section provides an overview of shared disk groups.

This section includes the following topics:

- Private and shared disk groups
- Activation modes of shared disk groups
- Connectivity policy of shared disk groups
- Limitations of shared disk groups

Private and shared disk groups

Table B-3 describes the disk group types.

Table B-3 Disk group types

Disk group	Description
Private	Belongs to only one node. A private disk group is only imported by one system. Disks in a private disk group may be physically accessible from one or more systems, but import is restricted to one system only. The root disk group is always a private disk group.
Shared	Is shared by all nodes. A shared (or cluster-shareable) disk group is imported by all cluster nodes. Disks in a shared disk group must be physically accessible from all systems that may join the cluster.

In a cluster, most disk groups are shared. Disks in a shared disk group are accessible from all nodes in a cluster, allowing applications on multiple cluster nodes to simultaneously access the same disk. A volume in a shared disk group can be simultaneously accessed by more than one node in the cluster, subject to licensing and disk group activation mode restrictions.

You can use the `vxdg` command to designate a disk group as cluster-shareable. When a disk group is imported as cluster-shareable for one node, each disk header

is marked with the cluster ID. As each node subsequently joins the cluster, it recognizes the disk group as being cluster-shareable and imports it. You can also import or deport a shared disk group at any time; the operation takes places in a distributed fashion on all nodes.

Each physical disk is marked with a unique disk ID. When cluster functionality for VxVM starts on the master, it imports all shared disk groups (except for any that have the `noautoimport` attribute set). When a slave tries to join a cluster, the master sends it a list of the disk IDs that it has imported, and the slave checks to see if it can access them all. If the slave cannot access one of the listed disks, it abandons its attempt to join the cluster. If it can access all of the listed disks, it imports the same shared disk groups as the master and joins the cluster. When a node leaves the cluster, it deports all its imported shared disk groups, but they remain imported on the surviving nodes.

Reconfiguring a shared disk group is performed with the co-operation of all nodes. Configuration changes to the disk group happen simultaneously on all nodes and the changes are identical. Such changes are atomic in nature, which means that they either occur simultaneously on all nodes or not at all.

Whether all members of the cluster have simultaneous read and write access to a cluster-shareable disk group depends on its activation mode setting.

See "Activation modes of shared disk groups" on page 120.

The data contained in a cluster-shareable disk group is available as long as at least one node is active in the cluster. The failure of a cluster node does not affect access by the remaining active nodes. Regardless of which node accesses a cluster-shareable disk group, the configuration of the disk group looks the same.

Note: Applications running on each node can access the data on the VM disks simultaneously. VxVM does not protect against simultaneous writes to shared volumes by more than one node. It is assumed that applications control consistency (by using Veritas Storage Foundation Cluster File System or a distributed lock manager, for example).

Activation modes of shared disk groups

A shared disk group must be activated on a node in order for the volumes in the disk group to become accessible for application I/O from that node. The ability of applications to read from or to write to volumes is dictated by the activation mode of a shared disk group. Valid activation modes for a shared disk group are `exclusivewrite`, `readonly`, `sharedread`, `sharedwrite`, and `off` (inactive).

Note: Disk group activation was a new feature in VxVM 3.0. To maintain compatibility with previous releases, the default activation mode for shared disk groups is `shared-write`.

Special uses of clusters, such as high availability (HA) applications and off-host backup, can use disk group activation to explicitly control volume access from different nodes in the cluster.

Table B-4 describes activation modes for shared disk groups.

Table B-4 Activation modes for shared disk groups

Activation mode	Description
`exclusivewrite (ew)`	The node has exclusive write access to the disk group. No other node can activate the disk group for write access.
`readonly (ro)`	The node has read access to the disk group and denies write access for all other nodes in the cluster. The node has no write access to the disk group. Attempts to activate a disk group for either of the write modes on other nodes fail.
`sharedread (sr)`	The node has read access to the disk group. The node has no write access to the disk group, however other nodes can obtain write access.
`sharedwrite (sw)`	The node has write access to the disk group.
`off`	The node has neither read nor write access to the disk group. Query operations on the disk group are permitted.

Table B-5 summarizes the allowed and conflicting activation modes for shared disk groups.

Table B-5 Allowed and conflicting activation modes

Disk group activated in cluster as...	exclusive-write	readonly	sharedread	sharedwrite
exclusivewrite	Fails	Fails	Succeeds	Fails
readonly	Fails	Succeeds	Succeeds	Fails
sharedread	Succeeds	Succeeds	Succeeds	Succeeds
sharedwrite	Fails	Fails	Succeeds	Succeeds

To place activation modes under user control

- Create a defaults file `/etc/default/vxdg` containing the following lines:

```
enable_activation=true
default_activation_mode=activation-mode
```

The activation-mode is one of `exclusivewrite`, `readonly`, `sharedread`, `sharedwrite`, or `off`.

When a shared disk group is created or imported, it is activated in the specified mode. When a node joins the cluster, all shared disk groups accessible from the node are activated in the specified mode.

The activation mode of a disk group controls volume I/O from different nodes in the cluster. It is not possible to activate a disk group on a given node if it is activated in a conflicting mode on another node in the cluster. When enabling activation using the defaults file, it is recommended that this file be made identical on all nodes in the cluster. Otherwise, the results of activation are unpredictable.

If the defaults file is edited while the `vxconfigd` daemon is already running, the `vxconfigd` process must be restarted for the changes in the defaults file to take effect.

If the default activation mode is anything other than `off`, an activation following a cluster join, or a disk group creation or import can fail if another node in the cluster has activated the disk group in a conflicting mode.

To display the activation mode for a shared disk group, use the `vxdg list diskgroup` command.

You can also use the `vxdg` command to change the activation mode on a shared disk group.

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

Connectivity policy of shared disk groups

The nodes in a cluster must always agree on the status of a disk. In particular, if one node cannot write to a given disk, all nodes must stop accessing that disk before the results of the write operation are returned to the caller. Therefore, if a node cannot contact a disk, it should contact another node to check on the disk's status. If the disk fails, no node can access it and the nodes can agree to detach the disk. If the disk does not fail, but rather the access paths from some of the nodes fail, the nodes cannot agree on the status of the disk.

Table B-6 describes the policies for resolving this type of discrepancy.

Table B-6 Policies

Policy	Description
Global	The detach occurs cluster-wide (globally) if any node in the cluster reports a disk failure. This is the default policy.
Local	In the event of disks failing, the failures are confined to the particular nodes that saw the failure. However, this policy is not highly available because it fails the node even if one of the mirrors is available. Note that an attempt is made to communicate with all nodes in the cluster to ascertain the disks' usability. If all nodes report a problem with the disks, a cluster-wide detach occurs.

Limitations of shared disk groups

The cluster functionality of VxVM does not support RAID-5 volumes, or task monitoring for cluster-shareable disk groups. These features can, however, be used in private disk groups that are attached to specific nodes of a cluster. Online relayout is supported provided that it does not involve RAID-5 volumes.

The root disk group cannot be made cluster-shareable. It must be private.

Only raw device access may be performed via the cluster functionality of VxVM. It does not support shared access to file systems in shared volumes unless the appropriate software, such as Veritas Storage Foundation Cluster File System, is installed and configured.

If a shared disk group contains unsupported objects, deport it and then re-import the disk group as private on one of the cluster nodes. Reorganize the volumes into layouts that are supported for shared disk groups, and then deport and re-import the disk group as shared.

Index